NAACL HLT 2009

BioNLP 2009

Proceedings of the Workshop

June 4-5, 2009
Boulder, Colorado

Production and Manufacturing by
Omnipress Inc.
2600 Anderson Street
Madison, WI 53707
USA

BioNLP Sponsor:

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-932432-30-5

BioNLP 2009

K. Bretonnel Cohen, Dina Demner-Fushman, Sophia Ananiadou,
John Pestian, Jun'ichi Tsujii, and Bonnie Webber

1 Introduction

Yearly BioNLP workshops have been held in conjunction with Association for Computational Linguistics and North American Association for Computational Linguistics conferences since 2002. Whereas other venues, such as NLP sessions at biomedical informatics and computational biology meetings, provide excellent opportunities for presenting applications of NLP in the biomedical domain, the ACL BioNLP workshop has become the venue that is most characterized by representation of work in a wide variety of areas of NLP. The BioNLP workshop has consistently been a venue for presenting work that is innovative, novel, and challenging from an NLP perspective. In addition to providing a venue for fundamental BioNLP research, this workshop exposes BioNLP researchers to the latest achievements in other NLP areas and facilitates dissemination of knowledge acquired in the BioNLP domain to the wider NLP community.

Compared to previous years, BioNLP 2009 was novel in two ways. The first is that it is the first workshop since formation of the SIGBIOMED Association for Computational Linguistics Special Interest Group. The second is that for the first time, there was a shared task associated with the workshop. This shared task is documented in a separate proceedings volume.

2 Submissions, acceptance rate, and themes

The workshop received 29 submissions, of which twelve were accepted as full papers and an additional twelve were accepted as posters. A number of themes were evident in this year's papers and posters. Lexical semantics was especially well-represented this year, with papers on ontology selection [10], lexicon construction [12], and synonymy [3]. Information extraction was also well-represented, with papers in this area tackling both the genomic [2, 8], and the clinical [1] domain. This included work that is novel in the biomedical domain in terms of dealing with speech and with the dental domain [1]. This year also saw continued work on contextual issues in biomedical text mining [6, 7]. Finally, the program was rounded out with work on a new formulation of the named entity recognition problem [11], the hot topic of species identification [5], and word sense disambiguation [9] and summarization [4].

Acknowledgments

The greatest debt owed by the organizers of a workshop like this is to the authors who graciously chose BioNLP 2009 as the venue in which to share the fruits of the countless hours of research that went into the work submitted for consideration. The next-biggest debt is, without question, to the many program committee

members (listed elsewhere in this volume); they produced three reviews per paper on a tight review schedule and with an admirable level of insight. Finally, we acknowledge the gracious sponsorship of the Computational Medicine Center and Division of Biomedical Informatics, Cincinnati Children's Hospital Medical Center.

References

[1] Lee Christensen, Henk Harkema, Peter Haug, Jeannie Irwin, and Wendy Chapman. ONYX: A system for the semantic analysis of clinical text. In *BioNLP*, 2009.

[2] Udo Hahn, Katrin Tomanek, Ekaterina Buyko, Jung-jae Kim, and Dietrich Rebholz-Schuhmann. How feasible and robust is the automatic extraction of gene regulation events? A cross-method evaluation under lab and real-life conditions. In *BioNLP*, 2009.

[3] Thierry Hamon and Natalia Grabar. Exploring graph structure for detection of reliability zones within synonym resources: Experiment with the Gene Ontology. In *BioNLP*, 2009.

[4] Feng Jin, Minlie Huang, Zhiyong Lu, and Xiaoyan Zhu. Towards automatic generation of gene summary. In *BioNLP*, 2009.

[5] Thomas Kappeler, Kaarel Kaljurand, and Fabio Rinaldi. TX Task: Automatic detection of focus organisms in biomedical publications. In *BioNLP*, 2009.

[6] Roser Morante and Walter Daelemans. Learning the scope of hedge cues in biomedical texts. In *BioNLP*, 2009.

[7] Danielle Mowery, Henk Harkema, John Dowling, Jonathan Lustgarten, and Wendy Chapman. Distinguishing historical from current problems in clinical reports – Which textual features help? In *BioNLP*, 2009.

[8] Sampo Pyysalo, Tomoko Ohta, Jin-Dong Kim, and Jun'ichi Tsujii. Static relations: a piece in the biomedical information extraction puzzle. In *BioNLP*, 2009.

[9] Mark Stevenson, Yikun Guo, Abdulaziz Alamri, and Robert Gaizauskas. Disambiguation of biomedical abbreviations. In *BioNLP*, 2009.

[10] He Tan and Patrick Lambrix. Selecting an ontology for biomedical text mining. In *BioNLP*, 2009.

[11] Wern Wong, David Martinez, and Lawrence Cavedon. Extraction of named entities from tables in gene mutation literature. In *BioNLP*, 2009.

[12] Rong Xu, Alexander A. Morgan, Amar Das, and Alan Garber. Investigation of unsupervised pattern learning techniques for bootstrap construction of a medical treatment lexicon. In *BioNLP*, 2009.

Organizers:

Kevin Bretonnel Cohen, Center for Computational Pharmacology, University of Colorado School of Medicine and The MITRE Corporation
Dina Demner-Fushman, Lister Hill National Center for Biomedical Communications, US National Library of Medicine
Sophia Ananiadou, University of Manchester and UK National Centre for Text Mining
John Pestian, Computational Medicine Center, University of Cincinnati, Cincinnati Children's Hospital Medical Center
Jun'ichi Tsujii, University of Tokyo and UK National Centre for Text Mining
Bonnie Webber, University of Edinburgh

Program Committee:

Alan Aronson, LHNCBC, US National Library of Medicine
Catherine Blake, University of North Carolina
Olivier Bodenreider, LHNCBC, US National Library of Medicine
Bob Carpenter, Alias-i
Wendy Chapman, University of Pittsburgh
Aaron Cohen, Oregon Health and Science University
Nigel Collier, National Institute of Informatics, Tokyo
Noemie Elhadad, Columbia University
Marcelo Fiszman, US National Library of Medicine
Carol Friedman, Columbia College of Physicians and Surgeons
Jin-Dong Kim, University of Tokyo
Marc Light, Thomson
Zhiyong Lu, NCBI, US National Library of Medicine
Aurelie Neveol, LHNCBC, US National Library of Medicine
Serguei Pakhomov, University of Minnesota
Thomas Rindflesch, LHNCBC, US National Library of Medicine
Daniel Rubin, Stanford University
Hagit Shatkay, Queen's University, Canada
Larry Smith, NCBI, US National Library of Medicine
Yuka Tateisi, University of Tokyo
Yoshimasa Tsuruoka, University of Manchester
Alfonso Valencia, Centro Nacional de Biotecnologia
Karin Verspoor, Center for Computational Pharmacology, University of Colorado School of Medicine
Peter White, Children's Hospital of Philadelphia
W. John Wilbur, NCBI, US National Library of Medicine
Limsoon Wong, National University of Singapore
Hong Yu, University of Wisconsin
Pierre Zweigenbaum, LIMSI

Table of Contents

Conference Program

Thursday, June 4, 2009

9:00–9:10 Opening Remarks

Session 1: Paper presentations

9:10–9:35 *Static Relations: a Piece in the Biomedical Information Extraction Puzzle*
Sampo Pyysalo, Tomoko Ohta, Jin-Dong Kim and Jun'ichi Tsujii

9:35–10:00 *Distinguishing Historical from Current Problems in Clinical Reports – Which Textual Features Help?*
Danielle Mowery, Henk Harkema, John Dowling, Jonathan Lustgarten and Wendy Chapman

10:00–10:25 *ONYX: A System for the Semantic Analysis of Clinical Text*
Lee Christensen, Henk Harkema, Peter Haug, Jeannie Irwin and Wendy Chapman

10:30–11:00 morning break

11:00–11:25 *Learning the Scope of Hedge Cues in Biomedical Texts*
Roser Morante and Walter Daelemans

11:25–11:50 *How Feasible and Robust is the Automatic Extraction of Gene Regulation Events? A Cross-Method Evaluation under Lab and Real-Life Conditions*
Udo Hahn, Katrin Tomanek, Ekaterina Buyko, Jung-jae Kim and Dietrich Rebholz-Schuhmann

11:50–12:15 *Extraction of Named Entities from Tables in Gene Mutation Literature*
Wern Wong, David Martinez and Lawrence Cavedon

12:15–10:40 *Selecting an Ontology for Biomedical Text Mining*
He Tan and Patrick Lambrix

12:40–2:00 lunch break

2:00–2:30 Invited Talk

2:35–3:00 *Investigation of Unsupervised Pattern Learning Techniques for Bootstrap Construction of a Medical Treatment Lexicon*
Rong Xu, Alexander A. Morgan, Amar Das and Alan Garber

Thursday, June 4, 2009 (continued)

3:00–3:25 *Disambiguation of Biomedical Abbreviations*
Mark Stevenson, Yikun Guo, Abdulaziz Alamri and Robert Gaizauskas

3:30–4:00 afternoon break

4:00–4:25 *TX Task: Automatic Detection of Focus Organisms in Biomedical Publications*
Thomas Kappeler, Kaarel Kaljurand and Fabio Rinaldi

4:25–4:50 *Exploring Graph Structure for Detection of Reliability Zones within Synonym Resources: Experiment with the Gene Ontology*
Thierry Hamon and Natalia Grabar

4:50–5:15 *Towards Automatic Generation of Gene Summary*
Feng Jin, Minlie Huang, Zhiyong Lu and Xiaoyan Zhu

Session 2: Poster presentations

5:00–6:00 *Incorporating GENETAG-style annotation to GENIA corpus*
Tomoko Ohta, Jin-Dong Kim, Sampo Pyysalo, Yue Wang and Jun'ichi Tsujii

5:00–6:00 *User-Driven Development of Text Mining Resources for Cancer Risk Assessment*
Lin Sun, Anna Korhonen, Ilona Silins and Ulla Stenius

5:00–6:00 *Transforming Controlled Natural Language Biomedical Queries into Answer Set Programs*
Esra Erdem and Reyyan Yeniterzi

5:00–6:00 *Incorporating Syntactic Dependency Information towards Improved Coding of Lengthy Medical Concepts in Clinical Reports*
Vijayaraghavan Bashyam and Ricky K Taira

5:00–6:00 *Identifying Interaction Sentences from Biological Literature Using Automatically Extracted Patterns*
Haibin Liu, Christian Blouin and Vlado Keselj

5:00–6:00 *Using Hedges to Enhance a Disease Outbreak Report Text Mining System*
Mike Conway, Son Doan and Nigel Collier

5:00–6:00 *Exploring Two Biomedical Text Genres for Disease Recognition*
Aurelie Neveol, Won Kim, W. John Wilbur and Zhiyong Lu

Thursday, June 4, 2009 (continued)

5:00–6:00 *Towards Retrieving Relevant Information for Answering Clinical Comparison Questions*
Annette Leonhard

5:00–6:00 *Bridging the Gap between Domain-Oriented and Linguistically-Oriented Semantics*
Sumire Uematsu, Jin-Dong Kim and Jun'ichi Tsujii

5:00–6:00 *Evaluation of the Clinical Question Answering Presentation*
Yong-Gang Cao, John Ely, Lamont Antieau and Hong Yu

5:00–6:00 *Clustering Semantic Spaces of Suicide Notes and Newsgroups Articles.*
Pawel Matykiewicz, Wlodzislaw Duch and John Pestian

5:00–6:00 *TEXT2TABLE: Medical Text Summarization System Based on Named Entity Recognition and Modality Identification*
Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Mashuichi and Kazuhiko Ohe

5:00–6:00 *Semantic Annotation of Papers: Interface & Enrichment Tool (SAPIENT)*
Maria Liakata, Claire Q and Larisa N. Soldatova

Static Relations: a Piece in the Biomedical Information Extraction Puzzle

Sampo Pyysalo* **Tomoko Ohta*** **Jin-Dong Kim*** **Jun'ichi Tsujii*†‡**
*Department of Computer Science, University of Tokyo, Tokyo, Japan
†School of Computer Science, University of Manchester, Manchester, UK
‡National Centre for Text Mining, University of Manchester, Manchester, UK
{smp,okap,jdkim,tsujii}@is.s.u-tokyo.ac.jp

Abstract

We propose a static relation extraction task to complement biomedical information extraction approaches. We argue that static relations such as part-whole are implicitly involved in many common extraction settings, define a task setting making them explicit, and discuss their integration into previously proposed tasks and extraction methods. We further identify a specific static relation extraction task motivated by the BioNLP'09 shared task on event extraction, introduce an annotated corpus for the task, and demonstrate the feasibility of the task by experiments showing that the defined relations can be reliably extracted. The task setting and corpus can serve to support several forms of domain information extraction.

1 Introduction

Relation Extraction (RE) is a key task in biomedical Information Extraction (IE). The automatic detection of relevant types of relations — for various definitions of relevant — between entities has been one of the primary focus points for significant domain research efforts over the past decade, and a substantial number of biomedical RE methods and annotated corpora have been published (Zweigenbaum et al., 2007). Motivated by the needs of biologists and e.g. database curation efforts, most domain RE efforts target relations involving biologically relevant changes in the involved entities, commonly to the complete exclusion of static relations. However, static relations such as entity membership in a family and one entity being a part of another are not only relevant IE targets in themselves but can also play an important supporting role in IE systems not primarily targeting them.

In this paper, we investigate the role of static relations in causal RE and event extraction. Here, we use *relation extraction* in the MUC and ACE (Sundheim, 1995; Doddington et al., 2004) sense to refer to the task of extracting binary relations, ordered pairs of entities, where both participating entities must be specified and their roles (agent, patient, etc.) are fixed by the relation. By contrast, *event extraction* is understood to involve events (things that happen) and representations where the number and roles of participants may vary more freely. We refer to relations where one one entity causes another to change as *causal relations*; typical domain examples are *phosphorylation* and *activation*. Static relations, by contrast, hold between two entities without implication of change or causality: examples from the ACE IE task include Physical.Located and Part-Whole.Artifact.

2 Task definition

In the following, we argue that static relations are relevant to much of current biomedical IE work, present a task setting making these relations explicit, and discuss applications of static relation annotation and extraction methods.

2.1 Named entity-driven IE and static relations

Named entities (NEs) provide a simple anchor connecting text to entities in the real world and thus a natural starting point for IE. Named entity recognition (NER) is well studied and several biomed-

Proceedings of the Workshop on BioNLP, pages 1–9,
Boulder, Colorado, June 2009.

ical NER systems are available (see e.g. (Wilbur et al., 2007; Leaman and Gonzalez, 2008)), and most domain IE approaches are NE-driven: a typical way to cast the RE task is as deciding for each pair of co-occurring NEs whether a relevant relation is stated for them in context. Like the previous LLL and BioCreative2-PPI relation extraction tasks (Nédellec, 2005; Krallinger et al., 2007), the BioNLP'09 shared task on event extraction (Kim et al., 2009) similarly proceeds from NEs, requiring participants to detect events and determine the roles given NEs play in them.

Any domain IE approach targeting nontrivial causal NE relations or events necessarily involves decisions relating to static relations. Consider, for example, the decision whether to extract a relation between NE_1 and NE_2 in the following cases (*affects* should here be understood as a placeholder for any relevant statement of causal relation):

1) NE_1 *affects* NE_2 *gene*
2) NE_1 *affects* NE_2 *promoter*
3) NE_1 *affects* NE_2 *mutant*
4) NE_1 *affects* NE_2 *antibody*
5) NE_1 *affects* NE_2 *activator*

The decision here depends on the interpretation of the noun compounds (NCs) NE_2 *gene*, NE_2 *promoter*, etc. Depending on the IE setting, one might, for example, judge that statements (1)–(3) justify the extraction of an (NE_1, NE_2) relation, while (4) and (5) do not. This question is rarely formalized as a separate (sub)task in domain studies, and methods targeting e.g. the LLL, BioCreative2-PPI and BioNLP'09 shared task relations and events must learn to resolve this question together with the separate issue of which words and syntactic structures express relevant causal relations.

2.2 Task setting

The relation extraction problems represented by examples (1)–(5) above are closely related to the well-studied issue of NC semantics. However, the problem extends past simple binary NCs to include judgments on the relations of arbitrary base NPs (nouns with premodifiers) to contained NEs,

NE_1 *affects truncated* NE_2

NE_1 *affects* NE_2/NE_3 *complexes*

NE_1 *affects* NE_2*-dependent phosphatase*

and further to relations of NPs with NEs that are syntactically less immediately attached:

NE_1 *affects first exon of* NE_2

NE_1 *affects an element in the* NE_2 *promoter*

NE_1 *affects members of the immediate-early activation genes family such as* NE_2

The problem thus encompasses also more general relations between nominals.

While these different cases could also be studied as separate tasks, in the current IE context they can be seen as presenting a continuum of different syntactic realizations of similar relations that also carry the same implications for further processing. We propose to treat them together, formulating the specific task studied in this paper as follows:

> Given: named entity NE and another entity E with their context in text,
> Determine: whether there is a relevant static relation R(NE, E) and its type.

Here, relevant relations are defined as those that justify an inference of some role for the NE in causal relations/events involving E. Additionally, the level of granularity chosen for typing is chosen according to the need to determine the role of the NE in the relations/events. These choices are intentionally dependent on the IE context: we do not expect to be able to formulate a universally accepted set of relevance criteria or relations. Our choice of relation scope and types here follows the perspective of a currently highly relevant IE problem, the BioNLP'09 shared task on event extraction. We aim to recognize a set of relations sufficient to capture the relevant relationships of the NEs provided as given information in the shared task (all of protein/gene/RNA type) and the terms annotated in the GENIA Event corpus (Kim et al., 2008) as participants in events.

We note that this task setting excludes the recognition of candidate NEs and other entities. The assumption that they are given is analogous to the common NE-NE causal relation extraction setting. Further, requiring their recognition would, in our view, unnecessarily complicate the task with aspects of NER and NP chunking, well-studied separate tasks.

We next sketch a formulation of an causal relation/event extraction task incorporating static relations and briefly present one possible way in which

static relation extraction could be applied in IE settings not explicitly targeting such relations.

2.3 Applications of static relations

In the following, we assume that NEs are detected in a prior processing step. Consider, then, the task of extracting relevant information from the following sentence:

NE_1 *is a subunit of the complex that inhibits the expression of mutant forms of* NE_2

An example causal relation extraction target here could be

Inhibit(NE_1,NE_2)

while an event extraction task might aim to recognize the events

E1:Expression(NE_2)
E2:Inhibit(NE_1, E1)

An IE system directly targeting either representation will need to simultaneously address issues relating to the causal statements and static relations. Static relation annotation makes this explicit (square brackets are used to mark non-NE entities):

Part-Whole.Component-Object(NE_1, [complex])
Variant(NE_2, [mutant forms])

This type of static relation detection as prior step to causal relation or event extraction could be applied in at least two different ways: primarily *augmenting* the extracted information, or alternatively *assisting* in the extraction of the information considered above. Assuming the successful extraction of the above static relations, the input can be reformulated as

NE_1 *is a subunit of the [complex] that inhibits the expression of [mutant forms] of* NE_2

Then, under the augmented extraction model, the causal relation and event extraction targets would be, respectively,

Inhibit([complex],[mutant forms])

and

E1:Expression([mutant forms])
E2:Inhibit([complex], E1)

Taken together with the static relations, this provides a more detailed representation of the information stated in the example sentence. Further, simple rules would suffice to derive the simplified representations involving only the NEs, and such rules would have the further benefit of making explicit which intervening static relations are taken to support the inference that an NE is involved in a stated causal relation or event.

Alternatively, under the assisted extraction model, with the assumption that the static relations are taken to allow the inference that any relation or event holding of the other entities holds for the NEs, the input to the causal relation or event extraction system can be recast as

NE_1 *is a subunit of the* NE'_1 *that inhibits the expression of* NE'_2 *of* NE_2

where NE'_1 and NE'_2 should be understood as aliases for NE_1 and NE_2, respectively. Now, under the causal relation extraction model, each of the (NE_1,NE_2), (NE'_1, NE_2), (NE_1,NE'_2), (NE'_1,NE'_2) pairs can serve as an example of the desired relation, both for the purposes of training and actual extraction (the event extraction case can be treated analogously). By increasing the number of positive cases, this application of information on static relations would be expected to have a positive effect on the performance of the primary causal relation/event extraction method.

While these two alternatives are only rough sketches of possible uses of static relation annotation, we expect either could be developed into a practical implementation. Further, these examples by no means exhaust the possibilities of this class of annotation. As static relation extraction can thus be seen to have multiple potential benefits for both causal relation and event extraction, we believe the efforts to pursue static relations as a separate task and to develop resources specific to this task are justified.

3 Relations

Based on an analysis of the shared task data (see Section 4.1), we recognize the static relations illustrated in Table 1. In the following, we briefly discuss the types and their selection.

Name	Examples
Variant	Bcl-6 gene, IL-1 mRNA, wild-type SHP1, TRADD mutant, human IL-1beta, [cell-surface isoforms] of CD43, phosphorylated CREB protein
PW.Object-Component	IL-6 promoter, GR N-terminal transactivation domain, SAA promoter sequence, proximal IL-2 promoter-enhancer, [transcriptional enhancers] including IFNB
PW.Component-Object	NF-kappa_B1/RelA heterodimer, p65 homodimer, p50-p65 complex, STAT1-containing [DNA-binding complex], [heterodimer] of p50 and p65
PW.Member-Collection	CREB/ATF family, p21ras small GTP binding proteins, [non-heat shock genes] such as IL1B, [cellular genes] including GM-CSF
PW.Place-Area	beta-globin locus

Table 1: Relations. In examples, NEs are underlined and square brackets are used to mark the extent of non-NE entities that do not span the entire example text.

3.1 Selection criteria

Relations could be recognized and split into different types at a number of different granularities. Motivated by practical IE applications, we aimed to define a static relation extraction subtask that fits naturally into existing IE frameworks and to create annotation that supplements existing annotation and avoids overlap in annotated information. The practical goals also motivate our aim to recognize a minimal set of different relation types that can satisfy other goals, fewer distinctions implying an easier task and more reliable extraction.

To decide whether to use a single relation type or introduce several subtypes to annotate a given set of cases, we aimed to introduce coherent relation types, each implying consistent further processing. More specifically, we required that each relation R(*NE*, entity) must uniquely and consistently define the relation and roles of the participants, and that in the relevant IE context the relation alone is sufficient to decide how to interpret the role of the *NE* in other relations/events. Specific examples are given in the introduction of the chosen relation types below.

In the following, we follow in part the relation taxonomy and relation definitions of (Winston et al., 1987). However, we recognize that there is no clear agreement on how to subdivide these relations and do not suggest this to be the only appropriate choice.

3.2 Part-whole relations

Part-whole, or meronymic, relations are, not surprisingly, the most common class of static relations in our data: a single generic Part-Whole relation could capture more than half of the relevant relations in the corpus. However, although the relations between the NE and entity in, for example, *[complex] containing* NE and *[site] in* NE are both types of Part-Whole (below PW) relations, the roles of participants are not consistently defined: in PW(*NE*, [site]) the entity is a component of the *NE*, while in PW(*NE*, [complex]) the roles are reversed. We thus recognize separate PW.Object-Component and PW.Component-Object relations. By contrast, while the relation between a *NE* representing a gene and a site on that gene is is arguably different from the relation between a protein *NE* and a site on the protein, we do not distinguish these relations as the annotation would duplicate information available in as part of the entity typing in the corpus and would further imply a static relation extraction task that incorporates aspects of NE recognition.

Also frequent in the data are relations such as that between a protein and a protein family it belongs to. While many cases are clearly identifiable as PW.Member-Collection relations, others could alternatively be analysed as Class-Member. As in our context the relations in e.g. *P, a member of the [type F protein family]* and *P, a [type F protein]* imply the same processing, we will apply the PW.Member-Collection label to both, as well as to *ad hoc* collections such as *[cellular genes] such as* NE, even if this requires a somewhat relaxed interpretation of the relation label. Finally, there are a few cases in our data (e.g. NE *locus*) that we view as instances of the PW.Place-Area relation.

3.3 Variant relations

To avoid unnecessary division of relations that imply in our context similar interpretation and processing, we define a task-specific Variant relation that

encompasses a set of possible relation types holding between an NE and its variants along multiple different axes. One significant class of cases annotated as Variant includes expressions such as NE *gene* and NE *protein*, under the interpretation that *NE* refers to the abstract information that is "realized" as either DNA, RNA or protein form, and the entity to one of these realizations (for alternative interpretations, see e.g. (Rosario and Hearst, 2001; Heimonen et al., 2008)).

The Variant relation is also used to annotate *NE*-entity relations where the entity expresses a different state of the *NE*, such as a phosphorylated or mutated state. While each possible post-translational modification, for example, could alternatively be assigned a specific relation type, in the present IE context these would only increase the difficulty of the task without increasing the applicability of the resulting annotation.

3.4 Other/Out annotation

We apply a catch-all category, Other/Out, for annotating candidate (NE, entity) pairs between which there is no relevant static relation. This label is thus applied to a number of quite different cases: causal relations, both implied (e.g. NE *receptors*, NE *response element*) and explicitly stated (NE *binds the [site]*), relations where the entity is considered too far removed from the NE to support reliable inference of a role for the NE in causal relations/events involving the entity (e.g. *[antibodies] for* NE), and cases where no relation is stated (e.g. NE *and other [proteins]*). The diversity of this generic category of irrelevant cases is a necessary consequence of the aim to avoid annotation involving decisions directly relating to other tasks by creating distinctions between e.g. *causal* and *no relation*.

3.5 Sufficiency of the setting and relation types

We have cast the static relation extraction task as always involving an NE, which in the present context is further always of a protein, gene or RNA type. This restriction considerably simplifies the task conceptually and reduces annotation effort as well as expected extraction difficulty, as the type of only one of the entities involved in the relation can vary significantly. However, it is not obvious that the restriction allows coherent relations types to be defined. If the corpus contained frequent cases where the stated relationship of the NE to the entity involved different types of relevant relations (e.g. collections of parts of an NE), it would be necessary to either recognized "mixed" or combined relations or extend the task to include general entity-entity relations.

Interestingly, during annotation we encountered only two cases (less than 0.1% of those annotated) involving two of the recognized relation types at once: *mutant* NE *promoter* and *5' truncation mutants of the* NE *promoter*[1]. While this result is likely affected by a number of complex factors (annotation criteria, NE and entity types, granularity of relations, etc.), we find the outcome — which was neither planned for nor forced on the data — a very encouraging sign of the sufficiency of the task setting for this and related domain IE tasks.

4 Data

We created the data set by building on the annotation of the GENIA Event corpus (Kim et al., 2008), making use of the rich set of annotations already contained in the corpus: term annotation for NEs and other entities (Ohta et al., 2002), annotation of events between these terms, and treebank structure closely following the Penn Treebank scheme (Tateisi et al., 2005).

4.1 Annotation

The existing GENIA annotations served as the basis of the new annotation. We initially selected as candidates entities annotated as participating in events considered in the BioNLP'09 shared task.

As the term annotation includes nesting of entities, NEs contained within these relevant entities were used as the starting point for the annotation. We first performed a preliminary study of the relevant static relations occurring between the entities and NEs occurring within them to determine the set of relations to annotate. Next, all unique cases where a selected entity contained an NE were annotated with the appropriate relation based on the contained text of the entity, with the text of the contained NE normalized away. For the present study, we excluded from consideration cases where the annota-

[1] To resolve these cases, we simply ignored the implied Variant relation.

tion indicated simple aliasing (e.g. *[CREB/ATF]*), a relation irrelevant to our purpose and found in the selected data only due to the annotation specifying one entity but two NEs in these cases. In this step, 830 unique cases representing a total of 1601 entities containing NEs were annotated.

The nesting structure of the term annotation does not, however, capture all relevant static relations: the term annotation scheme disallows discontinuous terms and annotation of terms with structure more complex than base NPs. Thus, the possible relations of NEs to entities to which they were connected e.g. by a prepositional phrase cannot be directly derived from the existing annotation. As an example, the nesting in *[*NE *region]* directly suggest the existence of a relation, while no such connection appears in *[region] of* NE. To annotate relations for entities for which the term annotation does not identify a candidate related NE, it is necessary to form (NE, entity) pairs with co-occurring NEs. Even when the candidate NEs were restricted to those occurring in the same sentence, the number of such pairs in the corpus was over 17,000, beyond the scope of what could be annotated as part of this effort. Further, as the great majority of co-occurring (NE, entity) pairs will have no relevant static relation, we used heuristics to increase the proportion of relevant and near-miss cases in the annotated data.

We first converted the gold standard annotation of the GENIA treebank (Tateisi et al., 2005) into a dependency representation using the Stanford parser tools (de Marneffe et al., 2006) and then determined the shortest paths in the dependency analyses connecting each relevant entity with each NE. The (NE, entity) pairs were then ordered according to the length of these paths, on the assumption that syntactically more closely related entities are more likely to have a relevant static relation. Annotation then proceeded on the ordered list of pairs. During the annotation, we further developed more ordering heuristics, such as giving higher ranking to candidate pairs connected by a path that contains a subpath known to connect pairs with relevant relations. Such known paths were first derived from the BioInfer static relation annotation (Pyysalo et al., 2007) and later extracted from previously annotated cases. In this annotation process, judgments were performed with reference to the full sentence context. In total, 1100 cases were annotated in this way. All stages of the annotation process involved only lists formatted as simple text files for markup and custom-written software for processing.

	Annotated instances		
Relation	cont.	nonc.	total
PW.Object-Component	394	133	527
PW.Component-Object	299	44	343
Variant	253	20	273
PW.Member-Collection	25	124	149
PW.Place-Area	4	1	5
Other/Out	626	778	1404
total	1601	1100	2701

Table 2: Statistics for annotated data. Number of instances given separately for relations annotated between entities with contained (cont.) and non-contained (nonc.) NEs.

Table 2 contains statistics for the annotated data, showing separately the number of annotated relations of entities to contained and non-contained NEs. There are interesting differences in the relation type distribution between these two categories, reflecting the different ways in which relations are typically stated. This difference in distribution suggests that it may be beneficial to give the two cases different treatment in extraction.

4.2 Representation

For simplicitly of use, we provide the annotated data in two equivalent representations: a simple inline XML format and a standoff format. The XML format closely resembles the representation used for the SemEval-2007 Semantic Relations between Nominals task (Girju et al., 2007). Here, each NE-Entity pair is given its own entry with its sentence context in which only the pair is marked. In the alternate standoff representation, all entities appearing in each sentence are tagged, and the annotated relations given separately. These representations are easily processed and should be usable with little modification with many existing relation extraction methods.

We further split the data into training, development-test and test sets according to the same division applied in the BioNLP'09 shared task on event extraction. This division allows the dataset to be easily integrated into settings using the shared task data, combining static relation and event extraction approaches.

5 Experiments

The selected task setting and representation form a natural basis for two alternative classification problems: a binary classification problem for detecting the presence of any relevant relation, and a multiclass classification problem where the correct relation type must also be determined. In the following, we describe experiments using the dataset in these two settings. While we apply a state-of-the-art machine learning method and a fairly expressive representation, the aim of the experiments is only to determine the relative difficulty of the relation extraction task and to establish a moderately competitive baseline result for the newly created dataset.

We use a linear Support Vector Machine (SVM) classifier (Chang and Lin, 2001) with N-gram features defined over token sequences delimited by the beginning and end of the entity and the position of the NE. The NE is treated as a single token and its text content blinded from the classifier to avoid overfitting on specific names. Features are generated from two sequences of tokens: those inside the entity and, when the NE is not contained in the entity, those between the entity and the NE (inclusive of the entity and NE at the sequence boundaries). In preliminary experiments on the development test set we found no clear benefit from including N-gram features extracted from a broader context, supporting an assumption that the problem can be mostly addressed on the basis of local features. By contrast, preliminary experiments supported the use of the simple Porter algorithm (Porter, 1980) for stemming, the inclusion of uni-, bi- and trigram features, and normalization of the feature vectors to unit length; these were adopted for the final experiment. The SVM regularization parameter was optimized using a sparse search with evaluation on the development test set.

We first reduced the annotated data into a binary classification problem with the Other/Out class representing negative (irrelevant) and the other relations positive (relevant) cases. The results for this experiment were very encouraging, giving both a high classification accuracy of 86.8% and an F-score of 84.1%. The test set contains 179 positive and 269 negative cases, giving a majority baseline accuracy of 60.0% and an all-true baseline F-score of 57.1%. The classifier notably and statistically significantly (McNemar's test, $p < 0.01$) outperforms these simple baselines. We then performed a separate multiclass classification experiment, predicting the specific type of the relation, also including the Other/Out type. In this experiment, accuracy remained relatively high at 81.9%, while per-class precision and recall results (considering each class in turn positive and all others negative, see Table 3) indicate some remaining challenges. The results vary somewhat predictably with the number of examples per relation type (Table 2): while PW.Object-Component relations can be predicted at high precision and fair recall, performance for PW.Member-Collection relations falls behind expectations for a local relation extraction problem.

	P	R	F
Relevant	81.2	87.2	84.1
PW.Object-Component	94.2	75.4	83.8
PW.Component-Object	60.0	71.2	65.1
Variant	88.0	57.9	69.8
PW.Member-Collection	54.5	37.5	44.4

Table 3: Classification results with (P)recision, (R)ecall and (F)-score for the binary Relevant/Irrelevant experiment and classwise results for the relevant classes (PW.Place-Area excluded for lack of data).

To briefly relate these results to domain causal RE results, we note that the recently proposed state-of-the-art method of (Airola et al., 2008) was reported to achieve F-scores ranging between 56.4–76.8% on five different causal RE corpora in a binary classification setting. As our relatively simple method achieves a notably higher 84.1% F-score at the binary static RE task, we can conclude that this static RE task is not as difficult as the causal RE tasks. This is encouraging for the prospects of static RE in support of domain causal RE and event extraction.

6 Related work

Relations of types that we have here termed static have figured prominently in the MUC and ACE series of events that have largely defined the "general domain" IE research program (Sundheim, 1995; Doddington et al., 2004). In this line of research, event-type annotation is used (as the name implies) to capture events, defined as "*[...] something that happens [...] [that] can frequently be described as a*

change of state" (LDC, 2005) and relation-type annotation is applied for relevant non-causal relationships. General static relations have been studied extensively also in broader, non-IE contexts (see e.g. (Girju et al., 2007)).

In the biomedical domain, static relations have received relatively little attention. Domain noun compound semantics, including static relations, have been considered in studies by (Rosario and Hearst, 2001) and (Nakov et al., 2005), but in IE settings static relations tend to appear only implicitly, as in the RelEx causal RE system of (Fundel et al., 2007), or through the causal relations they imply: for example, in the AIMed corpus (Bunescu et al., 2005) statements such as NE_1/NE_2 *complex* are annotated as a binding relation between the two NEs, not Part-Whole relations with the broader entity. By contrast, there has been considerable focus on the extraction of "things that happen," dominantly making use of relation-type corpus annotation and extraction approaches: a study of five corpora containing primarily causal relation annotation is found in (Pyysalo et al., 2008); more complete lists of domain corpora are maintained by Kevin Cohen[2] and Jörg Hakenberg[3]. For a thorough review of recent work in domain RE, we refer to (Zweigenbaum et al., 2007).

BioInfer (Pyysalo et al., 2007), to the best of our knowledge the first domain corpus to include event-type annotation, also includes annotation for a set of static relation types. The design of the BioInfer corpus and relationship type ontology as well as work applying the corpus in jointly targeting event extraction and static relation extraction (Heimonen et al., 2008; Björne et al., 2008) have considerably influenced the present study. A key difference in focus is that BioInfer primarily targets NE-NE relations, while our concern here has been the relations of NEs with other, non-NE entities, specifically focusing on the requirements of the BioNLP'09 shared task. A class of static relations, connecting Mutants and Fragments with their parent proteins, is annotated in the recently introduced ITI TXM corpora (Alex et al., 2008). While somewhat limited in the scope of static relations, this annotation covers an extensive number of instances, over 20,000, and could likely support the development of high-reliability methods for the extraction extraction of these specific static relations. As discussed in detail in Section 4.1, previously published versions of the GENIA corpus (Kim et al., 2008) contain NE, term and event annotation, but no static relations have been annotated in GENIA prior to this effort.

While previously introduced corpora thus cover aspects of the annotation required to address the static relation extraction task considered in this paper, we are not aware of previously published resources that would address this task specifically or contain annotation supporting the entire task as envisioned here.

7 Conclusions and future work

In this paper, we have argued for a position for static relations in biomedical domain IE, specifically advancing the subtask of extracting static relations between named entities and other entities appearing in their context. We explored this subtask in the specific IE context of the BioNLP'09 shared task on event extraction, identifying possible instances of static relations relevant to the task setting. We then studied these instances of detail, defining a minimal set of basic static relations argued to be sufficient to support the type of IE envisioned in the shared task. We annotated 2701 instances of candidate static relations, creating the first domain corpus of static relations explicitly designed to support IE, and performed experiments demonstrating that the static relation extraction task can be performed accurately, yet retains challenges for future work. The newly annotated corpus is publicly available at `www-tsujii.is.s.u-tokyo.ac.jp/GENIA` to encourage further research on this task.

Acknowledgments

Discussions with members of the BioInfer group were central for developing many of the ideas presented here. We are grateful for the efforts of Maki Niihori in producing supporting annotation applied in this work. This work was partially supported by Grant-in-Aid for Specially Promoted Research (Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan), and Genome Network Project (MEXT, Japan).

[2] http://compbio.uchsc.edu/ccp/corpora/obtaining.shtml

[3] http://www2.informatik.hu-berlin.de/∼hakenber/links/benchmarks.html

References

Antti Airola, Sampo Pyysalo, Jari Bjorne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, 9(Suppl 11):S2.

Bea Alex, Claire Grover, Barry Haddow, Mijail Kabadjov, Ewan Klein, Michael Matthews, Stuart Roebuck, Richard Tobin, and Xinglong Wang. 2008. The ITI TXM corpora: Tissue expressions and protein-protein interactions. In *Proceedings of LREC'08*.

Jari Björne, Sampo Pyysalo, Filip Ginter, and Tapio Salakoski. 2008. How complex are complex protein-protein interactions? In *Proceedings SMBM'08*.

Razvan C Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun Kumar Ramani, and Yuk Wah Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The Automatic Content Extraction (ACE) program: Tasks, data, and evaluation. In *Proceedings of LREC'04*, pages 837–840.

Katrin Fundel, Robert Kuffner, and Ralf Zimmer. 2007. RelEx–Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of SemEval'07*, pages 13–18.

Juho Heimonen, Sampo Pyysalo, Filip Ginter, and Tapio Salakoski. 2008. Complex-to-pairwise mapping of biological relationships using a semantic network representation. In *Proceedings of SMBM'08*.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(10).

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of bionlp'09 shared task on event extraction. In *Proceedings of BioNLP'09*.

Martin Krallinger, Florian Leitner, and Alfonso Valencia. 2007. Assessment of the second BioCreative PPI task: Automatic extraction of protein-protein interactions. In *Proceedings of BioCreative II*, pages 41–54.

LDC. 2005. ACE (automatic content extraction) english annotation guidelines for events. Technical report, Linguistic Data Consortium.

R. Leaman and G. Gonzalez. 2008. Banner: An executable survey of advances in biomedical named entity recognition. In *Proceedings of PSB'08*, pages 652–663.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC'06*, pages 449–454.

Preslav Nakov, Ariel Schwartz, Brian Wolf, and Marti Hearst. 2005. Scaling up bionlp: Application of a text annotation architecture to noun compound bracketing. In *Proceedings of BioLINK'05*.

Claire Nédellec. 2005. Learning language in logic - genic interaction extraction challenge. In *Proceedings of LLL'05*.

Tomoko Ohta, Yuka Tateisi, Hideki Mima, and Jun'ichi Tsujii. 2002. GENIA corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the Human Language Technology Conference (HLT'02)*, pages 73–77.

M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(2):130–137.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. 2007. BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(50).

Sampo Pyysalo, Antti Airola, Juho Heimonen, and Jari Björne. 2008. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, 9(Suppl. 3):S6.

Barbara Rosario and Marti Hearst. 2001. Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. In *Proceedings of EMLNP'01*, pages 82–90.

Beth M. Sundheim. 1995. Overview of results of the MUC-6 evaluation. In *Proceedings of MUC-6*, pages 13–31.

Yuka Tateisi, Akane Yakushiji, Tomoko Ohta, and Jun'ichi Tsujii. 2005. Syntax annotation for the GENIA corpus. In *Proceedings of IJCNLP'05*, pages 222–227.

John Wilbur, Larry Smith, and Lorrie Tanabe. 2007. Biocreative 2 gene mention task. In *Proceedings of BioCreative 2*, pages 7–16.

Morton E. Winston, Roger Chaffin, and Douglas Herrmann. 1987. A taxonomy of part-whole relations. *Cognitive Science*, 11.

Pierre Zweigenbaum, Dina Demner-Fushman, Hong Yu, and Kevin B. Cohen. 2007. Frontiers of biomedical text mining: Current progress. *Briefings in Bioinformatics*.

Distinguishing Historical from Current Problems in Clinical Reports—Which Textual Features Help?

Danielle L. Mowery MS, Henk Harkema PhD, John N. Dowling MS MD,
Jonathan L. Lustgarten PhD, Wendy W. Chapman PhD
Department of Biomedical Informatics
University of Pittsburgh, Pittsburgh, Pa 15260, USA
dlm31@pitt.edu, heh23@pitt.edu, dowling@pitt.edu, jll47@pitt.edu, wec6@pitt.edu

Abstract

Determining whether a condition is historical or recent is important for accurate results in biomedicine. In this paper, we investigate four types of information found in clinical text that might be used to make this distinction. We conducted a descriptive, exploratory study using annotation on clinical reports to determine whether this temporal information is useful for classifying conditions as historical or recent. Our initial results suggest that few of these feature values can be used to predict temporal classification.

1 Introduction

Clinical applications for decision support, biosurveillance and quality of care assessment depend on patient data described in unstructured, free-text reports. For instance, patient data in emergency department reports contain valuable indicators for biosurveillance applications that may provide early signs and symptoms suggestive of an outbreak. Quality assurance departments can use free-text medical record data to assess adherence to quality care guidelines, such as determining whether an MI patient was given an aspirin within twenty-four hours of arrival. In either application, one must consider how to address the question of time, but each of the applications requires a different level of temporal granularity: the biosurveillance system needs a coarse-grained temporal model that discerns whether the signs and symptoms are historical or recent. In contrast, the quality assurance system needs a fine-grained temporal model to identify the admission event, when (or if) aspirin was given, and the order and duration of time between these events. One important problem in natural language processing is extracting the appropriate temporal granularity for a given task.

Many solutions exist for extracting temporal information, and each is designed to address questions of various degrees of temporal granularity, including determining whether a condition is historical or recent, identifying explicit temporal expressions, and identifying temporal relations among events in text. (Chapman et al., 2007; Zhou et al., 2008; Irvine et al., 2008; Verhagen and Pustejovsky, 2008; Bramsen et al., 2006). We previously extended the NegEx algorithm in ConText, a simple algorithm that relies on lexical cues to determine whether a condition is historical or recent (Chapman et al., 2007). However, ConText performs with moderate recall (76%) and precision (75%) across different report types implying that trigger terms and simple temporal expressions are not sufficient for the task of identifying historical conditions.

In order to extend work in identifying historical conditions, we conducted a detailed annotation study of potentially useful temporal classification features for conditions found in six genres of clinical text. Our three main objectives were: (1) characterize the temporal similarity and differences found in different genres of clinical text; (2) determine which features successfully predict whether a condition is historical, and (3) compare ConText to machine learning classifiers that account for this broader set of temporal features.

2 Temporality in Clinical Text

For several decades, researchers have been studying temporality in clinical records (Zhou and Hripcsak, 2007). Readers use a variety of clues to distinguish temporality from the clinical narrative, and we wanted to identify features from other tem-

Proceedings of the Workshop on BioNLP, pages 10–18,
Boulder, Colorado, June 2009.

poral models that may be useful for determining whether a condition is historical or recent.

There are a number of automated systems for extracting, representing, and reasoning time in a variety of text. One system that emerged from the AQUAINT workshops for temporal modeling of newspaper articles is TARSQI. TARSQI processes events annotated in text by anchoring and ordering them with respect to nearby temporal expressions (Verhagen and Pustejovsky, 2008). A few recent applications, such as TimeText and TN-TIES (Zhou et al., 2008; Irvine et al., 2008), identify medically relevant events from clinical texts and use temporal expressions to order the events. One method attempts to order temporal segments of clinical narratives (Bramsen et al., 2006). One key difference between these previous efforts and our work is that these systems identify all temporal expressions from the text and attempt to order all events. In contrast, our goal is to determine whether a clinical condition is historical or recent, so we focus only on temporal information related to the signs, symptoms, and diseases described in the text. Therefore, we ignore explicit temporal expressions that do not modify clinical conditions. If a condition does not have explicit temporal modifiers, we still attempt to determine the historical status for that condition (e.g., "Denies cough"). In order to improve the ability to determine whether a condition is historical, we carried out this annotation study to identify any useful temporal information related to the clinical conditions in six clinical genres. Building on work in this area, we explored temporal features used in other temporal annotation studies.

TimeML is a well-known standard for complex, temporal annotation. TimeML supports the annotation of events defined as "situations that happen or occur" and temporal expressions such as *dates* and *durations* in order to answer temporal questions about these events and other entities in news text (Saur´ı, et al., 2006). One notable feature of the TimeML schema is its ability to capture verb tense such as *past* or *present* and verb aspect such as *perfective* or *progressing*. We annotated verb tense and aspect in medical text according to the TimeML standard.

Within the medical domain, Zhou et al. (2006) developed an annotation schema used to identify temporal expressions and clinical events. They measured the prevalence of explicit temporal expressions and key medical events like *admission* or *transfer* found in discharge summaries. We used the Zhou categorization scheme to explore temporal expressions and clinical events across genres of reports.

A few NLP systems rely on lexical cues to address time. MediClass is a knowledge-based system that classifies the content of an encounter using both free-text and encoded information from electronic medical records (Hazelhurst et al., 2005). For example, MediClass classifies smoking cessation care delivery events by identifying the status of a smoker as *continued*, *former* or *history* using words like *continues*. ConText, an extension of the NegEx algorithm, temporally classifies conditions as historical, recent, or hypothetical using lexical cues such as *history*, *new*, and *if*, respectively (Chapman et al., 2007). Drawing from these applications, we used state and temporal trigger terms like *active*, *unchanged*, and *history* to capture coarse, temporal information about a condition.

Temporal information may also be implied in the document structure, particularly with regards to the section in which the condition appears. SecTag marks explicit and implicit sections found throughout patient H&P notes (Denny et al., 2008). We adopted some section headers from the SecTag terminology to annotate sections found in reports.

Our long-term goal is to build a robust temporal classifier for information found in clinical text where the output is classification of whether a condition is historical or recent (historical categorization). An important first step in classifying temporality in clinical text is to identify and characterize temporal features found in clinical reports. Specifically, we aim to determine which expressions or features are predictive of historical categorization of clinical conditions in dictated reports.

3 Historical Assignment and Temporal Features

We conducted a descriptive, exploratory study of temporal features found across six genres of clinical reports. We had three goals related to our task of determining whether a clinical condition was historical or recent. First, to develop a temporal classifier that is generalizable across report types, we compared temporality among different genres

of clinical text. Second, to determine which features predict whether a condition is historical or recent, we observed common rules generated by three different rule learners based on manually annotated temporal features we describe in the following section. Finally, we compared the performance of ConText and automated rule learners and assessed which features may improve the ConText algorithm.

Next, we describe the temporal features we assessed for identification of historical signs, symptoms, or diseases, including temporal expressions, lexical cues, verb tense and aspect, and sections.

(1) **Temporal Expressions:** Temporal expressions are time operators like dates (*May 5th 2005*) and durations (*for past two days*), as well as clinical processes related to the encounter (*discharge, transfer*). For each clinical condition, we annotated whether a temporal expression modified it and, if so, the category of temporal expression. We used six major categories from Zhou et al. (2006) including: *Date and Time*, *Relative Date and Time*, *Durations*, *Key Events*, *Fuzzy Time*, and *No Temporal Expression*. These categories also have types. For instance, *Relative Date and Time* has a type *Yesterday, Today* or *Tomorrow*. For the condition in the sentence "The patient had a stroke in ***May 2006***", the temporal expression category is *Date and Time* with type *Date*. Statements without a temporal expression were annotated *No Temporal Expression* with type *N/A*.

(2) **Tense and Aspect:** Tense and aspect define how a verb is situated and related to a particular time. We used TimeML Specification 1.2.1 for standardization of tense and aspect where examples of tense include *Past* or *Present* and aspect may be *Perfective*, *Progressive*, *Both* or *None* as found in Saur´ı, et al. (2006). We annotated the verb that scoped a condition and annotated its tense and aspect. The primary verb may be a predicate adjective integral to interpretation of the condition (Left ventricle ***is enlarged***), a verb preceding the condition (***has*** hypertension), or a verb following a condition (Chest pain ***has resolved***). In "her chest pain ***has resolved,"*** we would mark "has resolved" with tense *Present* and aspect *Perfective*. Statements without verbs (e.g., No murmurs) would be annotated *Null* for both.

(3) **Trigger Terms:** We annotated lexical cues that provide temporal information about a condition. For example, in the statement, "Patient has past ***history*** of diabetes," we would annotate "history" as *Trigger Term: Yes* and would note the exact trigger term.

(4) **Sections:** Sections are "clinically meaningful segments which act independently of the unique narrative" for a patient (Denny et al. 2008). Examples of report sections include *Review of Systems* (Emergency Department), *Findings* (Operative Gastrointestinal and Radiology) and *Discharge Diagnosis* (Emergency Department and Discharge Summary).

We extended Denny's section schema with explicit, report-specific section headers not included in the original terminology. Similar to Denny, we assigned implied sections in which there was an obvious change of topic and paragraph marker. For instance, if the sentence "the patient is allergic to penicillin" followed the *Social History* section, we annotated the section as *Allergies*, even if there was not a section heading for allergies.

4 Methods

4.1 Dataset Generation

We randomly selected seven reports from each of six genres of clinical reports dictated at the University of Pittsburgh Medical Center during 2007 These included Discharge Summaries, Surgical Pathology, Radiology, Echocardiograms, Operative Gastrointestinal, and Emergency Department reports. The dataset ultimately contained 42 clinical reports and 854 conditions. Figure 1 show our annotation process, which was completed in GATE, an open-source framework for building NLP systems (http://gate.ac.uk/). A physician board-certified in internal medicine and infectious diseases annotated all clinical conditions in the set and annotated each condition as either historical or recent. He used a general guideline for annotating a condition as historical if the condition began more than 14 days before the current encounter and as recent if it began or occurred within 14 days or during the current visit. However, the physician was not bound to this definition and ultimately used his own judgment to determine whether a condition was historical.

Provided with pre-annotated clinical conditions and blinded to the historical category, three of the authors annotated the features iteratively in groups of six (one of each report type) using guidelines we

developed for the first two types of temporal features (temporal expressions and trigger terms.) Between iterations, we resolved disagreements through discussion and updated our guidelines. Cohen's kappa for temporal expressions and trigger terms by the final iteration was at 0.66 and 0.69 respectively. Finally, one author annotated sections, verb tense, and aspect. Cases in which assigning the appropriate feature value was unclear were resolved after consultation with one other author-annotator.

4.2 Data Analysis

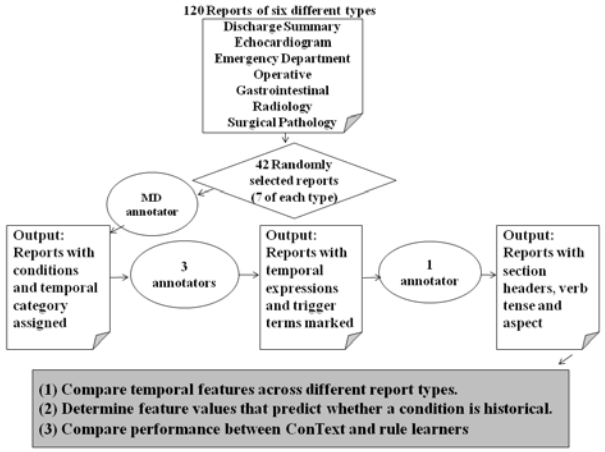


Figure 1. Annotation process for dataset and objectives for evaluation.

We represented each condition as a vector with temporal features and their manually-assigned values as input features for predicting the binary outcome value of historical or recent. We trained three rule learning algorithms to classify each condition as historical or recent: J48 Decision Tree, Ripper, and Rule Learner (RL) (Witten and Frank, 2005; Clearwater and Provost, 1990). Rule learners perform well at classification tasks and provide explicit rules that can be viewed, understood, and potentially implemented in existing rule-based applications. We used Weka 3.5.8, an openly-available machine learning application for prediction modeling, to implement the Decision Tree (J48) and Ripper (JRip) algorithms, and we applied an in house version of RL retrieved from www.dbmi.pitt.edu\probe. For all rule learners, we used the default settings and ran ten-fold cross-validation. The J48 algorithm produces mutually exclusive rules for predicting the outcome value. Thus, two rules cannot cover or apply to any one case. In contrast, both JRip and RL generate non-mutually-exclusive rules for predicting the outcome value. Although J48 and JRip are sensitive to bias in outcome values, RL accounts for skewed distribution of the data.

We also applied ConText to the test cases to classify them as historical or recent. ConText looks for trigger terms and a limited set of temporal expressions within a sentence. Clinical conditions within the scope of the trigger terms are assigned the value indicated by the trigger terms (e.g., historical for the term *history).* Scope extends from the trigger term to the end of the sentence or until the presence of a termination term, such as *presenting.* For instance, in the sentence "***History*** of CHF, ***presenting*** with chest pain," CHF would be annotated as historical.

5 Evaluation

To characterize the different reports types, we established the overall prevalence and proportion of conditions annotated as historical for each clinical report genre. We assessed the prevalence of each feature (temporal expressions, trigger terms, tense and aspect, and sections) by report genre to determine the level of similarity or difference between genres. To determine which features values are predictive of whether a condition is historical or recent, we observed common rules found by more than one rule learning algorithm. Amongst common rules, we identified new rules that could improve the ConText algorithm.

We also measured predictive performance with 95% confidence intervals of the rule learners and ConText by calculating overall accuracy, as well as recall and precision for historical classifications and recall and precision for recent classifications. Table 1 describes equations for the evaluation metrics.

Table 1. Description of evaluation metrics. RLP = rule learner prediction. RS = Reference Standard

	Historical		**Recent**	
	RLP	RS	RLP	RS
True Pos (TP)	Historical	Historical	Recent	Recent
False Pos (FP)	Historical	Recent	Recent	Historical
True Neg (TN)	Recent	Recent	Historical	Historical
False Neg (FN)	Recent	Historical	Historical	Recent

Recall: $\frac{\text{number of TP}}{\text{(number of TP + number of FN)}}$

Precision: $\frac{\text{number of TP}}{\text{(number of TP + number of FP)}}$

Accuracy: $\frac{\text{number of instances correctly classified}}{\text{total number of possible instances}}$

6 Results

Overall, we found 854 conditions of interest across all six report genre. Table 2 illustrates the prevalence of conditions across report genres. Emergency Department reports contained the highest concentration of conditions. Across report genres, 87% of conditions were recent (741 conditions). All conditions were recent in Echocardiograms, in contrast to Surgical Pathology reports in which 68% were recent.

Table 2. Prevalence and count of conditions by temporal category and report genre. DS = Discharge Summary, Echo = Echocardiogram, ED = Emergency Department, GI = Operative Gastrointestinal, RAD = Radiology and SP = Surgical Pathology. (%) = percent; Ct = count.

	Historical		Recent		
Report	(%)	Ct	(%)	Ct	Total Conditions
DS	(19)	38	(81)	158	196
Echo	(0)	0	(100)	199	199
ED	(17)	61	(83)	301	362
GI	(9)	3	(91)	32	35
RAD	(6)	2	(94)	32	34
SP	(32)	9	(68)	19	28
Total Conditions		113		741	854

6.1 Prevalence of Temporal Features

Table 3 shows that most conditions were not modified by a temporal expression or a trigger term. Conditions were modified by a temporal expression in Discharge Summaries more often than in other report genres. Similarly, Surgical Pathology had the highest prevalence of conditions modified by a trigger term. Operative Gastrointestinal and Radiology reports showed the lowest prevalence of both temporal expressions and trigger terms. Neither temporal expressions nor trigger terms occurred in Echocardiograms. Overall, the prevalence of conditions scoped by a verb varied across report types ranging from 46% (Surgical Pathology) to 81% (Echocardiogram).

Table 3. Prevalence of conditions modified by temporal features. All conditions were assigned a section and are thereby excluded. TE = temporal expression; TT = trigger term; V = scoped by verb.

	DS		Echo		ED		GI		RAD		SP	
	(%)	Ct	(%)	Ct	(%)	Ct	(%)	Ct	(%)	Ct	(%)	Ct
TE	(37)	73	(0)	0	(16)	59	(6)	2	(3)	1	(25)	7
TT	(25)	49	(0)	0	(19)	68	(9)	3	(3)	1	(39)	11
V	(70)	138	(81)	161	(62)	223	(69)	24	(76)	26	(46)	13

6.2 Common Rules

Rule learners generated a variety of rules. The J48 Decision Tree algorithm learned 27 rules, six for predicting conditions as historical and the remaining for classifying the condition as recent. The rules predominantly incorporated the trigger term and verb tense and aspect feature values. JRip learned nine rules, eight for classifying the historical temporal category and one 'otherwise' rule for the majority class. The JRip rules most heavily incorporated the section feature. The RL algorithm found 79 rules, 18 of which predict the historical category. Figure 2 illustrates historical rules learned by each rule learner. JRip and RL predicted the following sections alone can be used to predict a condition as historical: *Past Medical History*, *Allergies* and *Social History*. Both J48 and RL learned that trigger terms like *previous*, *known* and *history* predict historical. There was only one common, simple rule for the historical category found amongst all three learners: the trigger term *no change* predicts the historical category. All algorithms learned a number of rules that include two features values; however, none of the compound rules were common amongst all three algorithms.

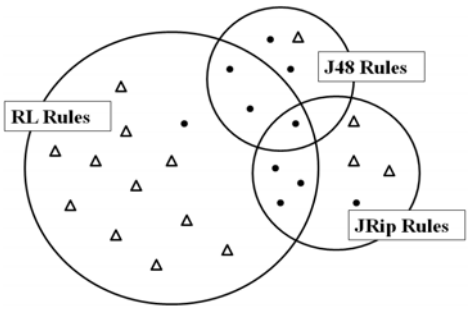


Figure 2. Historical rules learned by each rule learner algorithm. Black dots represent simple rules whereas triangles represent compound rules. Common rules shared by each algorithm occur in the overlapping areas of each circle.

6.3 Predictive Performance

Table 4 shows predictive performance for each rule learner and for ConText. The RL algorithm outperformed all other algorithms in almost all evaluation measures. The RL scores were computed based on classifying the 42 cases (eight historical) for which the algorithm did not make a prediction as recent. ConText and J48, which exclusively relied on trigger terms, had lower recall for the historical category.

All of the rule learners out-performed ConText. JRip and RL showed substantially higher recall for assigning the historical category, which is the most important measure in a comparison with ConText, because ConText assigns the default value of recent unless there is textual evidence to indicate a historical classification. Although the majority class baseline shows high accuracy due to high prevalence of the recent category, all other classifiers show even higher accuracy, achieving fairly high recall and precision for the historical cases while maintaining high performance on the recent category.

Table 4. Performance results with 95% confidence intervals for three rule learners trained on manually annotated features and ConText, which uses automatically generated features. Bolded values do not have overlapping confidence intervals with ConText. MCB = Majority Class Baseline (recent class)

Algorithm	Accuracy (Overall)	Recall (Historical)	Precision (Historical)	Recall (Recent)	Precision (Recent)
ConText	92.4	73.2	70.1	95.3	95.9
	90.8-94.4	70.5-76.4	67.2-73.4	94.1-96.1	94.8-97.5
J48	94.0	**62.8**	**88.8**	**98.8**	94.6
	92.6-95.8	**59.8-66.3**	**86.9-91.1**	**98.3-99.8**	93.3-96.3
JRip	**97.1**	**83.2**	**94.0**	**99.2**	97.5
	96.2-98.5	**80.9-85.9**	**92.6-95.8**	**98.8-100.0**	96.6-98.8
RL	**96.8**	**82.2**	**97.8**	**99.7**	97.5
	95.8-98.2	**79.9-85.0**	**97.0-99.0**	**99.5-100.0**	96.6-98.8
MCB	86.9	--	--	100.0	0.0

7 Discussion

Our study provides a descriptive investigation of temporal features found in clinical text. Our first objective was to characterize the temporal similarities and differences amongst report types. We found that the majority of conditions in all report genres were recent conditions, indicating that a majority class classifier would produce an accuracy of about 87% over our data set. According to the distributions of temporal category by report genre (Table 2), Echocardiograms exclusively describe recent conditions. Operative Gastrointestinal and Radiology reports contain similar proportions of historical conditions (9% and 6%). Echocardiograms appear to be most similar to Radiology reports and Operative Gastrointestinal reports, which may be supported by the fact that these reports are used to document findings from tests conducted during the current visit. Emergency Department reports and Discharge Summaries contain similar proportions of historical conditions (17% and 19% respectively), which might be explained by the fact that both reports describe a patient's temporal progression throughout the stay in the Emergency Department or the hospital.

Surgical Pathology reports may be the most temporally distinct report in our study, showing the highest proportion of historical conditions. This may seem counter-intuitive given that Surgical Pathology reports also facilitate the reporting of findings described from a recent physical specimen. However, we had a small sample size (28 conditions in seven reports), and most of the historical conditions were described in a single addendum report. Removing this report decreased the prevalence of historical conditions to 23% (3/13).

Discharge Summaries and Emergency Department reports displayed more variety in the observed types of temporal expressions (9 to 14 subtypes) and trigger terms (10 to 12 terms) than other report genres. This is not surprising considering the range of events described in these reports. Other reports tend to have between zero and three subtypes of temporal expressions and zero and seven different trigger terms. In all report types, temporal expressions were mainly subtype *past*, and the most frequent trigger term was *history*.

Our second objective was to identify which features predict whether a condition is historical or recent. Due to high prevalence of the recent category, we were especially interested in discovering temporal features that predict whether a condition is historical. With one exception (*date* greater than four weeks prior to the current visit), temporal expression features always occurred in compound rules in which the temporal expression value had to co-occur with another feature value. For instance, any temporal expression in the category *key event* had to also occur in the *secondary diagnosis* section to classify the condition as historical. For ex-

ample, in "SECONDARY DIAGNOSIS: Status post ***Coronary artery bypass graft*** with complication of mediastinitis" the key event is the *coronary artery bypass graft*, the section is *secondary diagnosis*, and the correct classification is historical.

Similarly, verb tense and aspect were only useful in conjunction with other feature values. One rule predicted a condition as historical if the condition was modified by the trigger term *history* and fell within the scope of a *present tense verb with no aspect*. An example of this is "The patient *is* a 50 year old male with ***history*** of hypertension." Intuitively, one would think that a past tense verb would always predict historical; however, we found the presence of a past tense verb with no aspect was a feature only when the condition was in the *Patient History* section. Sometimes the absence of a verb in conjunction with another feature value predicted a condition as historical. For example, in the sentences "PAST MEDICAL HISTORY: ***History*** of COPD. ***Also*** *diabetes…"* *also* functioned as a trigger term that extended the scope of a previous trigger term, *history*, in the antecedent sentence.

A few historical trigger terms were discovered as simple rules by the rule learners: *no change*, *previous*, *known*, *status post*, and *history*. A few rules incorporated both a trigger term and a particular section header value. One rule predicted historical if the trigger term was *status post* and the condition occurred in the *History of Present Illness* section. This rule would classify the condition *CABG* as historical in "HISTORY OF PRESENT ILLNESS: The patient is...***status post*** CABG." One important detail to note is that a number of the temporal expressions categorized as *Fuzzy Time* also act as trigger terms, such as *history* and *status post*—both of which were learned by J48. A historical trigger term did not always predict the category historical. In the sentence "No focal sensory or motor deficits on ***history***," *history* may suggest that the condition was not previously documented, but was interpreted as not presently identified during the current physical exam.

Finally, sections appeared in the majority of JRip and RL historical rules: 4/8 simple rules and 13/18 compound rules. A few sections were consistently classified as historical: *Past Medical History*, *Allergies*, and *Social History*. One important point to address is that these sections were manually annotated.

Our results revealed a few unexpected observations. We found at least two trigger terms indicated in the J48 rules, *also* and *status post*, which did not have the same predictive ability across report genres. For instance, in the statement "TRANSFER DIAGNOSIS: ***status post*** coiling for left posterior internal carotid artery aneurysm," *status post* indicates the reason for the transfer as an inpatient from the Emergency Department and the condition is recent. In contrast, *status post* in a Surgical Pathology report was interpreted to mean historical (e.g., PATIENT HISTORY: ***Status post*** double lung transplant for COPD.) In these instances, document knowledge of the meaning of the section may be useful to resolve these cases.

One other unexpected finding was that the trigger term *chronic* was predictive of recent rather than historical. This may seem counterintuitive; however, in the statement "We are treating this as *chronic* musculoskeletal pain with oxycodone", the condition is being referenced in the context of the reason for the current visit. Contextual information surrounding the condition, in this case treating or administering medication for the condition, may help discriminate several of these cases.

Our third objective was to assess ConText in relation to the rules learned from manually annotated temporal features. J48 and ConText emphasized the use of trigger terms as predictors of whether a condition was historical or recent and performed with roughly the same overall accuracy. JRip and RL learned rules that incorporated other feature values including sections and temporal expressions, resulting in a 12% increase in historical recall over ConText and a 31% increase in historical recall over J48.

Many of the rules we learned can be easily extracted and incorporated into ConText (e.g., trigger terms *previous* and *no change*). The ConText algorithm largely relies on the use of trigger terms like *history* and one section header, *Past Medical History*. By incorporating additional section headers that may strongly predict historical, ConText could potentially predict a condition as historical when a trigger term is absent and the header title is the only predictor as in the case of "ALLERGIES: peanut allergy". Although these sections header may only be applied to Emergency Department and Discharge Summaries, trigger terms and temporal expressions may be generalizable across genre of reports. Some rules do not lend themselves

to ConText's trigger-term-based approach, particularly those that require sophisticated representation and reasoning. For example, ConText only reasons some simple durations like *several day history*. ConText cannot compute dates from the current visit to reason that a condition occurred in the past (e.g., stroke in *March 2000).* The algorithm performance would gain from such a function; however, such a task would greatly add to its complexity.

8 Limitations

The small sample size of reports and few conditions found in three report genres (Operative Gastrointestinal, Radiology, and Surgical Pathology) is a limitation in this study. Also, annotation of conditions, temporal category, sections, verb tense and aspect were conducted by a single author, which may have introduced bias to the study. Most studies on temporality in text focus on the temporal features themselves. For instance, the prevalence of temporal expressions reported by Zhou et al. (2006) include all temporal expressions found throughout a discharge summary, whereas we annotated only those expressions that modified the condition. This difference makes comparing our results to other published literature challenging.

9 Future Work

Although our results are preliminary, we believe our study has provided a few new insights that may help improve the state of the art for historical categorization of a condition. The next step to building on this work includes automatically extracting the predictive features identified by the rule learners. Some features may be easier to extract than others. Since sections appear to be strong indicators for historical categorization we may start by implementing the SecTag tagger. Often a section header does not exist between text describing the past medical history and a description of the current problem, so relying merely on the section heading is not sufficient. The SecTag tagger identifies both implicit and explicit sections and may prove useful for this task. To our knowledge, SecTag was only tested on Emergency Department reports, so adapting it to other report genres will be necessary. Both JRip and RL produced high performance, suggesting a broader set of features may improve historical classification; however, because these features do not result in perfect performance, there are surely other features necessary for improving historical classification. For instance, humans use medical knowledge about conditions that are inherently chronic or usually experienced over the course of a patient's life (i.e., HIV, social habits like smoking, allergies etc). Moreover, physicians are able to integrate knowledge about chronic conditions with understanding of the patient's reason for visit to determine whether a chronic condition is also a recent problem. An application that imitated experts would need to integrate this type of information. We also need to explore adding features captured at the discourse level, such as nominal and temporal coreference. We have begun work in these areas and are optimistic that they will improve historical categorization.

10 Conclusion

Although most conditions in six clinical report genres are recent problems, identifying those that are historical is important in understanding a patient's clinical state. A simple algorithm that relies on lexical cues and simple temporal expressions can classify the majority of historical conditions, but our results indicate that the ability to reason with temporal expressions, to recognize tense and aspect, and to place conditions in the context of their report sections will improve historical classification. We will continue to explore other features to predict historical categorization.

Acknowledgments

This work was funded by NLM grant 1 R01LM009427-01, "NLP Foundational Studies and Ontologies for Syndromic Surveillance from ED Reports".

References

Philip Bramsen, Pawan Deshpande, Yoong Keok Lee, and Regina Barzilay. 2006. *Finding Temporal Order in Discharge Summaries.* AMIA Annu Symp Proc. 2006; 81–85

Wendy W Chapman, David Chu, and John N. Dowling. 2007. *ConText: An Algorithm for Identifying Contextual Features from Clinical Text.* Association for Computational Linguistics, Prague, Czech Republic

Scott H. Clearwater and Foster J. Provost. 1990. *RL4: A Tool for Knowledge-Based Induction.* Tools for Artificial Intelligence, 1990. Proc of the 2nd Intern IEEE Conf: 24-30.

Joshua C. Denny, Randolph A. Miller, Kevin B. Johnson, and Anderson Spickard III. 2008. *Development and Evaluation of a Clinical Note Section Header Terminology. SNOMED.* AMIA 2008 Symp. Proceedings: 156-160.

Brian Hazlehurst, H. Robert Frost, Dean F. Sittig, and Victor J. Stevens. 2005. *MediClass: A system for detecting and classifying encounter-based clinical events in any electronic medical record.* J Am Med Inform Assoc 12(5): 517-29

Ann K. Irvine, Stephanie W. Haas, and Tessa Sullivan. 2008. *TN-TIES: A System for Extracting Temporal Information from Emergency Department Triage Notes.* AMIA 2008 Symp Proc: 328-332.

Roser Saur´ı, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. *TimeML Annotation Guidelines Version 1.2.1.* at: http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf

Marc Verhagen and James Pustejovsky. 2008. *Temporal Processing with TARSQI Toolkit.* Coling 2008: Companion volume – Posters and Demonstrations, Manchester, 189–192

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

Li Zhou, Genevieve B. Melton, Simon Parsons and George Hripcsak. 2006. *A temporal constraint structure for extracting temporal information from clinical narrative.* J Biomed Inform 39(4): 424-439.

Li Zhou and George Hripcsak. 2007. *Temporal reasoning with medical data--a review with emphasis on medical natural language processing.* J Biomed Inform Apr; 40(2):183-202.

Li Zhou, Simon Parson, and George Hripcsak. 2008. *The Evaluation of a Temporal Reasoning System in Processing Discharge Summaries*. J Am Med Inform Assoc 15(1): 99–106.

ONYX: A System for the Semantic Analysis of Clinical Text

Lee M. Christensen, Henk Harkema, Peter J. Haug,
Jeannie Y. Irwin, Wendy W. Chapman
Department of Biomedical Informatics
University of Pittsburgh
Pittsburgh, PA 15214, USA
lmc61 heh23 rey3 wec6 @pitt.edu

University of Utah
Salt Lake City, Utah, 84143, USA
Peter.Haug@intermountainmail.org

Abstract

This paper introduces ONYX, a sentence-level text analyzer that implements a number of innovative ideas in syntactic and semantic analysis. ONYX is being developed as part of a project that seeks to translate spoken dental examinations directly into chartable findings. ONYX integrates syntax and semantics to a high degree. It interprets sentences using a combination of probabilistic classifiers, graphical unification, and semantically annotated grammar rules. In this preliminary evaluation, ONYX shows inter-annotator agreement scores with humans of 86% for assigning semantic types to relevant words, 80% for inferring relevant concepts from words, and 76% for identifying relations between concepts.

1 Introduction

This paper describes ONYX, a sentence-level medical language analyzer currently under development at the University of Pittsburgh. Since ONYX contains a number of innovative ideas at an early stage of development, the objective of this paper is to paint a broad picture of ONYX and to present preliminary evaluation results rather than analyzing any single aspect in detail.

ONYX is being developed as part of a project aimed at extracting information from spoken dental examinations. Currently, dental findings must be charted after an exam is completed or may be charted by an assistant who acts as a transcriptionist during the exam. Our goal is to design a system capable of automatically extracting chartable findings directly from spoken exams, potentially also supporting automated decision support and quality control. We are also developing tools to enable the system to be ported to other clinical domains and settings.

Extracting information from unedited speech transcriptions presents a number of challenges. Sentences may be fragmented or telegraphic, and much of the speech may be irrelevant for our purposes. The following example illustrates some of these difficulties:

> *"Okay. Okay. Open. Okay. No. 1 is missing. Two occlusal distal amalgam. Actually, make that occlusal. Also, one palatal amalgam. Can you close just slightly? And perfect. Okay, now open again."*

The relevant findings in this example are that tooth number one is missing and tooth number two has amalgam fillings on the occlusal and palatal surfaces. Our ultimate challenge is to create a system that can recognize relevant sentences and perform competently in the face of the inherent ambiguity and noise commonly found in conversational speech. ONYX does not yet address all of these challenges, although we have clear directions we are pursuing as described in the Future Work section of this paper. Our goal in this paper is to describe the current state of ONYX and the innovations we feel will enable it to be adapted to complex NLP tasks in the future.

2 Overview of ONYX

ONYX is the middle component of a pipelined architecture as illustrated in figure 1. The entry point to this architecture is a speech-to-text analyzer, which takes input from a microphone worn

Proceedings of the Workshop on BioNLP, pages 19–27,
Boulder, Colorado, June 2009.

by the dentist and produces a transcription that ONYX analyzes for semantic content. ONYX's output is then passed to a discourse analyzer that applies dental knowledge to assemble ONYX's sentence-level semantic representations into chartable exam findings.

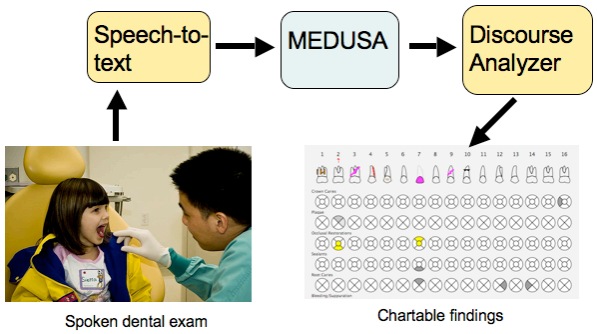


Figure 1. Speech-to-chart pipeline.

ONYX looks for dental conditions such as caries, fractures and translucencies; restorations such as fillings and crowns; tooth locations; and modifiers such as tooth part, tooth surface, and condition extent. It produces templates of words and concepts. Table 1 shows a summary of four templates (Dental Condition, Tooth Location, Surface and State) representing the meaning of "eight mesio might have a slight translucency."

Dental Condition	
Condition Concept	*translucency
Condition Term	"translucency"
Severity Concept	*superficial
Severity Term	"slight"
Tooth Location	
Location Concept	*numberEight
Tooth Number	"eight"
Surface	
Surface Concept	*mesial
Front/Back Term	"mesio"
State	
State Concept	*possible
State Term	"might"

Table 1: ONYX templates for "eight mesio might have a slight translucency." Terms with an * are inferred concepts.

ONYX's interpretations are represented as binary predicates that take the templates as arguments (for convenience, only the summary concepts from the templates are shown):

*ConditionAt(*translucency, *numberEight) &*
*LocationHasSurface(*numberEight, *mesial) &*
*StateOf(*translucency, *possible)*

ONYX builds on ideas from MPLUS (Christensen et al, 2002), which was used primarily to interpret radiology reports. MPLUS uses Bayesian networks (BNs) to produce filled templates. Through a training process, words from the corpus of training documents are manually associated with states of terminal nodes in a BN, and concepts are associated with states of nonterminal nodes. When MPLUS interprets a sentence, it instantiates the BNs with words from the sentence and infers the most probable concepts consistent with those words. It then generates templates filled with those words and concepts.

BNs have proven useful in semantic analysis (e.g. Ranum 1989, Koehler 1998, Christensen 2002): their performance degrades gracefully in the face of various types of lexical and syntactic noise. The main disadvantage with using BNs is their inherent computational complexity. ONYX employs a semantics-intensive form of parsing, interpreting each phrase as it is constructed rather than waiting until the syntactic analysis is completed to do the interpretation. For this reason we have developed an experimental probabilistic classifier for ONYX called a Concept Model (CM). CMs support a tree-structured representation of related words and concepts (figure 5), structurally similar to the BNs used by MPLUS, but using a more efficient model of computation. In essence CMs are trees of Naïve Bayes classifiers, although they contain enhancements, not described in this study, which in general make them more accurate than strict Naïve Bayes. Each node together with its children constitutes a single classifier. When a CM is applied to words in a sentence, word-level CM states are assigned a probability based on training data. Probabilities are propagated upwards through the CM, calculating probabilities for all concepts that depend directly or indirectly on the words of the sentence.

3 ONYX Syntactic Analyzer

For this project we desired a parser that was fast, flexible and robust. We designed a variation on a bottom-up chart parser (Kay, 1980) and hand-crafted an initial set of 52 context-free grammar

rules. Chart parsers based on Kay's algorithm maintain an agenda of "edges," which correspond to partially or completely instantiated grammar rules. In the original algorithm, for each new phrase added to the chart an edge is created for each rule that can begin with that phrase. In addition, each existing edge that abuts and can be extended with that phrase is duplicated with a pointer to the new phrase. When an edge has no more unmatched components, it is regarded as a new phrase that can begin or extend other edges. Since edges are used to anticipate all possible continuations of phrases vis-à-vis the grammar, the number of edges grows quickly relative to the number of words in the sentence. Charniak et al. (1998) noted that exhaustively parsing maximum-40-word sentences from the Penn II treebank requires an average of 1.2 million edges per sentence.

ONYX's parse algorithm replaces edges with binary links. We briefly describe this new algorithm. A set of binary link templates is defined for each grammar rule. For instance, the rule S->NP AUX VP (labeled S1) would produce the templates [s1:np,aux] and [s1:aux,vp]. When a phrase is added to the chart, binary links for all applicable rules are added from that phrase to juxtaposed phrases to the left and right on the chart. When a right or left-terminating link is added (all links for rules with two or three components are right or left terminating), a quick search is done in the other direction for links belonging to the same rule. Each complete set of links defines a new phrase of the target type, as shown in figure 2.

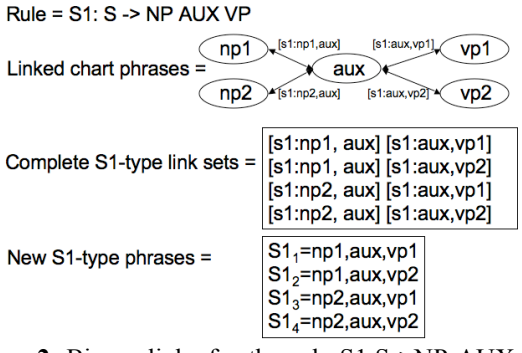


Figure 2. Binary links for the rule S1:S->NP AUX VP used to generate new phrases of type S1 from juxtaposed NP, AUX, and VP phrases on the chart.

Although we have not analyzed the time and space complexity of this algorithm, it has proven to be more efficient than the edge-based parser used by MPLUS. Time and space complexity for chart parsers is calculated based on the number of edges produced, which has been shown to be $O(n^3)$, with n words in a sentence. Since binary links, unlike edges, are only used to record grammatical relations between juxtaposed phrases on the chart (rather than anticipating possible continuations), are not duplicated, and can participate in the creation of multiple new phrases, the number of binary links grows more slowly than the number of edges. On the other hand, the need to search for completed link sets increases processing time. We plan to formally analyze the time and space requirements of this algorithm in a future study.

4 ONYX Semantic Analyzer

In ONYX syntax and semantics are highly integrated. Rather than waiting for a completed parse tree to begin the interpretation process, ONYX semantically interprets each phrase as it is created and before it is placed on the chart. Each phrase is assigned a "goodness" score based in part on the goodness of its semantic interpretation, and this score is used in determining the order in which phrases are expanded, resulting in a semantically guided best-first search.

To represent semantic relations between templates, ONYX uses a custom-built first-order predicate language with a syntax based roughly on the Knowledge Interchange Format (Genesareth & Fikes, 1992). ONYX interpretations are conjuncts of binary predicates formulated in this language, with templates as arguments. This language is for internal use only; ONYX will use standard language protocols for communicating with external systems. We decided to implement our own language rather than using an existing implementation in order to have access to the underlying data structures, which we use in three ways not traditionally applied to symbolic languages: 1- We have extended our language to include Java objects as constants and Java methods as functions and relations. In particular, CM templates are treated as constants in the language, and CMs are semantically typed functions that map words to templates. 2- As described next, ONYX's default mode of semantic interpretation is based on a form of graph unification. Binary predicates are treated as unifi-

able links in a graph as shown in figure 3. 3-ONYX uses the predicate structure of an interpretation to pass information between CMs. For instance, if an interpretation contains the relation *ConditionAt(Condition, Location)*, ONYX inserts the summary concept from the Location CM into the Condition CM. This allows the Condition CM to factor tooth location into its determination of the most probable Condition concept.

Figure 3 illustrates ONYX's unification-based interpretation process. ONYX relies on a semantic network that defines types and relations in the dental domain (figure 4). As dental concepts are brought together in a phrase, links connecting those concepts are extracted from the semantic network and formulated into binary predicates in an interpretation. As phrases are joined together in larger phrases, their relations and templates are merged, resulting in an interpretation tree denoting a dental object (e.g. dental condition, tooth location, tooth surface) with possibly multiple levels of modifiers. For instance, the interpretation for "eight mesio might have a slight translucency" can be generated from the partial interpretations of the phrases "eight mesio", "might" and "slight translucency" as shown in figure 3.

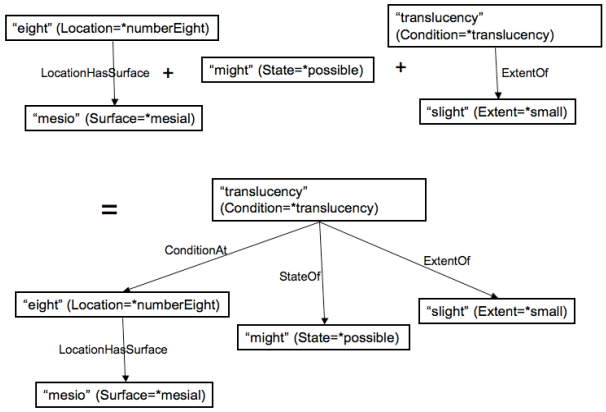


Figure 3. Interpreting "eight mesio might have a slight translucency" using graph unification.

There are two primary justifications for using unification in this way. First, conjoined phrases, particularly noun phrases, often contain unifiable partial descriptions of a single object. Second, if concepts appear together in a phrase, there is a good chance that relations connecting those concepts in the semantic network are captured, explicitly or implicitly, in the meaning of the phrase.

The dental semantic network is shown in figure 4. Terminal (white) nodes define concrete semantic types associated with dental CMs. For instance, the DentalCondition type is associated with the concept model shown in figure 5.

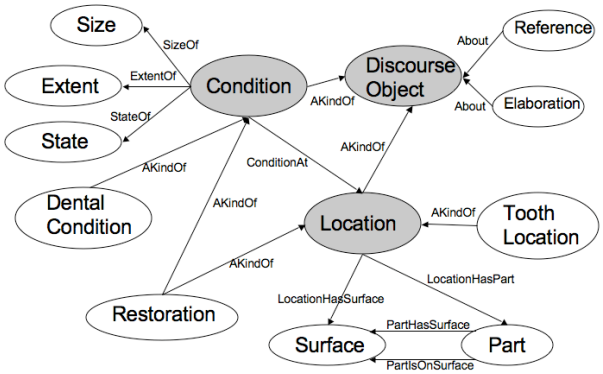


Figure 4. Semantic network for dental exams.

Nonterminal (gray) nodes represent abstract types with no associated CMs. A concrete type may have more than one abstract parent type. For instance, a Restoration, such as a crown, is both a Condition and a Location. As such, it can exist at a tooth location, e.g., "the crown on tooth 5," and it can be the location of condition, e.g., "the crack on the crown on tooth 5." Since a concrete type can have multiple parent types, ONYX often produces multiple alternative interpretations over words of a sentence. For instance, ONYX may produce two interpretations for "mesial amalgam"—one referring to the mesial surface of an amalgam filling, and one referring to an amalgam filling on the mesial surface of some unspecified tooth. ONYX uses probabilities derived from training cases to prefer the latter interpretation, which is the more likely of the two.

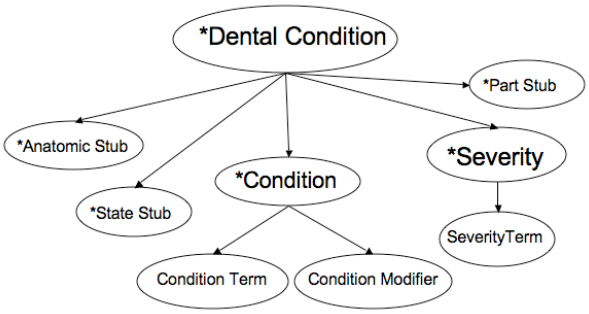


Figure 5. Dental Condition Concept Model.

Each concept model has a tree structure as illustrated in figure 5, which shows the structure of the Dental Condition CM. Nonterminal nodes repre-

sent concepts, and terminal nodes represent words, with the exception of stub nodes. The value of a stub node is the summary concept (i.e., root node) from the CM of the same name.

One problem with ONYX's graph-based model of interpretation is that the semantic network does not capture all relations that might be expressed in a dental exam. The network was deliberately kept simple by including mostly relations that are categorically true (e.g., all teeth have surfaces) or that are frequently talked about (e.g., restorations are frequently mentioned as being locations of other conditions). This restriction helps keep the unification process tractable and minimizes ambiguity, but interpretations may miss important points. For instance, the ONYX interpretation of "15 occlusal amalgam" is *ConditionAt(*filling, *toothFifteen) & LocationHasSurface(*toothFifteen, *occlusal)* which can be paraphrased as "a filling at tooth 15 and tooth 15 has an occlusal surface". This interpretation misses the important fact that the filling is *on* the occlusal surface of tooth 15, which we would normally infer from the fact that "occlusal" adjectively modifies "amalgam." Another limitation is that although the semantic network as it stands can describe single objects with their modifiers, it cannot be used to build up complex descriptions involving multiple objects of the same type.

To address these limitations we have added a second, more specialized mode of interpretation that is contingent on lexical and syntactic information from the parse and that can introduce into an interpretation predicates that do not exist in the semantic network. This mode of interpretation uses semantic types and patterns attached to grammar rules. As an example, the rule NP -> AP NP can be semantically annotated thus:

```
NP<Restoration> -> AP<Surface> NP<Restoration>
        => OnSurface(Restoration, Surface)
```

This rule captures the idea that if a Surface-type adjectival phrase modifies a Restoration-type noun phrase, the restoration exists on that surface. Applied to "occlusal amalgam" this rule would produce an interpretation *OnSurface(*filling, *occlusal),* which is the relation missing from the previous example. Semantically annotated grammar rules can also connect objects of the same semantic type. For instance, we might define a rule

```
NP<Condition> -> NP<Condition1> "caused by"
                        NP<Condition2>
        => CausedBy(condition1, condition2)
```

This rule can match phrases such as "leakage caused by a crack along the lingual surface", and link the two conditions (leakage and crack) with a *CausedBy* relation. This mechanism enables ONYX to construct complex descriptions with multiple objects.

We have added a mechanism to the ONYX training tool that allows semantically annotated grammar rules to be generated semi-automatically during training. A human annotator with sufficient linguistic background can view the parse trees generated by ONYX for corpus sentences, repair those parse trees and/or add new semantic relations if necessary, then apply a function that creates copies of the rules embodied in those trees with semantic types and predicates attached.

5 Integrating Syntax and Semantics

Although most NLP systems apply semantic analysis to completed parse trees, in humans the two processes are more integrated. Syntactic expectations are greatly influenced by word meanings, as illustrated by "garden path" sentences such as "The man whistling tunes pianos." In ONYX, syntax and semantics are highly interleaved. This is accomplished in several ways:

1- ONYX's parse algorithm permits words to be processed in any order, rather than strictly left-to-right, since binary grammar links can be added to the phrase chart in any order. This allows ONYX to be instructed to focus on semantically interesting words first, which can be used, among other things, to gather useful information from ungrammatical speech or run-on sentences where attempting to look for complete sentences in strict left-to-right fashion would be unsuccessful.

2- ONYX implements a variation on a probabilistic context free grammar (PCFG) (Charniak, 1997) that associates grammar rules with semantic types. Based on training, a conditional probability is calculated for each <rule, type> pair given specific

<rule, type> assignments to the rule's components. The probability of a phrase is then calculated as the product of the probabilities of the phrase rule and its semantic type, given the rule and type of each of its child phrases. ONYX is then able to prefer phrases that best accommodate the semantic types of their constituents. Specifically,

```
prob(phrase) =
∏(prob(rule(phrase) + semtype(phrase) |
    rule(childPhrase) + semtype(childPhrase)))
```

3- One hard problem in parsing is determining the correct structure of conjunctive noun phrases. ONYX applies semantic guidance to solve this problem. For instance, in a chest radiology report the words "right and left lower lobe opacity" can be grouped in several different ways, and different groupings can produce different interpretations. The correct grouping should be something like: [[[right and left] [lower lobe]] opacity], rather than [[right and [left lower]] [lobe opacity]]. ONYX currently employs a simplistic representation of the meaning of a conjunctive phrase as a list of interpretations. The correct interpretations for "right and left lower lobe opacity" would be two predicate expressions covering the words (right, lower, lobe, opacity) and (left, lower, lobe, opacity). ONYX generates a measure of the similarity of these expressions based on the cosine similarity of the lists of non-null nodes in their CM templates. This measure is factored into the phrase's goodness score under the heuristic that semantically balanced conjunctive phrases are more likely to be correct than imbalanced ones.

4- As mentioned earlier, ONYX can utilize grammar rules annotated with semantic types and patterns. Semantically annotated rules constrain phrases to match particular semantic types, and can contribute predicates to the interpretation of those phrases. This gives ONYX's grammar the character of a semantic grammar.

5- Phrases are weighted and preferred by ONYX according to their goodness score, which is based on three measures: the probability of the phrase as determined by the PCFG formula, the conjunct cosine similarity score, if applicable, and the goodness score of the phrase's semantic interpretation. The PCFG and conjunct similarity formulas are based on semantic criteria, as mentioned earlier. Interpretation goodness scores are calculated as a simple product of the probabilities of the semantic relation predicates they contain. Relation probabilities are in turn derived from training data, and are conditioned on the concepts they contain. The probability of a relation is calculated as the number of times a pair of concepts appears together in the target relation divided by the number of times they appear together in any set of relations. The goodness score of a phrase is thus highly semantically determined.

```
goodness(phrase) = F(prob(phrase, PCFG),
                     conjunctSimilarity(phrase),
                     goodness(interp(phrase))
goodness(interp(phrase)) =
            ∏prob(relations(interp(phrase)))
prob(relation) =
    count(relation + concepts(relation)) /
    count(anyConnection(concepts(relation)))
```

6 Evaluation

We performed a preliminary evaluation of ONYX for the extraction of relevant dental concepts and relations on a set of twelve documents in our current training corpus.

Reference Standard. Each document was independently annotated by three human annotators (authors LC, JI and HH), who used the ONYX training tool to fill in templates representing dental conditions, tooth locations and other relevant concepts, as well as to select the semantic relations linking those templates. The annotators then reviewed disagreements and by consensus created a reference standard set of templates and relations. Where the annotators did not have sufficient dental knowledge to reach an agreement they consulted dental clinicians.

Outcome Metrics. To evaluate ONYX on the relatively small corpus of documents, we applied a leave-one-out approach: for each sentence in the reference standard, ONYX was trained using the templates from the remaining reference standard sentences. ONYX was then applied to the target sentence, and the resulting templates and relations were compared to the reference standard. We measured inter-annotator agreement (IAA) between ONYX and the reference standard using the formula described in Roberts et al (2007):

IAA = (2 * correct) / (spurious + missing + correct)

We calculated IAA separately for CM words, concepts, and semantic relations. A correct match is a word, concept or relation generated by both the reference standard and ONYX; a spurious item is one ONYX generated that did not exist in the reference standard; and a missing item is one that existed in the reference standard but was not generated by ONYX. In addition to IAA we identified the concepts and relations most commonly in error and calculated percentages for those errors.

We compared ONYX's performance on the target documents with that of a simple baseline parser we created for this purpose. The baseline parser processes the words of a sentence from left to right, creating phrases for sets of juxtaposed words that can be interpreted together using the semantic network. No grammar rules are employed, there is no analysis of conjunctive phrases, and goodness scores are not calculated. Our goal was to get a feel for how much these factors contribute to generating correct interpretations. There is no precedence for this particular approach as far as we are aware, so we regard this comparison as informative but not definitive.

7 Results

IAA results for ONYX and the baseline parser are shown in table 2. ONYX performs best at inserting words into appropriate nodes in the CMs, with IAA of 86%, and less well for inferring the best concept (80%) and identifying relations among concepts (76%). ONYX consistently out-performs the baseline parser.

Table 2: IAA for assignment of words, concepts, and relations.

		IAA
Words (n = 904)	ONYX	86%
	Baseline	57%
Concepts (n = 1186)	ONYX	80%
	Baseline	53%
Relations (n = 297)	ONYX	76%
	Baseline	41%

Although this study does not examine all the reasons for the differences in performance between ONYX and the baseline parser, some reasons can be illustrated with an example. Conjunctive phrases are common in dental discourse, and a failure to handle conjuncts can result in both concept and relation errors. For instance, given the sentence "4, 5, 6, 7 fine" ONYX generates separate interpretations covering the word groupings (4, fine), (5, fine), (6, fine), and (7, fine), which would yield four ConditionAt relations, four Location concepts (*numberFour, *numberFive, *numberSix, *numberSeven) and one Condition concept (*normalTooth) appearing in each relation. The baseline parser in contrast does not discover this distribution of terms and so omits all but the ConditionAt relation over (7, fine). Trying to merge juxtaposed tooth numbers, the baseline parser also infers that at least some of these denote tooth ranges instead of individual teeth (e.g. interpreting "4, 5" as "4 to 5" instead of "4 and 5"), which causes it to misclassify Location concepts. The ability to generate correct parse trees and to use the structure of those parse trees in the interpretation process is important in generating correct interpretations.

Tables 3 and 4 show breakdowns by percentage of the concepts and relations most commonly in error in ONYX's interpretations (errors accounting for more than 15%).

Table 3: Per-concept error percentages

Dental Condition Summary Concept	18%
Tooth Location Summary Concept	17%
Dental Condition Intermediate Concept	16%
Surface Summary Concept	15%
Total	66%

Table 4: Per-relation error percentages.

Surface of Part	47%
Location of Condition	23%
Total	70%

8 Related Work

ONYX is a new application inspired by SPRUS (Ranum, 1989), Symtext (Koehler, 1998), and MPLUS (Christensen, 2002), which all used Bayesian Networks to infer relevant findings from text. Other medical language processing systems implement different approaches to encode clinical concepts and their modifiers, along with relations between concepts, including MedLEE (Friedman,

1994), a largely statistical system by Taira and colleagues (Taira, 2007), and MedSyndikate (Hahn, 2002).

Many of ONYX's components leverage research in the general and clinical NLP domains, including the use of chart parsing (Kay, 1980) and probabilistic context free grammars (Charniak, 1997). ONYX's use of semantically annotated grammar rules was inspired in part by MedLEE (Friedman et al, 1994), which uses a semantic grammar.

Although incorporating ideas and approaches from others, we feel that ONYX is unique in several ways, including its high level of syntactic/semantic integration and the ways in which it blends symbolic and probabilistic representations of domain knowledge. We plan to make ONYX available through open source when the system is more complete.

9 Limitations

There are several limitations to this study. Although ONYX introduces several innovations, these are not described in detail in this study and are not individually evaluated for their effect on ONYX's performance. Instead, this study presents a broad overview of ONYX and evaluates ONYX's overall performance against a reference standard on a small test sample. Another limitation of our study is the baseline system—because similar systems generate different output than ONYX and do not model the same domain, finding a competitive baseline application is difficult. In spite of its imperfection, we believe the baseline we implemented to be reasonable.

10 Future Work

One limitation of a system like ONYX is the overhead of manually creating complex training cases. To address this shortcoming, the ONYX training tool invokes ONYX to automatically create templates and relations for corpus sentences, and human trainers correct any mistakes. A semi-automated approach greatly speeds up the training process and facilitates agreement among human trainers. We plan to further automate this process using an approach derived from Thelen & Riloff (2002), which uses a classifier with features based on extraction patterns derived from Autoslog (Riloff, 1996). We plan to adapt this approach to automatically classify CM word assignments, and also to automatically classify semantic relations between CM templates. We will add this functionality to the training tool to enable it to find and annotate relevant sentences automatically where possible. We will also apply this functionality to enable ONYX to recognize relevant sentences in new documents based on their similarity to training sentences, and we will use semantic patterns stored with training sentences to aid in interpreting noisy segments of text that ONYX cannot parse. We plan to compare the performance of grammar-based and feature-based semantic analysis in future studies. With more fully automated training, we also hope to make ONYX more easily portable to new domains and clinical settings in the future.

Conclusions

This paper describes ONYX, which is being developed as part of a system for extracting chartable findings from spoken dental examinations. ONYX contains a number of innovative ideas including a novel adaptation of Kay's (1980) parse algorithm; a symbolic language extended to include probabilistic and procedural elements; an integration of syntax and semantics that includes a semantically weighted probabilistic context free grammar and interpretation based both on a semantic network and a semantic grammar. Considering ONYX's early stage of development it performed reasonably well in this limited evaluation but must be extended to address challenges in extracting findings from spoken dental exams.

Acknowledgments

This work was funded by NIDCR 1 R21DE018158-01A1 "Feasibility of a Natural Language Processing-based Dental Charting Application.

References

E. Charniak. 1997. *Statistical parsing with a context-free grammar and word statistics. In Proceedings of the Fourteenth National Conference on Artificial Intelligence,* pp. 598-603.

E. Charniak, S. Goldwater and M. Johnson. 1998. *Edge-Based Best-First Chart Parsing. In Proceedings of*

the Sixth Workshop on Very Large Corpora, pp. 127-133.

Lee M. Christensen, Peter J. Haug, and Marcelo Fiszman. 2002. MPLUS: A Probabilistic Medical Language Understanding System. *Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain, Philadelphia,* pp. 29 – 36.

Carol Friedman, Phil Alderson, John Austin, James Cimino, & Stephen Johnson. 1994. A general natural language text processor for clinical radiology. *Journal of American Medical Informatics Association* 1(2), pp. 161–174.

M. R. Genesereth and R. E. Fikes. Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-92-1, Stanford, CA, USA, 1992.

Hahn U, Romacker M, Schulz S. 2002. Medsyndikate-a natural language system for the extraction of medical information from findings reports. *Int J Med Inf.* 67(1-3), pp. 63-74.

M. Kay. 1980. Algorithm schemata and data structures in syntactic parsing. In Readings in Natural Language Processing, pp. 35 – 70. Morgan Kaufmann Publishers Inc.

Koehler, S. B. 1998. SymText: A natural language understanding system for encoding free text medical data. Ph.D. Dissertation, University of Utah.

Ranum D.L. 1989. Knowledge-based understanding of radiology text. *Comput Methods ProBiomed.* Oct-Nov;30(2-3) pp. 209-215.

Ellen Riloff, 1996. Automatically Generating Extraction Patterns from Untagged Text. *Proceedings of the Thirteenth National Conference on Artiticial Intelligence,* pp. 1044 – 1049. The AAAI Press/MIT Press.

Angus Roberts, Robert Gaizauskas, Mark Hepple, Neil Davis, George Demetriou, Yikun Guo, Jay Kola, Ian Roberts, Andrea Setzer, Archana Tapuria, Bill Wheeldin. 2007. The CLEF Corpus: Semantic Annotation of Clinical Text. *AMIA 2007*, pp. 625 – 629.

Taira R, Bashyam V, Kangarloo H. 2007. *A field theory approach to medical natural language processing. IEEE Transactions in Inform Techn in Biomedicine* 11(2).

Michael Thelen and Ellen Riloff. 2002. *A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts. Proceedings of the ACL-02 conference on Empirical methods in natural language processing,* pp. 214 – 221.

Learning the scope of hedge cues in biomedical texts

Roser Morante, Walter Daelemans
CNTS - Language Technology Group
University of Antwerp
Prinsstraat 13, B-2000 Antwerpen, Belgium
{Roser.Morante,Walter.Daelemans}@ua.ac.be

Abstract

Identifying hedged information in biomedical literature is an important subtask in information extraction because it would be misleading to extract speculative information as factual information. In this paper we present a machine learning system that finds the scope of hedge cues in biomedical texts. The system is based on a similar system that finds the scope of negation cues. We show that the same scope finding approach can be applied to both negation and hedging. To investigate the robustness of the approach, the system is tested on the three subcorpora of the BioScope corpus that represent different text types.

1 Introduction

Research on information extraction of biomedical texts has grown in the recent years. Most work concentrates on finding relations between biological entities, like genes and proteins (Krauthammer et al., 2002; Mitsumori et al., 2006; Krallinger et al., 2008a; Krallinger et al., 2008b). Determining which information has been hedged in biomedical literature is an important subtask of information extraction because extracted information that falls in the scope of hedge cues cannot be presented as factual information. It should be discarded or presented separately with lower confidence. The amount of hedged information present in texts cannot be understimated. Vincze et al. (2008) report that 17.70% of the sentences in the abstracts section of the BioScope corpus and 19.44% of the sentences in the full papers section contain hedge cues. Light et al. (2004) estimate that 11% of sentences in MEDLINE abstracts contain speculative fragments. Szarvas (2008) reports that 32.41% of gene names mentioned in the hedge classification dataset described in Medlock and Briscoe (2007) appears in a speculative sentence.

In this paper we present a machine learning system that finds the scope of hedge cues in biomedical texts. Finding the scope of a hedge cue means determining at sentence level which words in the sentence are affected by the hedge cue. The system combines several classifiers and works in two phases: in the first phase hedge cues (i.e., words indicating speculative language) are identified, and in the second phase the full scope of these hedge cues is found. This means that for a sentence like the one in Example (1) taken from the BioScope corpus (Szarvas et al., 2008), the system performs two actions: first, it detects that *suggest, might*, and *or* are hedge signals; second, it detects that *suggest* has as its scope *expression of c-jun, jun B and jun D genes might be involved in terminal granulocyte differentiation or in regulating granulocyte functionality*, that *might* has as its scope *be involved in terminal granulocyte differentiation or in regulating granulocyte functionality*, and that *or* has as its scope *in regulating granulocyte functionality*.

(1) These results <xcope id="X7.5.3" ><cue type= "speculation" ref="X7.5.3"> **suggest** </cue> that <xcope id= "X7.5.2">expression of c-jun, jun B and jun D genes <cue type= "speculation" ref= "X7.5.2"> **might** </cue> be involved <xcope id="X7.5.1">in terminal granulocyte differentiation <cue type= "speculation" ref="X7.5.1" >**or**</cue> in regulating granulocyte functionality </xcope></xcope></xcope>.

Proceedings of the Workshop on BioNLP, pages 28–36,
Boulder, Colorado, June 2009.

Contrary to current practice to only detect modality, our system also determines the part of the sentence that is hedged. We are not aware of other systems that perform this task. The system is based on a similar system that finds the scope of negation cues (Morante and Daelemans, 2009). We show that the system performs well for this task and that the same scope finding approach can be applied to both negation and hedging. To investigate the robustness of the approach, the system is tested on three subcorpora of the BioScope corpus that represent different text types. Although the system was developed and tested on biomedical text, the same approach can also be applied to text from other domains.

The paper is organised as follows. In Section 2, we summarise related work. In Section 3, we describe the corpus on which the system has been developed. In Section 4, we introduce the task to be performed by the system, which is described in Section 5. Results are presented and discussed in Section 6. Finally, Section 7 puts forward some conclusions.

2 Related work

Hedging has been broadly treated from a theoretical perspective. The term *hedging* is originally due to Lakoff (1972), who introduces it in relation to prototype theory. Palmer (1986) defines a term related to hedging, *epistemic modality*, which expresses the speaker's degree of commitment to the truth of a proposition. Saurí et al. (2006) research the modality of events, which "expresses the speaker's degree of of commitment to the events being referred to in a text". They treat a wide spectrum of modal types and present the codification of modality information with the specification language TimeML, which allows to mark modality cues at a lexical level and at a syntactic level.

As for research that focuses specifically on scientific texts with descriptive purposes, Hyland (1998) describes hedging in scientific research articles, proposing a pragmatic classification of hedge expressions based on an exhaustive analysis of a corpus. The catalogue of hedging cues includes modal auxiliaries, epistemic lexical verbs, epistemic adjectives, adverbs, and nouns. Additionally, it includes also a variety of non–lexical cues. Light et al. (2004) analyse the use of speculative language in MEDLINE abstacts. They studied the expression of levels of belief (hypothesis, tentative conclusions, hedges, and speculations) and annotated a corpus of abstracts in order to check if the distinction between high speculative, low speculative and definite sentences could be made reliably. They found that the speculative vs. definite distinction was reliable, but the distinction between low and high speculative was not. Thompson et al. (2008) report on a list of words and phrases that express modality in biomedical texts and put forward a categorisation scheme. The list and the scheme are validated by annotating 202 MEDLINE abstracts.

Some NLP applications incorporate modality information. Friedman et al. (1994) develop a medical text processor "that translates clinical information in patient documents into controlled vocabulary terms". The system uses a semantic grammar that consists of rules that specify well-formed semantic patterns. The extracted findings are assigned one of five types of modality information: *no, low certainty, moderate certainty, high certainty* and *cannot evaluate*. Di Marco and Mercer (2005) use hedging information to classify citations. They observe that citations appear to occur in sentences marked with hedging cues.

Work on hedging in the machine learning field has as a goal to classify sentences into speculative or definite (non speculative). Medlock and Briscoe (2007) provide a definition of what they consider to be hedge instances and define hedge classification as a weakly supervised machine learning task. The method they use to derive a learning model from a seed corpus is based on iteratively predicting labels for unlabeled training samples. They report experiments with SVMs on a dataset that they make publicly available[1]. The experiments achieve a recall/precision break even point (BEP) of 0.76. They apply a bag-of-words (BOG) approach to sample representation. Medlock (2008) presents an extension of this work by experimenting with more features (part-of-speech (PoS), lemmas, and bigrams). Experiments show that the PoS representation does not yield significant improvement over the results in

[1] Available at `http://www.benmedlock.co.uk/hedgeclassif.html`.

Medlock and Briscoe (2007), whereas with a lemma representation the system achieves a peak performance of 0.8 BEP, and with bigrams of 0.82 BEP. Szarvas (2008) follows Medlock and Briscoe (2007) in classifying sentences as being speculative or non-speculative. Szarvas develops a MaxEnt system that incorporates bigrams and trigrams in the feature representation and performs a complex feature selection procedure in order to reduce the number of keyword candidates. It achieves up to 0.85 BEP and 85.08 F1 by using an external dictionary. Kilicoglu and Bergler (2008) apply a linguistically motivated approach to the same clasification task by using knowledge from existing lexical resources and incorporating syntactic patterns. Additionally, hedge cues are weighted by automatically assigning an information gain measure and by assigning weights semi–automatically depending on their types and centrality to hedging. The system achieves results of 0.85 BEP.

As mentioned earlier, we are not aware of research that has focused on learning the scope of hedge signals inside or outside of the biomedical domain, which makes a direct comparison with the approaches described here impossible.

3 Hedge cues in the BioScope Corpus

The system has been developed using the BioScope corpus (Szarvas et al., 2008; Vincze et al., 2008)[2], a freely available resource that consists of medical and biological texts. In the corpus, every sentence is annotated with information about negation and speculation. The annotation indicates the boundaries of the scope and the keywords, as shown in (1) above. In the annotation, scopes are extended to the biggest syntactic unit possible, so that scopes have the maximal length, and the speculation cue is always included in the scope.

The BioScope corpus consists of three parts: clinical free-texts (radiology reports), biological full papers and biological paper abstracts from the GENIA corpus (Collier et al., 1999). Table 1 shows statistics about the corpora. Hedge cues are represented by one or more tokens, as (2) shows, where the hedge cues that appear in the three corpora are listed. The complete list of all hedge cues comprises 176 cues. In the same corpora the number of negation cues is lower, 38.

(2) apparent, apparently, appear, assume, can, consider, consistent with, could, either, indicate, likely, may, no evidence, not, or, perhaps, possible, possibly, presumably, probable, probably, should, suggestion, support, think, unclear, whether, would

35 hedge cues that occur in the clinical reports subcorpus do not occur in the abstracts subcorpus, and 34 hedge cues that appear in the papers subcorpus do not appear in the abstracts subcorpus. Only 15.90% of the total of hedge cues appear in the three subcorpora. The most frequent hedge cues in the abstracts subcorpus are *may* (19.15 %), *appear* (5.30 %), and *or* (4.45 %); in the papers subcorpus, *suggest* (10.26 %), *may* (9.97 %), and *might* (5.86 %); and in the clinical subcorpus, *or* (24.27 %), *suggest* (5.62 %), and *evaluate for* (5.27 %).

	Clinical	**Papers**	**Abstracts**
#Documents	1954	9	1273
#Sentences	6383	2670	11871
#Words	41985	60935	282243
#Lemmas	2320	5566	14506
Av. length sentences	7.73	26.24	26.43
%Hedge sentences	13.39	19.44	17.70
# Hedge cues	1189	714	2769
Av. length scopes	5.92	14.37	16.27
Av. length scopes to the right	5.15	13.00	15.44
Av. length scopes to the left	2.46	5.94	5.60
% Scopes to the right	73.28	76.55	82.45
% Scopes to the left	26.71	23.44	17.54

Table 1: Statistics about the subcorpora in the BioScope corpus and the hedge scopes ("Av". stands for *average*).

The texts have been processed with the GENIA tagger (Tsuruoka and Tsujii, 2005; Tsuruoka et al., 2005), a bidirectional inference based tagger that analyzes English sentences and outputs the base forms, part-of-speech tags, chunk tags, and named entity tags in a tab-separated format. Additionally, we converted the annotation about scope of negation into a token-per-token representation, following the standard format of the 2006 CoNLL Shared Task (Buchholz and Marsi, 2006), where sentences are separated by a blank line and fields are separated by a single tab character. A sentence consists of a sequence of tokens, each one starting on a new line.

[2]Web page: `www.inf.u-szeged.hu/rgai/bioscope`.

4 Finding the scope of hedge cues

We model this task in the same way that we modelled the task for finding the scope of negation (Morante and Daelemans, 2009), i.e., as two consecutive classification tasks: a first one that consists of classifying the tokens of a sentence as being at the beginning of a hedge signal, inside or outside. This allows the system to find multiword hedge cues. The second classification task consists of classifying the tokens of a sentence as being the first element of the scope, the last, or neither. This happens as many times as there are hedge cues in the sentence.

5 System description

The two classification tasks (identifying hedge cues and finding the scope) are implemented using supervised machine learning methods trained on part of the annotated corpus.

5.1 Identifying hedge cues

In this phase, a classifier predicts for all tokens in a sentence whether a token is the first token of a hedge cue (B-cue), inside a hedge cue (I-cue), or outside of it (O-cue). For sentence (3) the system assigns the B-cue class to *indicate*, the I-cue class to *that* and the O-cue class to the rest of tokens.

(3) These results *indicate that* a component or components of NF–AT have the potential to reconstitute NF(P)

The instances represent all tokens in the corpus and they have features about the token: lemma, word, part-of-speech (POS) and IOB[3] chunk tag; and features about the token context: Word, POS and IOB chunk tag of 3 tokens to the right and 3 to the left.

We use IGTREE as implemented in TiMBL (version 6.1.2) (Daelemans et al., 2007). We also experimented with IB1, but it produced lower results. The classifier was parameterised by using gain ratio for feature weighting. According to the gain ratio scores, the most informative features are the lemma and word of the token in focus, followed by the word of the token to the right and of the token to the left.

We performed two experiments. In one, the test file is preprocessed using a list of hedge cues extracted from the training corpus. The list comprises the following hedge cues listed in (4). Instances with these hedge cues are directly assigned their class. The classifier predicts the class of the rest of tokens. In the other experiment we don't preprocess the test file.

(4) appear, apparent, apparently, believe, either, estimate, hypothesis, hypothesize, if, imply, likely, may, might, or, perhaps, possible, possibly, postulate, potential, potentially, presumably, probably, propose, putative, should, seem, speculate, suggest, support, suppose, suspect, think, uncertain, unclear, unkwown, unlikely, whether, would

5.2 Scope finding

In this phase three classifiers predict for all tokens in the sentence whether a token is the first token in the scope sequence (F-scope), the last (L-scope), or neither (NONE). For the sentence in 3, the classifiers assign the class F-scope to *indicate*, L-scope to *NF(P)*, and NONE to the rest of tokens. A fourth classifier is a metalearner that uses the predictions of the three classifiers to predict the scope classes. An instance represents a pair of a hedge cue and a token from the sentence. This means that all tokens in a sentence are paired with all hedge cues that occur in the sentence. Hedge cues are those that have been classified as such in the previous phase. Only sentences that have hedge cues are selected for this phase. The three object classifiers that provide input to the metalearner were trained using the following machine learning methods:

- Memory-based learning as implemented in TiMBL (Daelemans et al., 2007), a supervised inductive algorithm for learning classification tasks based on the k-nearest neighbor classification rule (Cover and Hart, 1967). In this lazy learning approach, all training data is kept in memory and classification of a new item is achieved by extrapolation from the most similar remembered training items.
- Support vector machines (SVM) as implemented in SVM^{light}V6.01 (Joachims, 1999). SVMs are defined on a vector space and try to find a decision surface that best separates the data points into two classes. This is achieved by using quadratic programming techniques. Kernel functions can be used to map the original vectors to a higher-dimensional space that is linearly separable.

[3] *I* stands for 'inside', *B* for 'beginning', and *O* for 'outside'.

- Conditional random fileds (CRFs) as implemented in CRF++-0.51 (Lafferty et al., 2001). CRFs define a conditional probability distribution over label sequences given a particular observation sequence rather than a joint distribution over label and observation sequences, and are reported to avoid the label bias problem of HMMs and other learning approaches.

The memory-based learning algorithm was parameterised in this case by using overlap as the similarity metric, gain ratio for feature weighting, using 7 k-nearest neighbors, and weighting the class vote of neighbors as a function of their inverse linear distance. The SVM was parameterised in the learning phase for classification, cost factor of 1 and biased hyperplane, and it used a linear kernel function. The CRFs classifier used regularization algorithm L2 for training, the hyper-parameter and the cut-off threshold of features were set to 1.

We have used the same features used for the system that finds the scope of negation. The features of the first three classifers are:

- Of the hedge signal: Chain of words.
- Of the paired token: Lemma, POS, chunk IOB tag, type of chunk; lemma of the second and third tokens to the left; lemma, POS, chunk IOB tag, and type of chunk of the first token to the left and three tokens to the right; first word, last word, chain of words, and chain of POSs of the chunk of the paired token and of two chunks to the left and two chunks to the right.
- Of the tokens between the hedge cue and the token in focus: Chain of POS types, distance in number of tokens, and chain of chunk IOB tags.
- Others: A feature indicating the location of the token relative to the hedge cue (pre, post, same).

The fourth classifier, a metalearner, is also a CRFs as implemented in CRF++. The features of this classifier are:

- Of the hedge signal: Chain of words, chain of POS, word of the two tokens to the right and two tokens to the left, token number divided by the total of tokens in the sentence.
- Of the paired token: Lemma, POS, word of two tokens to the right and two tokens to the left, token number divided by the total of tokens in the sentence.
- Of the tokens between the hedge cue and the token in focus: Binary features indicating if there are commas, colons, semicolons, verbal phrases or one of the following words between the hedge cue and the token in focus: *Whereas, but, although, nevertheless, notwithstanding, however, consequently, hence, therefore, thus, instead, otherwise, alternatively, furthermore, moreover*.
- About the predictions of the three classifiers: prediction, previous and next predictions of each of the classifiers, full sequence of previous and full sequence of next predictions of each of the classifiers.
- Others: A feature indicating the location of the token relative to the hedge cue (pre, post, same).

Hedge cues in the BioScope corpus always scope over a consecutive block of tokens, including the cue token itself. However, the classifiers only predict the first and last element of the scope. We need to process the output of the classifers in order to build the complete sequence of tokens that constitute the scope. We apply the following postprocessing:

(5) - If one token has been predicted as FIRST and one as LAST, the sequence is formed by the tokens between first and last.

- If one token has been predicted as FIRST and none has been predicted as LAST, the sequence is formed by the token predicted as FIRST.

- If one token has been predicted as LAST and none as FIRST, the sequence will start at the hedge cue and it will finish at the token predicted as LAST.

- If one token has been predicted as FIRST and more than one as LAST, the sequence will end with the first token predicted as LAST after the token predicted as FIRST, if there is one.

- If one token has been predicted as LAST and more than one as FIRST, the sequence will start at the hedge signal.

- If no token has been predicted as FIRST and more than one as LAST, the sequence will start at the hedge cue and will end at the first token predicted as LAST after the hedge signal.

6 Results

The results provided for the abstracts part of the corpus have been obtained by performing 10-fold cross validation experiments, whereas the results provided

for papers and clinical reports have been obtained by training on the full abstracts subcorpus and testing on the papers and clinical reports subcorpus. The latter experiment is therefore a test of the robustness of the system when applied to different text types within the same domain. The evaluation is made using the precision and recall measures (Van Rijsbergen, 1979), and their harmonic mean, F-score. We report micro F1.

In the hedge finding task, a hedge token is correctly classified if it has been classified as being at the beginning or inside the hedge signal. We also evaluate the percentage of hedge cues that have been correctly identified. In the scope finding task, a token is correctly classified if it has been correctly classified as being inside or outside of the scope of all the hedge cues that there are in the sentence. This means that when there is more than one hedge cue in the sentence, the token has to be correctly assigned a class for as many hedge signals as there are. Additionally, we evaluate the percentage of correct scopes (PCS). A scope is correct if all the tokens in the sentence have been assigned the correct scope class for a specific hedge signal. The evaluation in terms of precision and recall measures takes as unit a token, whereas the evaluation in terms of PCS takes as unit a scope.

6.1 Hedge cue finding

An informed baseline system has been created by tagging as hedge cues the tokens with the words listed in (4) above. The list has been extracted from the training corpus. The results are shown in Table 2.

Corpus	Prec.	Recall	F1	% Correct
Abstracts	55.62	71.77	62.67	70.91
Papers	54.39	61.21	57.60	64.46
Clinical	66.55	40.78	50.57	51.38

Table 2: Baseline results of the hedge finding system.

The fact that the results are lower for the papers and clinical subcorpora can be explained by the fact that the list of cues has been extracted from the training corpus.

Table 3 shows the results of the system. The results of the system for abstracts and papers are higher than baseline, but for clinical they are lower. This is due to the fact that in the baseline system the hedge cue *or* that accounts for 24.53 % of the hedge cues is 100 % correct, whereas the system achieves only 0.72 % of correct predictions. The score obtained by *or* is also the reason why the system produces lower results for the clinical subcorpus.

Corpus	Prec.	Recall	F1	% Correct
Abstracts	90.81	79.84	84.77	78.67
Papers	75.35	68.18	71.59	69.86
Clinical	88.10	27.51	41.92	33.36

Table 3: Results of the hedge finding system without preprocessing.

Table 4 shows the results of the system with preprocessing. In terms of % of correct cues, the system that uses a preprocessed test set gets higher scores, but in terms of F1 it gets lower results, except for the clinical subcorpus. The drop in F1 of this system is caused by a drop in precision due to the excess of false positives.

Corpus	Prec.	Recall	F1	% Correct
Abstracts	60.74	94.83	74.05	96.03
Papers	56.56	84.03	67.61	88.60
Clinical	71.25	52.33	60.34	64.49

Table 4: Results of the hedge finding system with preprocessing.

In the abstracts subcorpus the hedge cue that has the biggest proportion of false positives is *or*. Of the 1062 accurrences of *or*, in 88.32% of the cases *or* is not a hedge cue. The system that uses preprocessing produces 938 false positives and 4 false negatives, whereas the other system produces 21 false positives and 108 false negatives. In the papers subcorpus, the hedge cues *if, or, can, indicate* and *estimate* cause 67.38% of the false positives. In the clinical subcorpus the hedge cues *evidence, evidence of, no* and *appear* cause 88.27% of the false positives. In contrast with the abstracts subcorpus, the hedge cue *or* has only 5 false positives and scores an F1 of 99.10. So, in the clinical corpus *or* is not ambiguous, whereas in the abstracts subcorpus it is very ambiguous. An example of *or* as hedge cue in the clinical subcorpus is shown in (6). An example of *or* as hedge cue in the abstracts subcorpus is shown in (7), and as a non cue in (8).

(6) Findings compatible with reactive airway disease or viral lower respiratory tract infection.

(7) Nucleotide sequence and PCR analyses demonstrated the presence of novel duplications or deletions involving the NF-kappa B motif.

(8) In nuclear extracts from monocytes or macrophages, induction of NF-KB occurred only if the cells were previously infected with HIV-1.

Compared to negation cues, hedge cues are more varied and more ambiguous. Both the system without and with preprocessing for negation finding performed better than the hedge finding system.

6.2 Scope finding

An informed baseline system has been created by calculating the average length of the scope to the right of the hedge cue in each corpus and tagging that number of tokens as scope tokens. We take the scope to the right for the baseline because it is much more frequent than the scope to the left, as is shown by the statistics contained in Table 1 of Section 3. Baseline results are presented in Table 5. The low PCS for the three subcorpora indicates that finding the scope of hedge cues is not a trivial task. The fact that, despite a very low PCS, precision, recall and F1 are relatively high indicates that these measures are in themselves not reliable to evaluate the performance of the system.

Corpus	Prec.	Recall	F1	PCS
Abstracts	78.92	62.19	69.56	3.15
Papers	72.03	50.43	59.33	2.19
Clinical	64.92	25.10	36.20	2.72

Table 5: Baseline results of the scope finding system.

The upper-bound results of the metalearner system assuming gold standard identification of hedge cues are shown in Table 6.

Corpus	Prec.	Recall	F1	PCS	PCS-2
Abstracts	89.71	89.09	89.40	77.13	78.21
Papers	77.78	77.10	77.44	47.94	58.21
Clinical	79.16	78.13	78.64	60.59	63.94

Table 6: Results of the scope finding system with gold-standard hedge signals.

The percentage of correct scopes has been measured in two ways: PCS measures the proportion of correctly classified tokens in the scope sequence, whereas PCS-2 measures the proportion of nouns and verbs that are correctly classifed in the scope sequence. This less strict way of computing correctness is motivated by the fact that being able to determine the concepts and relations that are speculated (indicated by content words) is the most important use of the hedge scope finder.

Results show that the system achieves a high percentage of fully correct scopes, and that, although performance is lower for the papers and clinical corpora, the system is portable. Table 7 shows the results of the negation scope finding system also with gold standard negation cues. The comparison of results shows that for abstracts and papers the scores are higher for the hedge system, which means that the system can be used for finding both types of scope.

Corpus	Prec.	Recall	F1	PCS	PCS-2
Abstracts	90.68	90.68	90.67	73,36	74.10
Papers	84.47	84.95	84.71	50.26	54.23
Clinical	91.65	92.50	92.07	87.27	87.95

Table 7: Results of the negation scope finding system with gold-standard negation signals.

The results of the hedge system with predicted hedge cues are presented in Table 8. The hedge cues have been predicted by the system without the preprocessing step presented in Subsection 6.1.

Corpus	Prec.	Recall	F1	PCS	PCS-2
Abstracts	85.77	72.44	78.54	65.55	66.10
Papers	67.97	53.16	59.66	35.92	42.37
Clinical	68.21	26.49	38.16	26.21	27.44

Table 8: Results of the scope finding system with predicted hedge signals.

In terms of PCS, which is a scope based measure, results are considerably higher than baseline results, whereas in terms of precision, recall and F1, which are token based measures, results are lower. Evaluating the system in terms of a more relaxed measure (PCS-2) does not reflect a significant increase in its performance. This suggests that when a scope is incorrectly predicted, main content tokens are also incorrectly left out of the scope or added.

Results also show that the system based on predicted hedge cues performs lower for all corpora,

which is also a trend observed for the negation scope finding system. The difference in performance for abstracts and papers follows the same trends as in the negation system, whereas the drop in performance for the clinical subcorpus is bigger. This can be explained by the results obtained in the cues finding phase, where the clinical subcorpus obtained only 41.92% F1. However, gold standard results show that if the hedge cues are identified, then the system is portable.

	Abstracts		Papers		Clinical	
	#	**PCS**	**#**	**PCS**	**#**	**PCS**
appear	143	58.04	39	28.20	-	-
can	48	12.5	25	0.00	22	0.00
consistent with	-	-	-	-	67	0.00
could	67	11.94	28	14.28	36	22.22
either	28	0.00	-	-	-	-
evaluate for	-	-	-	-	86	3.84
imply	21	90.47	-	-	-	-
indicate	23	73.91	-	-	-	-
indicate that	276	89.49	-	-	-	-
likely	59	59.32	36	30.55	63	66.66
may	516	81.39	68	54.41	107	80.37
might	72	73.61	40	35.00	-	-
or	120	0.00	-	-	276	0.00
possible	50	66.00	24	54.16	26	80.76
possibly	25	52.00	-	-	-	-
potential	45	28.88	-	-	-	-
potentially	21	52.38	-	-	-	-
propose	38	63.15	-	-	-	-
putatitve	39	17.94	-	-	-	-
rule out	-	-	-	-	61	0.00
suggest	613	92.33	70	62.85	64	90.62
think	35	31.42	-	-	-	-
unknown	26	15.38	-	-	-	-
whether	96	72.91	-	-	-	-
would	-	-	21	28.57	-	-

Table 9: PCS per hedge cue for hedge cues that occur more than 20 times in one of the subcorpus.

Table 9 shows the PCS results per hedge cue. The cues that get better scores in the clinical and papers subcorpora are cues that appear in the abstracts subcorpus and get a good score. Cues that occur in the clinical subcorpus and do not occur in the abstracts (training) subcorpus, get 0.00 score or close to 0.00, whereas cues that appear in both subcorpora tend to get a similar or better score in the clinical subcorpus. This is a trend that we also observed in the negation scope finding system. As with that system, we also observed that the papers subcorpus tends to get lower scores than the abstracts subcorpus.

The results of the system based on gold standard hedge cues showed that the system can be applied to negation scope finding and hedge scope finding, but these results show that the results of the second phase of the system depend on the results of the first phase of the system, and that finding hedge cues is a domain dependent task. The cues that are not present in the training data cannot be learned in the test data and the same applies to their scope. This observation is consistent with the observation that the portability of hedge classifiers is limited, made by Szarvas (Szarvas, 2008).

7 Conclusions

In this paper we have presented a metalearning approach to processing the scope of hedge cues, based on a system that finds the scope of negation cues. We have shown that the same system can find both the scope of negation and hedge cues. The performance of the system is evaluated in terms of percentage of correct scopes on three text types.

In the hedge finding phase, the system achieves an F1 of 84.77% in the abstracts subcorpus. Existing systems that classify sentences as speculative or not reach an 85.00 BEP. Although the tasks are different, we consider that the results of our system are competitive. In the scope finding phase, the system that uses predicted hedge cues achieves 65.55% PCS in the abstracts corpus, which is very similar to the result obtained by the negation scope finding system with predicted negation cues (66.07% PCS). However, the results for the papers and clinical subcorpora are considerably lower than the results for the abstracts subcorpus in the two phases. In the case of the negation scope finding system, the evaluation on the clinical subcorpus yielded a 4.23% PCS higher result, whereas in the case of the hedge scope finding system the results are almost 30.00% PCS lower, confirming the observation that the portability of hedge classifers is limited. Future research will focus on trying to improve the first phase of the system and anlysing errors in depth in order to get insights into how to get a better performance.

Acknowledgments

Our work was made possible through financial support from the University of Antwerp (GOA project BIOGRAPH). We are thankful to three anonymous reviewers for their valuable comments.

References

S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proc. of the X CoNLL Shared Task*, New York. SIGNLL.

N. Collier, H.S. Park, N. Ogata, Y. Tateisi, C. Nobata, T. Sekimizu, H. Imai, and J. Tsujii. 1999. The GENIA project: corpus-based knowledge acquisition and information extraction from genome research papers. In *Proc. of EACL 1999*.

T. M. Cover and P. E. Hart. 1967. Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13:21–27.

W. Daelemans, J. Zavrel, K. Van der Sloot, and A. Van den Bosch. 2007. TiMBL: Tilburg memory based learner, version 6.1, reference guide. Technical Report Series 07-07, ILK, Tilburg, The Netherlands.

C. Di Marco and R.E. Mercer, 2005. *Computing attitude and affect in text: Theory and applications*, chapter Hedging in scientific articles as a means of classifying citations. Springer-Verlag, Dordrecht.

C. Friedman, P. Alderson, J. Austin, J.J. Cimino, and S.B. Johnson. 1994. A general natural–language text processor for clinical radiology. *JAMIA*, 1(2):161–174.

K. Hyland. 1998. *Hedging in scientific research articles*. John Benjamins B.V, Amsterdam.

T. Joachims, 1999. *Advances in Kernel Methods - Support Vector Learning*, chapter Making large-Scale SVM Learning Practical, pages 169–184. MIT-Press, Cambridge, MA.

H. Kilicoglu and S. Bergler. 2008. Recognizing speculative language in biomedical research articles: a linguistically motivated perspective. *BMC Bioinformatics*, 9(Suppl 11):S10.

M. Krallinger, F. Leitner, C. Rodriguez-Penagos, and A. Valencia. 2008a. Overview of the protein–protein interaction annotation extraction task of BioCreative II. *Genome Biology*, 9(Suppl 2):S4.

M. Krallinger, A. Valencia, and L. Hirschman. 2008b. Linking genes to literature: text mining, information extraction, and retrieval applications for biology. *Genome Biology*, 9(Suppl 2):S8.

M. Krauthammer, P. Kra, I. Iossifov, S.M. Gomez, G. Hripcsak, V. Hatzivassiloglou, C. Friedman, and A.Rzhetsky. 2002. Of truth and pathways: chasing bits of information through myriads of articles. *Bioinformatics*, 18(Suppl 1):S249–57.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML 2001*, pages 282–289.

G. Lakoff. 1972. Hedges: a study in meaning criteria and the logic of fuzzy concepts. *Chicago Linguistics Society Papers*, 8:183–228.

M. Light, X.Y.Qiu, and P. Srinivasan. 2004. The language of bioscience: facts, speculations, and statements in between. In *Proc. of the BioLINK 2004*, pages 17–24.

B. Medlock and T. Briscoe. 2007. Weakly supervised learning for hedge classification in scientific literature. In *Proc. of ACL 2007*, pages 992–999.

B. Medlock. 2008. Exploring hedge identification in biomedical literature. *JBI*, 41:636–654.

T. Mitsumori, M. Murata, Y. Fukuda, K Doi, and H. Doi. 2006. Extracting protein-protein interaction information from biomedical text with svm. *IEICE - Trans. Inf. Syst.*, E89-D(8):2464–2466.

R. Morante and W. Daelemans. 2009. A metalearning approach to processing the scope of negation. In *Proc. of CoNLL 2009*, Boulder, Colorado.

F.R. Palmer. 1986. *Mood and modality*. CUP, Cambridge, UK.

R. Saurí, M. Verhagen, and J. Pustejovsky. 2006. Annotating and recognizing event modality in text. In *Proc. of FLAIRS 2006*, pages 333–339.

G. Szarvas, V. Vincze, R. Farkas, and J. Csirik. 2008. The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In *Proc. of BioNLP 2008*, pages 38–45, Columbus, Ohio. ACL.

G. Szarvas. 2008. Hedge classification in biomedical texts with a weakly supervised selection of keywords. In *Proc. of ACL 2008*, pages 281–289, Columbus, Ohio, USA. ACL.

P. Thompson, G. Venturi, J. McNaught, S. Montemagni, and S. Ananiadou. 2008. Categorising modality in biomedical texts. In *Proc. of the LREC 2008 Workshop on Building and Evaluating Resources for Biomedical Text Mining 2008*, pages 27–34, Marrakech. LREC.

Y. Tsuruoka and J. Tsujii. 2005. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proc. of HLT/EMNLP 2005*, pages 467–474.

Y. Tsuruoka, Y. Tateishi, J. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii, 2005. *Advances in Informatics - 10th Panhellenic Conference on Informatics*, volume 3746 of *LNCS*, chapter Part-of-Speech Tagger for Biomedical Text, Advances in Informatics, pages 382–392. Springer, Berlin/Heidelberg.

C.J. Van Rijsbergen. 1979. *Information Retrieval*. Butterworths, London.

V. Vincze, G. Szarvas, R. Farkas, G. Móra, and J. Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9(Suppl 11):S9.

How Feasible and Robust is the Automatic Extraction of Gene Regulation Events ? A Cross-Method Evaluation under Lab and Real-Life Conditions

Udo Hahn[1] **Katrin Tomanek**[1] **Ekaterina Buyko**[1] **Jung-jae Kim**[2] **Dietrich Rebholz-Schuhmann**[2]

[1]Jena University Language & Information Engineering (JULIE) Lab
Friedrich-Schiller-Universität Jena, Germany
{udo.hahn|katrin.tomanek|ekaterina.buyko}@uni-jena.de

[2]EMBL-EBI, Wellcome Trust Genome Campus, Hinxton, Cambridge, UK
{kim|rebholz}@ebi.ac.uk

Abstract

We explore a rule system and a machine learning (ML) approach to automatically harvest information on gene regulation events (GREs) from biological documents in two different evaluation scenarios – one uses self-supplied corpora in a clean lab setting, while the other incorporates a standard reference database of curated GREs from REGULONDB, real-life data generated independently from our work. In the lab condition, we test how feasible the automatic extraction of GREs really is and achieve F-scores, under different, not directly comparable test conditions though, for the rule and the ML systems which amount to 34% and 44%, respectively. In the REGULONDB condition, we investigate how robust both methodologies are by comparing them with this routinely used database. Here, the best F-scores for the rule and the ML systems amount to 34% and 19%, respectively.

1 Introduction

The extraction of binary relations from biomedical text has caught much attention in the recent years. Progress on this and other tasks has been monitored in challenge competitions such as BIOCREATIVE I and II,[1] which dealt with gene/protein names and and protein-protein interaction.

The BIOCREATIVE challenge and other related ones have shown at several occasions that participants continue to use two fundamentally different systems: symbolic pattern-based systems (rule systems), on the one hand, and feature-based statistical machine learning (ML) systems, on the other hand. This has led to some rivalry with regard to the interpretation of their performance data, the costs of human efforts still required and their scalability for the various tasks. While rule systems are often hand-crafted and fine-tuned to a particular application (making a major manual rewrite often necessary when the application area is shifted), ML systems are trained automatically on manually annotated corpora, i.e., without manual intervention, and thus have the advantage to more easily adapt to changes in the requested identification tasks. Time costs (human workload) are thus shifted from rule design and adaptation to metadata annotation.

Text mining systems as usually delivered by BioNLP researchers render biologically relevant entities and relations on a limited set of test documents only. While this might be sufficient for the BioNLP community, it is certainly insufficient for bioinformaticians and molecular biologists since they require large-scale data with high coverage and reliability. For our analysis, we have chosen the topic of gene regulatory events in E. coli, which is a domain of very active research and grand challenges.[2] Currently the gold standard of the existing body of knowledge of such events is represented by the fact database REGULONDB.[3] Its content has been man-

[1]http://biocreative.sourceforge.net/

[2]The field of gene regulation is one of the most prominent topics of research and often mentioned as one of the core fields of future research in molecular biology (cf, e.g., the Grand Challenge I-2 described by Collins et al. (2003)).

[3]http://regulondb.ccg.unam.mx/

Proceedings of the Workshop on BioNLP, pages 37–45,
Boulder, Colorado, June 2009.

ually gathered from the scientific literature and describes the curated computational model of mechanisms of transcriptional regulation in E. coli. Having this gold standard in mind, we face the challenging task to automatically reproduce this content from the available literature, to enhance this content with reliable additional information and to update this resource as part of a regular automatic routine.

Hence, we first explore the *feasibility* and performance of a rule-based and an ML-based system against special, independently created corpora that were generated to enable measurements under clean experimental lab conditions. This part, due to different experimental settings, is not meant as a comparison between both approaches though. We then move to the even more demanding real-life scenario where we evaluate and compare these solutions for the identification of gene regulatory events against the REGULONDB data resource. This approach targets the *robustness* of the proposed text mining solutions from the perspectives of completeness, correctness and novelty of the generated results.

2 Related Work

Considering relation extraction (RE) in the biomedical domain, there are only few studies which deal primarily with gene regulation. Yang et al. (2008) focus on the detection of sentences that contain mentions of transcription factors (proteins regulating gene expression). They aim at the detection of new transcription factors, while relations are not taken into account. In contrast, Šarić et al. (2004) extract gene regulatory networks and achieve in the RE task an accuracy of up to 90%. They disregard, however, ambiguous instances, which may have led to the low recall around 20%. The *Genic Interaction Extraction Challenge* (Nédellec, 2005) was organized to determine the state-of-the-art performance of systems designed for the detection of gene regulation interactions. The best system achieved a performance of about 50% F-score. The results, however, have to be taken with care as the LLL corpus used in the challenge is of extremely limited size.

3 Extraction of Gene Regulation Events

Gene regulation is a complex cellular process that controls the expression of genes. These genes are then transcribed into their RNA representation and later translated into proteins, which fulfill various tasks such as maintaining the cell structure, enabling the generation of energy and interaction with the environment.

The analysis of the gene regulatory processes is ongoing research work in molecular biology and affects a large number of research domains. In particular the interpretation of gene expression profiles from microarray analyses could be enhanced using our understanding of gene regulation events (GREs) from the literature.

We approach the task of the automatic extraction of GREs from literature from two different methodological angles. On the one hand, we provide a set of hand-crafted rules – both for linguistic analysis and conceptual inference (cf. Section 3.1), the latter being particularly helpful in unveiling only implicitly stated biological knowledge. On the other hand, we supply a machine learning-based system for event extraction (cf. Section 3.2). No regularities are specified *a priori* by a human although, at least in the supervised scenario we have chosen, this approach relies on training data supplied by human (expert) annotators who provide sufficiently many instances of ground truth decisions from which regularities can automatically be learnt. At the level of system performance, rules tend to foster precision at the cost of recall and ML systems tend to produce inverse figures, while there is no conclusive evidence for or against any of these two approaches.

The extraction of GREs, independent of the approach one subscribes to, is a complex problem composed of a series of subtasks. Abstracting away from lots of clerical and infrastructure services (e.g., sentence splitting, tokenization) at the core of any GRE extraction lie the following basic steps:

- the identification of pairs of gene mentions as the arguments of a relation – the well-known named entity recognition and normalization task,
- the decision whether the entity pair really constitutes a relation,
- and the identification of the roles of the arguments in the relation which implicitly amounts to characterize each argument as either agent or patient.

3.1 Rule-based Extraction

The rule-based system extracts GREs from text employing logical inference. The motivation of using inference is that the events under scrutiny are often expressed in text in either a compositional or an incomplete way. We address this issue by compositionally representing textual semantics and by logically inferring implicit meanings of text over the compositional representation of textual semantics.

Entity Identification. The system first recognizes named entities of the types that can be participants of the target events. We have collected 15,881 E. coli gene/protein and operon names from REGULONDB and UNIPROT. Most of the gene/protein names are associated with UNIPROT identifiers. An operon in prokaryotes is a DNA sequence with multiple genes whose expression is controlled by a shared promoter and which thus express together. We have mapped the operon names to corresponding gene sets.

Named entity recognition relies on the use of dictionaries. If the system recognizes an operon name, it then associates the operon with its genes. The system further recognizes multi-gene object names (e.g., "acrAB"), divides them into individual gene names (e.g., "acrA", "acrB") and associates the gene names with the multi-gene object names.

Relation Identification. The system then identifies syntactic structures of sentences in an input corpus by utilizing the ENJU parser (Sagae et al., 2007). The ENJU parser generates predicate-argument structures, and the system converts them into dependency structures.

The system then analyzes the semantics of the sentences by matching syntactic-semantic patterns to the dependency structures. We constructed 1,123 patterns for the event extraction according to the following workflow. We first collected keywords related to gene regulation, from GENE ONTOLOGY, INTERPRO, WORDNET, and several papers about information extraction from biomedical literature (Hatzivassiloglou and Weng, 2002; Kim and Park, 2004; Huang et al., 2004). Then we collected subcategorization frames for each keyword and created patterns for the frames manually.

Each pattern consists of a syntactic pattern and a semantic pattern. The syntactic patterns comply with dependency structures. The system tries to match the syntactic patterns to the dependency structures of sentences in a bottom-up way, considering syntactic and semantic restrictions of syntactic patterns. Once a syntactic pattern is successfully matched to a sub-tree of the available dependency structure, its corresponding semantic pattern is assigned to the sub-tree as one of its semantics. The semantic patterns are combined according to the dependency structures to form a compositional semantic structure.

The system then performs logical inference over the semantic structures by using handcrafted inference rules and extracts target information from the results of the inference. We have manually created 28 inference rules that reflect the knowledge of the gene regulation domain. Only relations where the identified agent is one of those known TFs are kept, while all others are discarded.

3.2 Generic, ML-based Extraction

Apart from the already mentioned clerical preprocessing steps, the ML-based extraction of GREs requires several additional syntactic processing steps including POS-tagging, chunking, and full dependency- and constituency-based parsing.[4]

Entity Identification. To identify gene names in the documents, we applied GENO, a multi-organism gene name recognizer and normalizer (Wermter et al., 2009) which achieved a top-rank performance of 86.4% on the gene normalization task of BIOCREATIVE-II. GENO recognizes gene mentions by means of an ML-based named entity tagger trained on publicly available corpora. Subsequently, it attempts to map all identified mentions to organism-specific UNIPROT[5] identifiers. Mentions that cannot be mapped are discarded; only successfully mapped mentions are kept. We utilized GENO in its original version, i.e., without special adjustments to the E. coli organism. However, only those mentions detected to be genes of E. coli were fed into the relation extraction component.

[4] These tasks were performed with the OPENNLP tools (`http://opennlp.sourceforge.net/`) and the MST parser (`http://sourceforge.net/projects/mstparser`), both retrained on biomedical corpora.

[5] `http://www.uniprot.de`

Relation Identification. The ML-based approach to GRE employs Maximum Entropy models and constitutes and extension of the system proposed by Buyko et al. (2008) as it also makes use of dependency parse information including dependency tree level features (Katrenko and Adriaans, 2006) and shortest dependency path features (Kim et al., 2008). In short, the feature set consists of:

- word features (covering words before, after and between both entity mentions);
- entity features (accounting for combinations of entity types, flags indicating whether mentions have an overlap, and their mention level);
- chunking and constituency-based parsing features (concerned with head words of the phrases between two entity mentions; this class of features exploits constituency-based parsing as well and indicates, e.g., whether mentions are in the same NP, PP or VP);
- dependency parse features (analyzing both the dependency levels of the arguments as discussed by Katrenko and Adriaans (2006) and dependency path structure between the arguments as described by Kim et al. (2008));
- and relational trigger (key)words (accounting for the connection of trigger words and mentions in a full parse tree).

An advantage of ML-based systems is that they allow for thresholding. To achieve higher recall values for our system, we may set the confidence threshold for the negative class (i.e., a pair of entity mentions does not constitute a relation) to values > 0.5. Clearly, this is at the cost of precision as the system more readily assigns the positive class.

4 Intrinsic Evaluation of Feasibility

The following two sections aim at evaluating the rule-based and ML-based GRE extraction systems. The systems are first "intrinsically" evaluated, i.e., in a cross-validation manner on corpora annotated with respect to GREs. Second, in a more realistic scenario, both systems were evaluated against REGULONDB, a database collecting knowledge about gene regulation in E. coli. This scenario tests which part of manually accumulated knowledge about gene regulation in E. coli can automatically be identified by our systems and at what level of quality.

4.1 Rule-based system

Corpus. For the training and evaluation of the rule-based system, we annotated 209 MEDLINE abstracts with three types of events: specific events of gene transcription regulation, general events of gene expression regulation, and physical events of binding of transcription factors to gene regulatory regions. Strictly speaking, only the first type is relevant to REGULONDB. However, biologists often report gene transcription regulation events in the scientific literature as if they are gene expression regulation events, which is a generalization of gene transcription regulation, or the binding event, which itself is insufficient evidence for gene transcription regulation. The two latter types may indicate that the full-texts contain evidence of the first type.

We asked two curators to annotate the abstracts. Curator A was trained with example annotations and interactive discussions. Curator B was trained only with example annotations and guidelines. For cross-checking of annotations, we asked them to annotate an unseen corpus of 97 abstracts and found that Curator A made 10.8% errors, misjudging three event additions and, in the other 14 errors, mistaking in annotating event types, event attributes, and passage boundaries, while Curator B made 32.4% errors as such. This result indicates that the annotation of GREs requires intensive and interactive training. The curators have discussed and agreed on the final release of the corpora.[6]

Results. The system has successfully extracted 79 biologically meaningful events among them (21.1% recall) and incorrectly produced 15 events (84.0% precision) which constitutes an overall F-score of 33.6%. Among the 79 events, the system has correctly identified event types of 39 events (49.4% precision), polarity of 46 events (58.2% precision), and directness of 51 events (64.6% precision). Note that the system employed a fully automatic module for named entity recognition. The event type recognition is impaired, because it often fails to recognize

[6]The resultant annotated corpora are available at `http://www.ebi.ac.uk/~kim/eventannotation/`.

the specific event type of transcription regulation, but only identifies the general event type of gene expression regulation due to the lack of identified evidence.

4.2 ML-based system

GeneReg corpus. The GENEREG corpus (Buyko et al., 2008) constitutes a selection of 314 MEDLINE abstracts related to gene regulation in E. coli. These abstracts were randomly drawn from a set of 32,155 selected by MESH term queries from MEDLINE using keywords such as *Escherichia coli*, *Gene Expression* and *Transcription Factors*. These 314 abstracts were manually annotated for named entities involved in gene regulatory processes (such as transcription factor, including co-factors and regulators, and genes) and pairwise relations between transcription factors (TFs) and genes, as well as triggers (e.g., clue verbs) essential for the description of gene regulation relations. As for the relation types, the GENEREG corpus distinguishes between (a) unspecified regulation of gene expression, (b) positive, and (c) negative regulation of gene expression. Out of the 314 abstracts a set of 65 were randomly selected and annotated by a second annotator to identify inter-annotator agreement (IAA) values. For the task of correct identification of the pair of interacting named entities in gene regulation processes, an IAA of 78.4% (R), 77.3% (P), 77.8% (F) was measured , while 67% (R), 67.9% (P), 67.4% (F) were achieved for the identification of interacting pairs plus the 3-way classification of the interaction relation.

Experimental Setting. The ML-based extraction system merges all of the above mentioned three types (unspecific, negative and positive) into one common type "relation of gene expression". So, it either finds that there is a relation of interest between a pair of gold entity mentions or not. We evaluated our system by a 5-fold cross-validation on the GENEREG corpus. The fold splits were done on the abstract-level to avoid the otherwise unrealistic scenario where a system is trained on sentences from an abstract and evaluated on other sentences but from the same abstract (Pyysalo et al., 2008). As our focus here is only on the performance of the GRE extraction component, gold entity mentions as annotated in the respective corpus were used.

Results. For the experimental settings given above, the system achieved an F-score of 42% with a precision of 59% and a recall of 33%. Increasing the confidence threshold for the negative class increases recall as shown for two different thresholds in Table 1. As expected this is at the cost of precision. It shows, that using an extremely high threshold of 0.95 results in a dramatically increased recall of 73% compared to 33% with the default threshold. Although at the cost of diminished precision of 32% compared to originally 59%, the lifted threshold increases the overall F-score (44%) by 2 points.

threshold	R	P	F
default (0.5)	0.33	0.59	0.42
0.80	0.54	0.43	0.48
0.95	0.73	0.32	0.44

Table 1: Different confidence thresholds for the ML-based system achieved by intrinsic evaluation

5 Extrinsic Evaluation of Robustness

REGULONDB is the primary and largest reference database providing manually curated knowledge of the transcriptional regulatory network of E. coli K12. On K12, approximately for one-third of K12's genes, information about their regulation is available. REGULONDB is updated with content from recent research papers on this issue. While REGULONDB contains much more, for this paper our focus was solely on REGULONDB's information about gene regulation events in E. coli. In the following, the term REGULONDB refers to this part of the REGULONDB database. REGULONDB includes e.g., the following information for each regulation event: regulatory gene (the "agent" in such an event, a transcription factor), the regulated gene (the "patient"), the regulatory effect on the regulated gene (activating, suppression, dual, unknown), and evidence that supports the existence of the regulatory interaction.

Evaluation against REGULONDB constitutes a real-life scenario. Thus, the complete extraction systems were run, including gene name recognition and normalization as well as relation detection. Hence, the systems' overall recall values are highly affected by the gene name identification. REGULONDB is here taken as a "true" gold standard and thus as-

sumed to be correct and exhaustive with respect to the GREs contained. As, however, every manually curated database is likely to be incomplete and might contain some errors, we supplement our evaluation against REGULONDB with a manual analysis of false positives errors caused by our system (cf. Section 5.4).

5.1 Evaluation Scenario and Experimental Settings

To evaluate our extraction systems against REGULONDB we first processed a set of input documents (see below), collected all unique gene regulation events extracted and compared this set of events against the full set of known events in REGULONDB. A true positive (TP) hit is obtained, when an event found automatically corresponds to one in REGULONDB, i.e., having the same agent and patient. The type of regulation is not considered. A false positive (FP) hit is counted, if an event was found which does not occur in the same way in REGULONDB, i.e., either patient or agent (or both) are wrong. False negatives (FN) are those events covered by REGULONDB but not found by a system automatically. From these hit values, standard precision, recall, and F-score values are calculated. Of course, the systems' performance largely depend on the size of the base corpus collection processed. Thus, for both systems and all three document sets we got separate performance scores.

Table 2 gives an overview to the document collections used for evaluating the robustness of our systems: The "ecoli-tf" variants are documents filtered both with E. coli TF names and with relevance to E. coli. Abstracts are taken from Medline citations, while full texts are from a corpus of different biomedical journals. The third document set, "regulon ra", is a set containing abstracts from the REGULONDB references.

name	type	# documents
ecoli-tf.abstracts	abstract	4,347
ecoli-tf.fulltext	full texts	1,812
regulon ra	abstracts	2,704

Table 2: Document sets for REGULONDB evaluation

5.2 Rule-based-System

Table 3 shows the evaluation results of the rule-based system against REGULONDB. Though the system distinguishes the three types of events, we have considered them all as events of gene transcription regulation for the evaluation. For instance, the system has extracted 718 unique events with single-unit participants (i.e., excluding operons), not considering event types and attributes (e.g., polarity), from the "ecoli-tf.fulltext" corpus. Among the events, 347 events are found in Regulon (9.7% recall, 48.3% precision). If we only consider the events that are specifically identified as gene transcription regulation, the system has extracted 379 unique events among which 201 are also found in Regulon (5.6% recall, 53.0% precision).

participant	document set	R	P	F
single-unit	ecoli-tf.abstracts	0.09	0.60	0.15
multi-unit	ecoli-tf.abstracts	0.24	0.61	0.34
single-unit	ecoli-tf.fulltext	0.10	0.48	0.16
multi-unit	ecoli-tf.fulltext	0.25	0.49	0.33
single-unit	regulon ra	0.07	0.73	0.13
multi-unit	regulon ra	0.18	0.70	0.28

Table 3: Results of evaluation against REGULONDB of rule-based system.

When we split multi-unit participants into individual genes, the rule-based system shows better performance, as shown in Table 3 with the participant type "multi-unit". This may indicate that the gene regulatory events of E. coli are often described as interactions of operons. At best, the system shows 34% F-score with the "ecoli-tf.abstracts" corpus.

5.3 ML-based System

The ML-based system was designed to recognize all types of gene regulation events. REGULONDB, however, contains only the subtype, i.e., regulation of transcription. Thus, the ML-based system was evaluated against REGULONDB in two modes: by default, all events extracted by the systems are considered; in the "TF-filtered" mode, only relations with an agent from the list of all known TFs in E. coli are considered (as done for the rule-based system by default). Thus, comparing to the rule-based system, only the results obtained in the "TF-filtered" mode should be considered.

5.3.1 Raw performance scores

The results for the ML-based system are shown in Table 4. Recall values here range between 7 and 10%, while precision is between 29 and 78% depending on both the document set as well as the application of the TF filter. The low recall of the ML-based system is partially due to the fact that the system does not recognize multi-gene object names (e.g., "acrAB"), in this configuration the recall is similar to the recall of the rule-based system in a "single-unit modus" (see Table 3).

mode	document set	R	P	F
TF-filtered	ecoli-tf.abstracts	0.09	0.70	0.16
default	ecoli-tf.abstracts	0.09	0.45	0.15
TF-filtered	ecoli-relevant.fulltext	0.10	0.54	0.17
default	ecoli-relevant.fulltext	0.10	0.29	0.15
TF-filtered	regulon ra	0.07	0.78	0.13
default	regulon ra	0.07	0.47	0.12

Table 4: Results of evaluation against REGULONDB of ML-based system

As already shown in the intrinsic evaluation, application of different confidence thresholds increases the recall of the ML-based system. This was also done for the evaluation against REGULONDB. Table 5 shows the impact of increased confidence thresholds for the negative class on the "regulon ra" set for the "TF-filtered" evaluation mode. Given an extremely high threshold of 0.95, the recall is increased from 7 to 11% which constitutes a relative increase of over 60%. Precision obviously drops, however, the overall F-score has improved from 13 to 19%. These results emphasize that an ML-based system has an important handle which allows to adjust recall according to the application needs.

threshold	R	P	F
default (0.5)	0.07	0.78	0.13
0.8	0.09	0.70	0.16
0.95	0.11	0.63	0.19

Table 5: Different confidence thresholds for the ML-based system tested on the "regulon ra" set

5.4 Manual analysis of false positives

REGULONDB was taken as an absolute gold standard in this evaluation. If a system correctly extracts an event which is not contained in REGULONDB for some reason, this constitutes a FP. Moreover, all kinds of error (e.g., agent and patient mixed up) were subsumed as FP errors. To analyze the cause and distribution of FPs in more detail, a manual analysis of the FP errors was performed and original FP hits were assigned to one out of four FP error categories:

Cat1: Not a GRE This is really an FP error, as the extracted relation does not at all constitute a gene regulation event.

Cat2a: GRE but other than transcription Unlike REGULONDB which contains only one subtype of GREs, namely transcriptions, the ML-based system identifies all kinds of GREs. Therefore, the ML-based system clearly identifies events which cannot be contained in REGULONDB and, therefore, are not really FPs.

Cat 3: Partially correct transcription event This category deals with incorrect arguments of GREs. We distinguish three types of FPs: (a) the patient and the agent role are interchanged, (b) the patient is wrong, while the agent is right, and (c) the agent is wrong, while the patient is right. In all these three cases, though errors were committed human curators might find the partially incorrect information useful to speed up the curation process.

Cat4: Relation missing in REGULONDB Those are relations which should be contained in REGULONDB but are missing for some reason. The agent is a correctly identified transcription factor and the sentence contains a mention of a transcription event. There are several reasons why this relation was not found in REGULONDB as we will discuss in the following.

Table 6 shows the results of the manual FP analysis of the ML-based system (no TF filter applied) on the "ecoli-tf-abstracts" and "ecoli-tf-fulltexts". It shows that the largest source of error is due to `Cat1`, i.e., an identified relation is completely wrong. As fulltext documents are generally more complex, the relative amount of this kind of errors is higher here than on abstracts (54.5 % compared

category	abstracts (%)	fulltexts (%)
Cat 1	44.5	54.5
Cat 2	11.2	10.9
Cat 3a	3.8	3.9
Cat 3b	8.5	4.4
Cat 3c	8.2	5.4
Cat 4	23.8	21.0

Table 6: Manual analysis of false positive errors (FP). Percentages of FPs by category are reported on "ecoli-tf-abstracts" and "ecoli-tf-fulltexts"

to 44.5 %). However, on abstracts and fulltexts, a bit more than 10 % of the FP are because the system found too general GREs which, by definition, are not contained in REGULONDB (`Cat2`). Identified GREs that were partially correct constitute 20.5 % (abstracts) or 13.7 % (fulltexts) of the FP errors (`Cat3`).

Finally, 23.8% and 21.0% of the FPs for abstracts and fulltext, respectively, are correct transcription events but could not be found in REGULONDB (`Cat4`). This is due to several reasons. For instance, identified gene names were incorrectly normalized so that they could not be found in REGULONDB, REGULONDB curators have not yet added a relation or simply overlooked it; relations are correctly identified as such in the narrow context of a paragraph of a document but were actually of speculative nature only (this includes relations whose status is unsure, often indicated by "likely" or "possibly").

Summarizing, the manual FP analysis shows that about 50% of all FPs are not completely erroneous. These numbers must clearly be kept in mind when interpreting the raw numbers (especially for precision) reported on in the previous subsection.

5.5 Integration of text mining results

We have integrated the results of the two different text mining systems and found that both systems are complementary to each other such that their result sets do not heavily overlap. For instance, from the "ecoli-tf.abstract" corpus, the rule-based system extracts 992 events, while the ML-based system extracts 705 events. For the integration, we have considered only the events whose participants are associated with UNIPROT identifiers. Among the extracted events, only 285 events are extracted by both systems. We might speculate that the overlapping events are more reliable than the rest of the extracted events. It also leaves 71.3% of the results from the rule-based system and 59.6% of results from the ML-based system as unique contributions from each of the approaches for the integration.

6 Conclusions

We have explored a rule-based and a machine learning-based approach to the automatic extraction of gene regulation events. Both approaches were evaluated under well-defined lab conditions using self-supplied corpora, and under real-life conditions by comparing our results with REGULONDB, a well-curated reference data set. While the results for the first evaluation scenario are state of the art, performance figures in the real-life scenario are not so shiny (the best F-scores for the rule-based and the ML-based system are on the order of 34% and 19%, respectively). This holds, in particular, for the comparison with the work of Rodríguez-Penagos et al. (2007). Still, at least the ML-based approach is much more general than the very specifically tuned manual rule set from Rodríguez-Penagos et al. (2007) and has potential for increases in performance. Also, this has been the first extra-mural evaluation of automatically generating content for REGULONDB.

Still, the analysis of false positives reveals that the strict criteria we applied for our evaluation may appear in another light for human curators. Confounded agents and patients (21% on the abstracts, 14% on full texts) and information not contained in REGULONDB (24% on the abstracts, 21% on full texts) might be useful from a heuristic perspective to focus on interesting data during the curation process.

Acknowledgements

This work was funded by the EC within the BOOTStrep (FP6-028099) and the CALBC (FP7-231727) projects. We want to thank Tobias Wagner (Centre for Molecular Biomedicine, FSU Jena) for performing the manual FP analysis.

References

Ekaterina Buyko, Elena Beisswanger, and Udo Hahn. 2008. Testing different ACE-style feature sets for the extraction of gene regulation relations from MEDLINE abstracts. In *Proceedings of the 3rd International Symposium on Semantic Mining in Biomedicine (SMBM 2008)*, pages 21–28.

Francis Collins, Eric Green, Alan Guttmacher, and Mark Guyer. 2003. A vision for the future of genomics research. *Nature*, 422(6934 (24 Feb)):835–847.

Vasileios Hatzivassiloglou and Wubin Weng. 2002. Learning anchor verbs for biological interaction patterns from published text articles. *International Journal of Medical Informatics*, 67:19–32.

Minlie Huang, Xiaoyan Zhu, Donald G. Payan, Kunbin Qu, and Ming Li. 2004. Discovering patterns to extract protein-protein interactions from full texts. *Bioinformatics*, 20(18):3604–3612.

Sophia Katrenko and P. Adriaans. 2006. Learning relations from biomedical corpora using dependency trees. In *KDECB 2006 – Knowledge Discovery and Emergent Complexity in Bioinformatics*, pages 61–80.

Jung-jae Kim and Jong C. Park. 2004. BioIE: retargetable information extraction and ontological annotation of biological interactions from the literature. *Journal of Bioinformatics and Computational Biology*, 2(3):551–568.

Seon-Ho Kim, Juntae Yoon, and Jihoon Yang. 2008. Kernel approaches for genic interaction extraction. *Bioinformatics*, 24(1):118–126.

Claire Nédellec. 2005. Learning language in logic - genic interaction extraction challenge. In *Learning language in logic - genic interaction extraction LLL' 2005*, pages 31–37.

Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Björne, Filip Ginter, and Tapio Salakoski. 2008. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, 9(3), April.

Carlos Rodríguez-Penagos, Heladia Salgado, Irma Martínez-Flores, and Julio Collado-Vides. 2007. Automatic reconstruction of a bacterial regulatory network using natural language processing. *BMC Bioinformatics*, 8(293).

Kenji Sagae, Yusuke Miyao, and Junichi Tsujii. 2007. HPSG parsing with shallow dependency constraints. In *Annual Meeting of Association for Computational Linguistics*, pages 624–631.

Jasmin Šarić, Lars J. Jensen, Rossitza Ouzounova, Isabel Rojas, and Peer Bork. 2004. Extracting regulatory gene expression networks from pubmed. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 191, Morristown, NJ, USA. Association for Computational Linguistics.

Joachim Wermter, Katrin Tomanek, and Udo Hahn. 2009. High-performance gene name normalization with GeNo. *Bioinformatics*, 25(6):815–821.

Hui Yang, Goran Nenadic, and John Keane. 2008. Identification of transcription factor contexts in literature using machine learning approaches. *BMC Bioinformatics*, 9(Supplement 3: S11).

Extraction of Named Entities from Tables in Gene Mutation Literature

Wern Wong, **David Martinez****, **Lawrence Cavedon*****

**NICTA Victoria Research Laboratory
*Dept of Computer Science and Software Engineering
The University of Melbourne

{*wongwl,davidm,lcavedon*}*@csse.unimelb.edu.au*

Abstract *We investigate the challenge of extracting information about genetic mutations from tables, an important source of information in scientific papers. We use various machine learning algorithms and feature sets, and evaluate performance in extracting fields associated with an existing hand-created database of mutations. We then show how classifying tabular information can be leveraged for the task of named entity detection for mutations.*[1]

Keywords Information extraction; tables; biomedical applications.

1 Introduction

We are interested in applying *information extraction* and *text mining* techniques to aiding the construction of databases of biomedical information, in particular information about genetic mutations. Such databases are currently constructed by hand: a long, involved, time-consuming and human-intensive process. Each paper considered for inclusion in the database must be read, the interesting data identified and then entered by hand into a database.[2]

However, the biomedical domain throws up many new and serious challenges to information extraction and text mining. Unusual terminology and underdeveloped standards for nomenclature present problems for tokenisation and add complexity to standard information extraction tasks, such as *named entity recognition (NER)*. A lack of resources (at least compared to other domains), such as collections of annotated full-text documents and relevance judgements for various tasks, are a bottleneck to developing and evaluating the core techniques required.

In this paper, we report on work performed on extracting information from *tables* in biomedical research papers. Tables present a succinct and information-rich format for providing information, and are particularly important when reporting results in biological and medical research papers. For example, the Human Genome Variation Society (HGVS), in its general recommendations for mutation nomenclature, recommends making use of tabular listings when several changes are described in a manuscript.[3] A specific premise of our work is that the highly-structured nature of tabular information allows leverage of some techniques that are not so sensitive to the well-reported problems inherent in biomedical terminology, which complicate NER tasks in this domain. In particular, we describe initial techniques for extending NER performance through the analysis of tables: columns/rows are classified as containing items of the entities of interest, thereby allowing those entities to be recognized as of the target type. Since a significant amount of such entities may be found in tables in biomedical scientific papers, this can have positive impact on the performance of base NER techniques.

NER tools specifically targeted at recognising mutations have been developed (e.g. (Horn et al., 2004; Baker and Witte, 2006; Caporaso et al., 2007; Lee et al., 2007)); however, they only detect a subclass of mutations, so-called *single-point mutations*,

[1] A short version of this paper was presented at the *Australasian Document Computing Symposium*, 2008. All copyrights from that event were retained by the authors.

[2] Karamis *et al* (2008) illustrate how even simple tools can have an impact on improving the database-curation process.

[3] http://www.hgvs.org/mutnomen/recs.html#general

Proceedings of the Workshop on BioNLP, pages 46–54,
Boulder, Colorado, June 2009.

i.e. those that affect a single base. MutationFinder (Caporaso et al., 2007) is the only publicly available tool, built with around 700 automatically-generated rules (both for different nomenclatures and natural language). However, most of the mutations that we find in our dataset are not point mutations or do not follow point-mutation nomenclature, limiting the usefulness of MutationFinder (and related tools) over our document collection.

In the next section, we describe the setting of our task, the *Mismatch Repair (MMR) Database*, and outline the task of extraction from tables. In Section 3, we describe the preparation of our document collection, and in Section 4, we analyse the amount of mutation-related information that is in the associated tables. Section 5 describes the main task, which is classifying table rows and columns as containing mutations, and Section 6 leverages this technique to detect mutations of interest to the MMR Database. We discuss the results in Section 7.

2 Background

In this section, we discuss the MMR database—the setting for our task and from which we construct our document collection—and previous approaches to table processing.

2.1 The MMR Database

Our extraction task is grounded in the specific context of the *Mismatch Repair (MMR) Database* compiled at the Memorial University of Newfoundland (Woods et al., 2007)—a database of known genetic mutations related to hereditary non-polyposis colorectal cancer (HNPCC), a hereditary form of bowel cancer. The MMR Database contains information on genetic mutations known to be related to HNPCC, along with links to the research papers from which the database has been constructed.[4] From the database and its links to papers, we were able to construct a collection of tables related to HNPCC mutations, and then use the MMR database records themselves as a gold standard for evaluating our techniques. As of May 2008, the MMR database contained a total of 5,491 records on mutations that occur on any one of four genes that have been identified as related to colon cancer. An example record from the MMR database is the following:

MLH1	Exon13	c.1491delG	Yamamoto et al.	9500462

Respectively, this record contains: gene; exon; mutation; citation of the paper the information was sourced from;[5] and the paper's PubMedID. These fields are important because they contain information researchers are directly interested in (gene, exon, mutation) and the paper said information was found in. Note that if a gene/mutation pair is referenced in multiple papers, then there are correspondingly multiple entries in the database. Conversely, if a single paper mentions multiple (relevant) genes, then that paper is mentioned in multiple database records.

2.2 Table Processing

An important but less-researched sub-problem in text mining is information extraction from tables. This is particularly important in the biomedical domain since much important data is present in tabular form, such as experimental results, relations between entities, and other information that may not be contained elsewhere in the text. For example, the table shown in Figure 1 (taken from an article in our collection) contains much of the same data that was present in database records, in a similar format.

Tabular information extraction can be divided into two broad sub-tasks:

- table detection: identifying tables within documents;
- table processing: extraction of data from tables.

Several systems have been developed to handle both tasks, some are designed only to handle table detection, and others focus only on extracting data. Both machine learning and heuristic / rule-based approaches have been proposed.

Table detection techniques depend heavily on the input format. Most work that tackles this problem tends to assume one homogeneous input format, but tables generally come in one of two varieties:[6]

[4] I.e. a team of geneticists manually trawled the biomedical literature for information on HNPCC-related mutation information, and added links to any papers relevant to those mutations in the context of HNPCC.

[5] This field has been abbreviated. We have also omitted fields such as "internal id".

[6] We don't consider the possibility of processing bitmaps or other images from scanned documents.

Category	Patient No	Family No	Gene	Exon	Base change	Base No	Codon	AA change	MSI	Comment
1	1	1	MLH1	11	Del G	1046	349	Frameshift	MSI-H	Stop codon 380
Amsterdam	2	1	MLH1	11	Del G	1046	349	Frameshift	MSI-H	Stop codon 380
criteria I	3	2	MLH1	16	Del AAG	1846–8	616	Del Lys	MSI-H	Pathogenic, see text
	4	3	MSH2	12	Del AAT	1786–8	596	Del Asn	MSI-H	Pathogenic, see text
	5	4	MSH2	11	T>A	2 bases downstream	—	Splice mutation	MSI-H	Pathogenic, see text
2	12	11	MLH1	1	C>T	76	26	Gln>stop	MSI-H	Stop codon
Amsterdam	13	12	MSH2	12	Del AAT	1786–8	596	Del Asn	MSI-H	Pathogenic, see text
criteria II	16	4	MSH2	11	T>A	2 bases downstream	—	Splice mutation	MSI-H	Pathogenic, see text
6*	37	34	MSH2	12	Del AAT	1786–8	596	Del Asn	MSI-H	Pathogenic, see text
7*	42	39	MSH2	12	Del AAT	1786–8	596	Del Asn	MSI-H	Pathogenic, see text

*Category 6, proband at age 40 years or younger with at least one CRC among family members; category 7, both proband and one first degree relative with CRC <55 years.

MSI, microsatellite instability; MSI-H, high microsatellite instability; MSI-L, low microsatellite instability; MS-S, stable microsatellites; CRC, colorectal cancer.

Figure 1: Sample table containing mutation information related to HNPCC

- raw text tables: generally ASCII text in monospace font, delimited by whitespace and/or special characters;
- rich text tables: those formatted using LaTeX, PDF, HTML and other such formats.

Tables in plain text tend to be more difficult to detect, as the detection system must be sensitive to whitespace and symbols used to align cells in tables. Efforts to handle rich text formats generally focus on HTML-based representations. Raw HTML is easier to parse than raw LaTeX or PDF, and most formats are easily converted to HTML. HTML tables can theoretically be trivially detected using `<table>` tags. However, Lerman *et al* (2004) note that in HTML files taken from the web, only a fraction of tabular data was presented using `<table>` tags, and those tags were also used to format multi-column text, images and other non-table applications. Hurst (2001) attests that less than 30% of HTML tables on the web contain actual tabular content; for many, the HTML table tags are often used simply for formatting purposes.

Zanibbi *et al* (2004) present a survey of table recognition in general. Of greatest relevance to us here are approaches that adopt a machine learning approach to detecting and/or extracting table data.

Cohen *et al* (2002) use features based on HTML table tags, number of rows and columns of specific lengths, and ratios of single-square cells to total number of cells, to perform table detection, and then form a geometric representation of the data using algorithms based on table-rendering techniques implemented by browsers.

Pinto, Wei, and their colleagues have used *conditional random fields (CRFs)* to both detect and process tables simultaneously. Pinto *et al* (2003) compare the output of their CRF system with a previous effort using hidden Markov machines (HMMs). These systems use features such as: presence of whitespace of varying length (different lengths of whitespace are used as separate features); domain-specific lexical features (such as month names, year strings, specified keywords); separator characters (e.g. '+', '-', etc). In subsequent work they develop a system for performing question answering over table data (Wei et al., 2004) by treating each extracted data cell as a discrete document.

To our knowledge, no previous system has attempted to extract data from tables in biomedical literature. This is possibly because of a combination of the lack of resources for this domain (e.g.

collections of full-text documents; relevance judgements), as well as the lesser focus on text mining in general in this area. As will be seen in the next section, the vagaries of the construction of our collection of tables means we were effectively able to ignore the issue of table detection and focus directly on the problem of processing.

3 Experimental Setting

Our experiments were designed to identify mentions of mutations in the biomedical literature, focusing on tabular content. In this section, we first describe our target dataset, built from the hand-curated MMR database (Woods et al., 2007); we then explain the table extraction process; finally, we introduce the task design.

3.1 Mutation Mention Dataset

We relied on the MMR Database and MEDLINE in order to build our test collection. First we collected all the information available in the hand-curated MMR records, obtaining a total of 5,491 mutations linked to 719 distinct PubMedIDs[7].

Our next step was to crawl the full-text articles from MEDLINE. We used an automatic crawler that followed the links from the PubMed interface, and downloaded those papers that had a full-text HTML version, and which contained at least one content table.

The tables were then extracted from the full text HTML files. It is important to note that the tables were already present as links to separate HTML files rather than being presented as inline tables, making this process easier. Papers that did not contain tables in HTML format were discarded.

Our final collection consisted of 70 papers out of the original 719 PubMedIDs. Some of the papers were not available in full text, and for others our crawling script failed to extract the full version. Our approach was conservative, and our collection could be augmented in the future, but we decided to focus on this dataset for the experiments presented in this paper. This set of articles is linked to 717 MMR records (mutations), which constitutes our gold standard hand-curated annotation. The collection contains 197 tables in all.

[7]Data was downloaded from the web interface in May 2008.

3.2 Table extraction

Once scraped, the tables were then pre-processed into a form that more readily allowed experimentation. The tables were therefore split into three parts: column headers, row headers, and data cells. This was done based on the HTML formatting, which was consistent throughout the data set as the tables were automatically generated.

The first step was to deconstruct the HTML tables into nested lists of cells based on HTML table tags. Inconsistencies introduced by `colspan` and `rowspan` attributes were resolved by replicating a cell's contents across its spanned lengths. That is, a cell with `colspan=3` would be duplicated across the three columns, and likewise for cells spanning multiple rows. Single-cell rows at the top or bottom of a table were assumed to be captions and discarded.

The remaining HTML was stripped, save for the following tags which contained important information:

- `img` tags were replaced by their *alternate* text, where available. Such images often represent a mathematical symbol, which is important information to retain;
- `hr` tags proved to be an important indicator for dividing header cells from data cells.

Tables were broken up into row headers, column headers, and data cells by making use of the `hr` tags, denoting horizontal lines, to detect column headers. Such tags tend to be present as a separator between column header cells and data cells; in fact, the only tables in our collection that did not have the separators did not have column headers either. The `hr` tags were subsequently stripped after this use. Detecting row headers was performed by checking if the top left cell of the table was blank, a pattern which occurred in all row-major tables. The vast majority of tables had column headers rather than row headers, although some had both and a small proportion had only row headers. We acknowledge that this processing may be specific to the vagaries of the specific format of the HTML generation used by PubMed (from which we sourced the tables). However, our whole task is specific to this domain; further, our focus is on the extraction task rather than the actual detection of row/column headers.

Class	Class Freq.	Cell Freq.
Gene	64	1,618
Exon	48	1,004
Codon	23	435
Mutation	90	2,174
Statistic	482	8,788
Other	576	14,324
Total	1,283	28,343

Table 1: Frequency per class and number of cells in the collection.

3.3 Task Design

In order to extract mutation information from tables, we first performed classification of full columns/rows into relevant entities. The content of a column (or row, depending on whether the table was row- or column-oriented) tends to be homogeneous; this allowed us to build classifiers that can identify full vectors of relevant entities in a single step. We refer to this task as *table vector classification*.

We identified the following classes as relevant: Gene, Exon, Mutation, Codon, and Statistic. The first four were chosen directly from the MMR Database. We decided to include "Statistic" after inspecting the tabular dataset, since we found that this provides relevant information about the importance of a given mutation. Of the five classes, Mutation is the most informative for our final information extraction goal.

The next step was to hand-annotate the headers of the 197 tables in our collection by using the five classes and the class "Other" as the tagset. Some headers belonged to more that one class, since the classes were collapsed into a single field of the table. The frequency per class and the number of cells, across the collection of tables, is shown in Table 1.

3.4 Evaluation

We evaluated our systems in two ways:

- Header classification: performance of different systems on predicting the classes of each column/row of the tables;
- Mutation extraction: recall of our system over the subset of the hand-curated MMR database.

Evaluation for the header classification step was performed using precision, recall and f-score, micro-averaged amongst the classes. Micro-averaging involves multiplying the score of a class by the number of instances of the class in the gold standard, and dividing by the total number of instances. For the machine learning algorithms, evaluation was performed using 10-fold cross-validation. For mutation extraction we focus on a single class, and produce recall and a lower-bound on precision.

4 Mutation Mentions in Tables

In order to determine the value of processing tabular data for mutation-mining purposes, we obtained a sample of 100 documents that were hand-annotated by curators prior to their introduction in the database—the curators highlighted relevant mutations found in each paper. We found that for 59 of the documents, only the tabular parts of the paper were selected; 33 of the documents had only textual parts highlighted; and for 8 documents both tables and text were selected. This is an indicator of the importance of tabular data in this context.

Our next step was to measure the amount of information that we could potentially extract from the tables in our collection. Since we are interested in mutations, we extracted all cells from the vectors that were manually annotated as "Mutation" in order to compare them to the goldstandard, and measure the recall. This comparison was not straightforward, because mutation mentions have different nomenclatures. Ideally we would normalise the different references into a standard form, and then perform the comparison. However, normalisation is a complex process in itself, and we resorted to evaluation by hand at this point.

We found that 198 of the 717 goldstandard mutations were present in tables (28%). This is a significant amount, taking into account that their extraction should be much easier than parsing the raw text. We also tested MutationFinder over the full text, and found that only 6 of the goldstandard mutations were retrieved (0.8%), which indicates that point mutation identification is not sufficient for this task.

Finally, we measured the amount of information that could be extracted by a simple string look-up system separately over the tabular and textual parts

of the articles. We were looking for mutation mentions that correspond exactly to the goldstandard record from each article, which meant that mentions in different nomenclatures would be missed. We found that a total of 177 mentions (24.7%) could be found with the same spelling; of those 142 (80.1%) were found in tables only, and the remaining 35 (20.9%) were found in both tables and text; i.e., no mention was found in text only.

These results indicate that we can find relevant information in tables that is not easy to detect in running text.

5 Table Vector Classification

We built automatic classifiers to detect relevant entities in tables. Two separate approaches were attempted for vector classification: applying heuristic rules, and machine learning (ML) techniques. These are described here, along with an analysis of their performance.

5.1 Heuristic Baseline

As a baseline method, we approached the task of classifying headers by matching the header string to the names of the classes in a case-insensitive manner. When the class name was found as a substring of the header, the class would be assigned to it. For example, a header string such as "Target Mutation" would be assigned the class "Mutation". Some headers had multiple annotations (E.g. "Gene/Exon").

For better recall, we also matched synonyms for the class "Mutation" (the terms "Variation" and "Missense") and the class "Statistic" (the terms "No.", "Number" and "%"). For the remaining classes we did not identify other obvious synonyms.

The results are shown in Table 2. Precision was reasonably high for the "Codon", "Exon" and "Statistic" classes. However, this was not the case for "Mutation", and this illustrates that different types of information are provided under this heading; illustrative examples include the heading "Mutation Detected" on a "Gene" vector, or the heading "Germline Mutation" referring to "Statistics". The recall was also low for "Mutation" and most other classes, showing that more sophisticated approaches are required in order to exploit the information contained in the tables. Notice also that the micro-average is highly biased by the classes "Statistic" and "Others", since they contain most of the test instances.

Class	Precision	Recall	FScore
Gene	0.537	0.620	0.575
Exon	0.762	0.615	0.681
Codon	0.850	0.654	0.739
Mutation	0.283	0.301	0.292
Statistic	0.911	0.324	0.478
Other	0.581	0.903	0.707
Micro Avg.	**0.693**	**0.614**	**0.651**

Table 2: Naive Baseline results across the different classes and micro-averaged

Class	Precision	Recall	FScore
Gene	0.537	0.611	0.571
Exon	0.762	0.615	0.681
Codon	0.850	0.654	0.739
Mutation	0.600	0.452	0.515
Statistic	0.911	0.340	0.495
Other	0.579	0.910	0.708
Micro Avg.	**0.715**	**0.633**	**0.672**

Table 3: Results integrating MutationFinder across the different classes and micro-averaged

Our second step was to build a more informed classifier for the class "Mutation" using the point mutation NER system MutationFinder (Caporaso et al., 2007). We applied this tool to the text in the table-cells, and identified which table-vectors contained at least one mutation mention. These vectors were also classified as mutations. The results are shown in Table 3. This approach caused the "Mutation" results to improve, but the overall f-score values are still in the range 50%-70%.

We considered other heuristic rules that could be applied, such as looking for different kinds of patterns for each class: for instance, numbers for "Exon", or the normalised form `c.N[symbol]N` for mutation, or trying to match against term lists (e.g. using Gene dictionaries). Future work will explore extending the ML approach below with features such as these.

5.2 Classification Techniques

For the ML experiments we used the Weka (Witten and Frank, 2005) toolkit, as it contains a wide selection of in-built algorithms. We selected a variety of well-known approaches in order to obtain a better picture of the overall performance. As a baseline, we applied the majority class from the training data to all test instances. We applied the following ML systems:[8] Naive Bayes (NB); Support Vector Machines (SVM); Propositional Rule Learner (JRip); and Decision Trees (J48). We did not tune the parameters, and relied on the default settings.

In order to define our feature sets, we used the text in the headers and cells of the tables, without tokenisation. Other possible sources of information, such as captions or the running text referring to the table were not employed at this stage. We applied four feature sets:

- **Basic (Basic):** Four basic features, consisting of the header string, the average and median cell lengths, and a binary feature indicating whether the data in the cells was numeric.
- **Cell Bag-of-Words (C_bow):** Bag of words over the tokens in the table cells.
- **Header Bag-of-Words (H_bow):** Bag of words over the tokens in the header strings.
- **Header + Cell Bag-of-Words (HC_bow):** Combination of bags of words formed by the tokens in headers and cells, represented as separate types of features.

The micro-averaged results of the different learning methods and feature sets are shown in Table 4. Regarding the feature sets, we can see that the best performance is obtained by using the headers as bag-of-words, while the content of the cells seems to be too sparse to guide the learning methods. SVM is the best algorithm for this dataset, with JRip and J48 following, and NB performing worst of the four in most cases.

Overall, the results show that the ML approach is superior to the baselines when using the header bag of words feature to classify the relevant entities.

[8] We applied a number of other ML algorithms as well, but these showed significantly lesser performance.

Method	Feature Sets			
	Basic	**C_bow**	**H_bow**	**HC_bow**
Mj. Cl.	0.288			
NB	0.614	0.454	0.678	0.581
SVM	**0.717**	**0.599**	**0.839**	**0.816**
JRip	0.564	0.493	0.790	0.749
J48	0.288	0.532	0.793	0.782

Table 4: Results for ML Algorithms - Micro-Averaged FScores. Mj.Cl.: Majority Class. The best results per column are given in bold.

Class	Precision	Recall	FScore
Gene	0.778	0.737	0.757
Exon	0.786	0.707	0.745
Codon	0.833	0.882	0.857
Mutation	0.656	0.679	0.667
Statistic	0.919	0.853	0.885
Other	0.82	0.884	0.850
Micro Avg	**0.839**	**0.841**	**0.839**

Table 5: Results for SVM and the feature set *H_bow* per class and micro-averaged.

SVM is able to reach a high f-score of 83.9%, which has been found to be significantly better than the best baseline after applying a paired t-test (p-value under 0.0001).

We break down the results per class in Table 5, using the outputs from SVM and feature-set *H_bow*. We can see that all classes show an improvement over the heuristic baselines. There is a big increase for the classes "Gene" and "Statistic", and all classes except mutation are above 70% f-score. "Mutation" is the most difficult class to predict, but it still reaches 66.7% f-score, which can be helpful for some tasks, as we explore in the next section.

6 Automatic Mutation Extraction

We applied the results of our classifier to a practical application, i.e., the detection of mutations in the literature for the MMR Database project. Table vector classification allows us to extract lists of candidate mutation names from tables to be added to the database. We would like a system with high recall that identifies all relevant candidates, but also acceptable precision so that not all the tables need to

System	Mut. Found	TP	% in MMR	Rec.
Automatic	1,702	153	9.0	77.3
Gold standard	1,847	198	10.7	100

Table 6: Results for Mutation detection. TP indicates the number of true positives, "% in MMR" shows the percentage of positives found in the database.

be hand-checked.

In order to test the viability of this approach, we measured the results of the system in detecting the existing hand-curated mutations in MMR. We calculated the recall in retrieving those mutations, and also the rate of false positives; however, note that we also consider as false positives those valid mutations that were not relevant for MMR, and therefore the reported precision is artificially low.

Results for the automatic extraction and the gold-standard annotation are given in Table 6. As expected, there is a high rate of false positives in the goldstandard and automatic systems; this shows that most of the mutations detected are not relevant for the MMR database. More interestingly, we were able to retrieve 77.3% of relevant mutation mentions automatically using the ML approach, which corresponds to 21.3% of all the hand-curated data.

The vector classifier discriminates 1,702 mutation cells out of a total of 27,700 unique cells in the table collection, and it effectively identifies 153 out of the 198 relevant mutations present in the tabular data. This means that we only need to hand-check 6.1% of the tabular content to retrieve 77.3% of relevant mutations, saving the curators a significant amount of time. The classifiers could also be biased towards higher recall by parameter tuning—this is an area for further investigation.

Finally, after the evaluation process we observed that many false mutation candidates could be removed by discarding those that do not contain two consecutive digits or any of the following n-grams: "c.", "p.", '>', "del", "ins", "dup". This heuristic reduces the number of mutation candidates from 1,702 to 989 with no cost in recall.

7 Discussion

While this is early work, our preliminary results on the task of identifying relevant entities from gene mutation literature show that targeting tables can be a fruitful approach for text mining. By relying on ML methods and simple bag-of-words features, we were able to achieve good performance over a number of selected entities, well above header word-matching baselines. This allowed us to identify lists of mentions of relevant entities with minimal effort. An advantage of our approach is that the annotation of examples for training and evaluation is considerably easier, since many entities can be annotated in a single step, opening the way to faster annotation of other entities of interest in the biomedical domain.

The approach of using table vector classification for the named entity task also has promise. In particular, the wide variety and non-standard terminology of biomedical entities (i.e. genes, proteins, mutations) is one of the challenges to NER in this domain. However, since a column of homogeneous information may include representatives of the heterogeneous nomenclature schemes, classification of a whole column or row potentially helps nullify the effect of the terminological variability.

For future work, we plan to study different types of features for better representing the entities targeted in this work. Specially for mutation mentions, we observed that the presence of certain ngrams (e.g. "del") can be a strong indicator for this class. Another issue we plan to address is that of the normalisation of mutation mentions into a standard form, for which we have started developing a collection of regular expressions. Another of our goals is to increase the size of our dataset of articles by improving our web crawler, and by hand-annotating the retrieved table vectors for further experimentation. Finally, we also aim to explore the potential of using tabular data for NER of different entities in the biomedical domain, such as gene mentions.

Acknowledgements NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.
Thanks to Mike Woods and his colleagues at the Memorial University of Newfoundland for making the MMR database and their curation data available to us. Eric Huang wrote several of the scripts mentioned in Section 3 for creating the table collection.

References

C. J. O. Baker and R. Witte. 2006. Mutation mining–a prospector's tale. *J. of Information Systems Frontiers*, 8(1):45–57.

J. G. Caporaso, W. A. Baumgartner Jr., D. A. Randolph, K. B. Cohen, and L. Hunter. 2007. Mutationfinder: A high-performance system for extracting point mutation mentions from text. *Bioinformatics*, 23(14):1862–1865.

W. W. Cohen, M. Hurst, and L. S. Jensen. 2002. A flexible learning system for wrapping tables and lists in html documents. In *WWW '02: Proc. 11th Int'l Conf. on World Wide Web*, pages 232–241, Honolulu.

F. Horn, A. L. Lau, and F. E. Cohen. 2004. Automated extraction of mutation data from the literature: Application of MuteXt to g protein-coupled receptors and nuclear hormone receptors. *Bioinformatics*, 20(4):557–568.

M. Hurst. 2001. Layout and language: Challenges for table understanding on the web. Technical report, WhizBang!Labs.

N. Karamanis, R. Seal, I. Lewin, P. McQuilton, A. Vlachos, C. Gasperin, R. Drysdale, and T. Briscoe. 2008. Natural language processing in aid of flybase curators. *BMC Bioinformatics*, 9:193–204.

Lawrence C. Lee, Florence Horn, and Fred E. Cohen. 2007. Automatic extraction of protein point mutations using a graph bigram association. *PLoS Computational Biology*, 3(2):e16+, February.

K. Lerman, L. Getoor, S. Minton, and C. Knoblock. 2004. Using the structure of web sites for automatic segmentation of tables. In *SIGMOD'04*, pages 119–130, Paris.

D. Pinto, A. McCallum, X. Wei, and W. B. Croft. 2003. Table extraction using conditional random fields. In *SIGIR '03*, pages 235–242.

X. Wei, W.B. Croft, and D. Pinto. 2004. Question answering performance on table data. *Proceedings of National Conference on Digital Government Research*.

I. H. Witten and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition.

M.O. Woods, P. Williams, A. Careen, L. Edwards, S. Bartlett, J. McLaughlin, and H. B. Younghusband. 2007. A new variant database for mismatch repair genes associated with lynch syndrome. *Hum. Mut.*, 28:669–673.

R. Zanibbi, D. Bolstein, and J. R. Cordy. 2004. A survey of table recognition. *Int'l J. on Document Analysis and Recognition*, 7(1).

Selecting an Ontology for Biomedical Text Mining

He Tan, Patrick Lambrix
Department of Computer and Information Science
Linköpings universitet, Sweden

Abstract

Text mining for biomedicine requires a significant amount of domain knowledge. Much of this information is contained in biomedical ontologies. Developers of text mining applications often look for appropriate ontologies that can be integrated into their systems, rather than develop new ontologies from scratch. However, there is often a lack of documentation of the qualities of the ontologies. A number of methodologies for evaluating ontologies have been developed, but it is difficult for users by using these methods to select an ontology. In this paper, we propose a framework for selecting the most appropriate ontology for a particular text mining application. The framework comprises three components, each of which considers different aspects of requirements of text mining applications on ontologies. We also present an experiment based on the framework choosing an ontology for a gene normalization system.

1 Introduction

With the explosive growth of the volume of published biomedical research, it is challenging to keep up to date with the underlying knowledge available in the form of free text. The necessity of understanding actions of individual biological components in system context rather than in isolation, extends the coverage of literature far beyond the capabilities of individual scientists. Text mining is an emerging field that attempts to deal with these challenges (Ananiadou and McHought, 2006; Cohen and Hersh, 2005; Spasic et al., 2005). Text mining requires a significant amount of domain knowledge. A large number of biomedical ontologies already exist, and hold much of the information. Some of the ontologies have been designed for modeling domain knowledge, e.g. FMA (Rosse and Mejino, 2003) and GO (Ashburner et al., 2000), others are developed for potential applications, e.g. MeSH (Lowe and Barnett, 1994) for indexing the medical literature. Whatever purposes the ontologies were built for, they are used to support text mining for tasks such as access to text, natural language processing, and information integration. Developers of text mining applications often look for appropriate ontologies that can be integrated into their systems, rather than develop new ontologies from scratch. The choice of ontology may, however, have a major impact on the performance of the text mining system, including the quality of the results.

Selecting an appropriate ontology relies on evaluation and comparison of the available ontologies. Unfortunately, there is often a lack of documentation of the qualities of the ontologies. A number of methodologies for evaluating ontologies have been proposed, mainly for evaluating one ontology at a time. However, it is difficult for users by using these methods to make a decision on ontology selection. The various evaluation methods can be classified into three main categories: 1) Those that evaluate ontologies against a set of criteria defined by human (e.g. (Lozano and Gómez, 2004) suggests 160 criteria). Most of the evaluation criteria are from the point of view of ontology building. They are not applicable for the selection of ontologies for a particular application. 2) Those that include methods to gather statistics about the nature of ontologies (e.g. (Gangemi et al., 2006) proposes 32 measures for depth, width, fan-out, etc). The problem

Proceedings of the Workshop on BioNLP, pages 55–62,
Boulder, Colorado, June 2009.

for this kind of methods is that it is not clear how to use these statistics for ontology selection among alternatives. 3) Those that are application-dependent evaluations. For example, ontologies are evaluated against a corpus in (Brewster et al., 2004) regarding the domain of an IE system, and (Porzel and Malaka, 2004) evaluates the quality of an ontology regarding a relation tagging task by using gold standard data tagged by human. While evaluating ontologies for the particular application is a relatively straightforward method, evaluations may be sensitive to the test dataset, and it may also be expensive to perform evaluations for many ontologies.

In this paper we propose a framework for selecting the most appropriate ontology for a particular text mining application. The framework comprises three components for selection, and uses and extends some of existing ontology evaluation and validation methods regarding the requirements of text mining in the area. After a brief overview of the related work, we introduce our framework in section 3. In section 4 we show how to use the framework in the setting of selecting an ontology for the gene normalization system described in (Tan, 2008).

2 Related Work

Not much work has been done yet for ontology selection in the biomedical area. Bioportal (Bioportal, 2.0) is an ontology library, currently mainly for browsing and navigating biomedical ontologies. The system also allows users of ontologies to submit information about their applications and comments on the content of ontologies, and stores mappings between concepts of ontologies in the library. This information may help users to select ontologies from the repository.

Some work has been done for ontology selection for the Semantic Web. In (Sabou et al., 2006), authors indicate the challenges for ontology evaluation posed by ontology selection for the Semantic Web, such as the need for automation and good performance. Two examples of ontology selection for the Semantic Web are OntoSelect (Buitelaar et al., 2004) and AKTiveRank (Alani and Brewster, 2005). Both are mainly based on the second category of ontology evaluation methods. OntoSelect is an ontology library that gives a functionality for selecting ontologies for a given knowledge markup task. The selection relies on measuring the number of concepts and properties, and popularity of ontologies. The ontology selection algorithm in AKTiveRank combines the measures of coverage of an ontology given search terms, and locations of the terms and semantic similarity between the terms in the structure of the ontology.

OntoMetric is a hierarchical framework proposed in (Lozano and Gómez, 2004) for general ontology selection. The tool offers a default hierarchy of criteria to evaluate ontologies from the point of view of building ontologies, and also allows users to adapt it for their needs. The selection is based on a weighted addition of value of each criteria.

3 The Framework

In principle biomedical ontologies provide formal representations of domain knowledge for text mining in the area, but they are used for different purposes such as providing a model for storing, searching and querying a repository of text; providing domain knowledge for natural language processing; providing a framework for information integration; or several of the above purposes.

Based on a literature study and experience in building systems, in this section we suggest criteria for selecting an ontology for a given biomedical text mining application. The criteria are organized in three components of a framework.

Component 1

In the first component the criteria for ontology selection are given in two dimensions: content and supporting technologies. The combined requirements from the two dimensions lead to a list of candidate ontologies.

Content What are the requirements on the content of ontology given an application? We propose three characteristics to be considered,

- *Type* What is the kind of the ontology required in the scenario? Ontologies can be distinguished into three basic types (Stenzhorn et al., 2005): *top ontologies*, e.g. BFO (Smith, 2004), contain only a restricted set of highly general classes, such as Function and Object, which are not tied to any particular domain of interest;

top-domain ontologies, e.g. BioTop (Stenzhorn et al., 2005), contain all classes that are essentially needed to describe a certain domain, such as Organism, Cell and Tissue in the case of biology; and *domain ontologies*, e.g. GO (Ashburner et al., 2000), contain classes that comprehensively describe a certain domain of interest. For example, for natural language processing tasks such as entity recognition, a top-domain ontology may be sufficient.

- *Scope* What are the main subjects that need to be covered in a top-domain or domain ontology required by the application? For example, gene and disease are the domains that concern (Hristovski et al., 2005).

- *Representation* What kind of information needs to be present in the ontology? From a knowledge representation point of view, ontologies can have the following components: *concepts*, which represent sets or classes of entities in a domain; *relations* between concepts, *instances*, which represent the actual entities; and *axioms*, which represent facts that are always true in the topic area of the ontology (Lambrix et al., 2007). Which components should be present in the ontology, depends to some extent on the purpose of the ontology in the application. For example, if an ontology is used as resource for NLP in the application, components *concepts* and *instances* both may be necessary, but not *relations* and *axioms*.

Supporting technologies Three questions are considered in this dimension:

1. What technologies are needed to support the use of the ontology in the scenario? Supporting technologies may include ontology representation languages, ontology browsers, tools for ontology alignment, reasoning services, and ontology-driven NLP tools.

2. Are tools available to provide the supporting technologies for the ontology? For example, ontology alignment systems, e.g. (Lambrix and Tan, 2008) are available for biomedical ontologies in OWL and OBO format.

3. What is the cost to develop new tools to support the use of an ontology if there is no existing tools? Does the cost meet the expectations of the application?

Component 2

In this level the criteria for selection focus on detailed content of candidate ontologies which are considered in two aspects: verification and evaluation.

Verification Considering the maturity level of current biomedical ontologies, verification of taxonomic knowledge in candidate ontologies is valuable for selection. Reasoners, such as Racer, Pellet, and FaCT++, can check consistency, incompleteness and redundancy in ontologies.

Evaluation First, we list a set of characteristics of ontologies that are of interest for text mining applications,

- *Coverage* Does an ontology cover the concepts or/and relations concerned in the application? Do their definitions meet the expected scope?

- *Taxonomic Knowledge* The two relations *is-a* and *part-of* have a central role in almost all ontologies (Smith et al., 2005). Do the locations of essential concepts in the *is-a* and *part-of* hierarchies meet the expectation?

- *Instance* The instantiation of concepts and the number of instances for each concept could effect greatly the performance of many text mining tasks such as entity recognition. Do they satisfy the requirements?

These characteristics may be evaluated,

- against domain experts or references if they exist. For example, if an ontology involves integration of literature with data from biological databases, schemas of databases can be the references.

- among candidate ontologies. There is no gold standard in this situation and therefore the candidate ontologies are compared directly to each other. For instance, aligning the ontologies can provide information about their similarities and differences regarding their coverage.

Component 3

The results from component 1 and 2 may not lead to a direct and confident decision. Also, in many cases there exist gold standards or benchmarks which we can use to evaluate our application. Therefore, the purpose of component 3 is to evaluate the use of ontologies in a real system. This component could provide a relatively straightforward method for evaluating the quality of ontologies for use in the application.

The field of biomedical text mining has expended considerable effort in building evaluation resources. A number of challenges, such as TREC Genomics track, BioCreative, BioNLP/JNLPBA 2004 and LLL05, have been organized in the community. They contribute to the creation of shared gold standard datasets, prepared by domain experts and also suggest evaluation measures. Comparison of the system individually bundled with candidate ontologies can be performed by using these gold standard datasets and measures.

4 Experiment

In this section we present an experiment in which we use our framework to select a biomedical ontology for the gene normalization task.

4.1 Scenario

The purpose of gene normalization (GN) is to link genes and proteins mentioned in the literature to their entries in structured databases of biological data. It has a substantial impact on tasks such as retrieval of relevant documents, identification of interesting genes and proteins, and relationships between them. The task is challenging even for scientists, since there is no community wide agreement on how a particular gene and gene product should be named. Heavy use of short forms (acronyms or abbreviations) in biomedical literature makes the problem worse. In (Tan, 2008) we developed an ontology-centred system for gene normalization. It relies on information about gene candidates, contexts of the symbol and external knowledge sources. Information about gene candidates is extracted from gene databases. Ontologies are used for semantic interpretation of contexts of gene symbols and identification of their relevance to gene candidate information. Normalization is based on matching contexts of the symbol to relevant information about gene candidates.

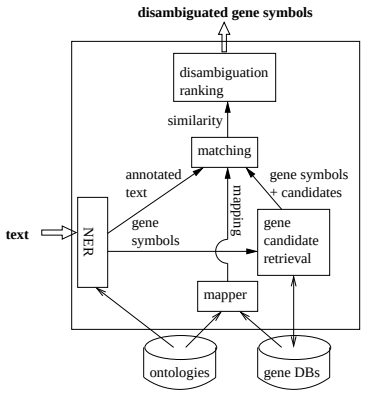


Figure 1: The components and data flows of the system

Figure 1 illustrates the components and data flows of the system. The system receives a raw text as the input, and yields database entries of genes appearing in the text: 1) *Mapper* receives the ontology and schemas of gene databases, and outputs a list of concepts from the ontology and their mappings to categories of information stored in gene databases. 2) *Named Entity Recognizer* (*NER*) recognizes relevant biomedical entities in text by linking them to the concepts of the ontology which have been mapped to categories of gene information in *mapper*. 3) *Gene candidate retrieval* retrieves gene candidates for each gene symbol. The categories of information about each gene candidate are collected. 4) *Matching* includes various algorithms that match contexts of a gene symbol to information about its gene candidates according to the mappings obtained in *mapper*, and returns similarity values between them. 5) *Normalization ranking* ranks gene candidates based on results of *Matching* for each gene symbol appearing in a text.

4.2 Selecting the ontology

In this application the components *NER* and *Mapper* count on the deployment of ontologies. The ontology-driven *NER* component involves associating text with the correct concepts in the ontology by means of associating mentions in the text with instances in the ontology. The ontology provides the formal representation of domain knowledge for the NER. The component *Mapper* involves mapping the

ontology to database schemas. The ontology provides the model to link information extracted from literature and data in biological databases.

Component 1

Following the criteria of the component 1, we define the requirements on the ontology.

Type Based on studies of categories of information stored in gene databases such as EntrezGene (Magloot et al., 2005), we decide that top-domain ontologies are suitable for our application.

Scope The ontology should represent domain knowledge about genes and gene products. The ontology must contain concepts covering the categories of gene information, e.g. location, functions, and biological process.

Representation In order to support the task of *NER*, the ontology must at least have concepts and instances.

Supporting Technologies In the system the supporting technologies include an ontology-driven NLP tool that support the task of *NER*, and ontology alignment algorithms, that can be used to map the ontology to categories of information in databases.

We look for suitable ontologies by searching OBO (OBO, 2009) and Bioportal, and reviewing literature on biomedical ontologies. There are a few ontologies covering genes and gene products such as GO, MeSH, the Unified Medical Language System (UMLS) knowledge sources (Lingberg et al., 1993) and ontologies from the GENIA project (Kim et al., 2003). Only two of these meet all the above requirements, the UMLS knowledge sources and the GENIA ontologies. The UMLS Semantic Network and GENIA entity and event ontologies can be seen as top-domain ontologies, of which the concepts are on the level of the requirement. The UMLS Metathesaurus is primarily concerned with names used in the domain. It contains a huge number of biomedical entities and events (called concepts in the Metathesaurus) and their various names from more than 100 biomedical vocabularies. All of the concepts are assigned to at least one semantic type in the Semantic Network. The MetaMap program (Aronson, 2001) is available to map text to the concepts and semantic type. The GENIA project collects a set of MEDLINE articles concerning transcription factors in human blood cells. Every biomedical entity and event appearing in the articles are identified with a concept in the GENIA ontology. Several NLP tools, e.g. LingPipe (LingPipe, 3.7.0), support statistical name entity recognition by using the GENIA corpus as training data.

Component 2

Verification We checked the consistency and redundancy in the UMLS Semantic Network and GENIA ontologies, respectively, by translating them into OWL format and then sending to the reasoner, Racer. Both of them are consistent and have minimal representations.

Evaluation We perform two evaluations. The first evaluation is to investigate the coverage of concepts of the two ontologies against categories of EntrezGene. The coverage determines the extent of information that can be used for gene normalization. In the second one we compare biomedical entities and events belonging to concepts of the two ontologies, since they influence the performance of *NER*.

- *Evaluation 1* Manually, we identify a list of categories of EntrezGene that are used for gene normalization. Evaluation is performed by aligning concepts of the two ontologies to the categories. A first observation is that UMLS covers more topics than GENIA and therefore may give better results for NER. The topics of the GENIA corpus can be represented by a query using human, blood cell, transcription factor. To be able to compare the coverage of UMLS and GENIA on an overlapping piece of a domain we align the two ontologies based on the GENIA topics. The evaluation is based on an instance-based strategy proposed in (Tan et al., 2006). The algorithm consists of the following steps: 1) *Instance collection*: We retrieve a list of human genes from the database using human, blood cell, and transcription factor. All entities appearing in the categories of information are collected as instances. 2) *Classification*: For each ontology, each instance is classified to concepts by a NLP tool. The MetaMap program is used for the UMLS Semantic Network, and the LingPipe entity recognition program is used for the GENIA ontology. An instance can be classified to more than one concept. 3) *Calculation*: The similarity between a concept from an ontology (A) and a category from the EntrezGene (B_j) is determined by the extent to which the instances of the category

EntrezGene	UMLS (value)	GENIA (value)
Name/Syn.	Gene or Genome (0,66)	protein_molecule (0,83)
Chromosome	Cell Component (1.0)	protein_molecule (1.0)
Map Location	Nucleotide Sequence (1.0)	DNA_domain_or_region (0,5) protein_molecule (0,4)
Protein	Gene or Genome (0,42) Amino Acid, Peptide, or Protein (0,25)	protein_molecule (0,60)
Pathway	Molecular Function (0,29) Cell Function (0,24)	other_name (0,76)
GOA_function	Molecular Function (0,75)	other_name (0,85)
GOA_component	Cell Component (0,96)	cell_component (0,40) other_name (0,15)
GOA_process	Cell Function (0,49) Molecular Function (0,16)	other_name (0,78)

Table 1: Comparison: EntrezGene, UMLS and GENIA

support the concept. The similarity value is calculated as: $sim(A, B_j) = \frac{\sum_{i=1}^{n} S(a_i, B_j)}{\sum_{k=1}^{m} \sum_{i=1}^{n} S(a_i, B_k)}$, where a_i are the instances belonging to A, n is the number of instances belonging to A, m is the number of categories, and S is a function calculating the degree to which an instance of an ontology concept supports a category from EntrezGene. S is defined as:

$$S(a_i, B_j) = \begin{cases} 0 & \textit{if } a_i \textit{ does not associate with } B_j \\ \frac{1}{p} & \textit{otherwise; } p \textit{ is the number of categories } B_k \\ & \textit{that } a_i \textit{ associates with} \end{cases}$$

Table 1 shows the alignment results. The left column gives the categories of gene information we use for normalization. If the similarity value between a category and the UMLS semantic type or GENIA concept is higher than 0.6, they are considered as corresponding to the category. If no similarity value is higher than 0.6, we list the two semantic types or concepts with the highest values. Three of eight alignments fall into this situation for UMLS semantic types, and two for GENIA concepts. We also note that the GENIA concept other_name appears 4 times, but the meaning of this concept is not well-defined. Most other categories are aligned to protein_molecule, although the categories are very different. In this evaluation, it is more likely that the UMLS semantic network is more appropriate than the GENIA ontology for our system.

- *Evaluation 2* 35,515 entities of interest in the biological domain are identified in the GENIA entity corpus, 7,089 entities of which have been found in the UMLS Metathesaurus. Since we could not obtain all the UMLS Metathesaurus concepts for each semantic type, we decided to compare the UMLS semantic types and GENIA concepts that appear in the first evaluation, based on these 7,089 shared instances. The comparison is based on the same instance-based alignment strategy used in the first evaluation. Tables 2 and 3 show the alignment results. In table 2 the value illustrates the extent to which instances of GENIA concepts support the UMLS semantic types. We list the three concepts with the highest values in the table, if they exist. The concepts in italics are event concepts, the others are entities. All the UMLS semantic types cover instances from more than one GENIA concept belonging to a subtree. For example, Gene or Genome could be aligned to the three GENIA concepts DNA_family_or_group, DNA_domain_or_region and RNA_molecule, which are leaves of the subtree Nucleotide_acid_compound. The granularity of the part of the GENIA ontology concerning the application is finer than the corresponding part of the UMLS semantic network. Table 3 shows the distribution of the instances of GENIA concepts when they support UMLS semantic types. Instances of DNA_domain_or_region and cell_component mainly support Gene or Genome and Cell Component, respectively, which is consistent with our observation in table 1. Another observation from this table is that protein_molecule covers both Gene or Genome and Amino Acid, Peptide, or Protein. The result also shows that other_name is not well defined. However, this evaluation does not give a conclusive answer to which ontology supports the task of *NER* better.

Component 3

Since component 1 and 2 could not lead to a direct and confident decision, we evaluate the use of the two ontologies in our system We use a training dataset from the BioCreative II gene normalization task (Morgan et al., 2008) in the evaluation. The BioCreative datasets contain Medline abstracts along with the EntrezGene identifiers corresponding to the human genes and direct gene products appearing in the abstracts. The training dataset contain 281 Medline abstracts and 640 human genes. The MetaMap program is the NLP tool that uses the UMLS Semantic Network, and the LingPipe entity recognition program employs GENIA ontology.

Table 4 illustrates the quality of normalization in the two systems. The quality is measured in terms of precision, recall and f-measure. Recall (Re.) is defined as the number of correctly disambiguated gene symbols divided by the number of the gene symbols to be disambiguated. Precision is measured in

GENIA	UMLS	value
DNA_family_or_group	Gene or Genome	0.63
DNA_domain_or_region	Gene or Genome	0.54
RNA_molecule	Gene or Genome	0.44
DNA_N/A	Nucleotide Sequence	0.23
DNA_substructure	Nucleotide Sequence	0.23
DNA_domain_or_region	Nucleotide Sequence	0.17
cell_component	Cell Component	0.76
RNA_N/A	Cell Component	0.50
DNA_molecule	Cell Component	0.46
protein_family_or_group	Amino Acid, Peptide, or Protein	0.37
protein_subunit	Amino Acid, Peptide, or Protein	0.33
amino_acid_monomer	Amino Acid, Peptide, or Protein	0.33
protein_domain_or_region	Amino Acid Sequence	0.42
Cellular_physiological_process	Cell Function	0.26
Cell_communication	Cell Function	0.24
Cell_differentiation	Cell Function	0.19
Protein_amino_acid_deacetylation	Molecular Function	1.0
Protein_amino_acid_dephosphorylation	Molecular Function	0.80
Protein_ubiquitination	Molecular Function	0.50

Table 2: Comparison: UMLS and GENIA (1)

GENIA	UMLS	value
protein_molecule	Gene or Genome	0.30
protein_molecule	Amino Acid, Peptide, or Protein	0.29
protein_molecule	Biologically Active Substance	0.10
DNA_domain_or_region	Gene or Genome	0.54
DNA_domain_or_region	Biologically Active Substance	0.07
DNA_domain_or_region	Nucleotide Sequence	0.06
cell_component	Cell Component	0.76
cell_component	Biomedical or Dental Material	0.03
cell_component	Amino Acid, Peptide, or Protein	0.02
other_name	Disease or Syndrome	0.15
other_name	Cell Function	0.09
other_name	Neoplastic Process	0.08

Table 3: Comparison: UMLS and GENIA (2)

two ways. Pre.-1 is defined as the number of gene symbols correctly and uniquely identified to the real gene, divided by the total number of genes proposed in the result regarding the whole data set. Pre.-2 is computed as the average of the precision of disambiguation for each gene symbol. F-measure is the evenly weighted harmonic mean of precision and recall. Pre.-1 and Pre.-2 are used in the computation of Fm.-1 and Fm.-2, respectively. For both datasets the quality of normalization from the system bundled with the UMLS is better than the one with the GENIA ontology.

Result

Overall, the UMLS knowledge source can be considered as the ontology that is most appropriate for our gene normalization system. The ontology covers the subject, genes and gene products, well. The meaning of the concepts is defined well enough for the use in the application. The granularity of the part of the ontology meets the need of the task. The system bundled with the UMLS and its supporting technologies produced better results in a gold standard dataset than the other one.

KB	Dataset	Pre.-1	Pre.-2	Re.	Fm.-1	Fm.-2
GENIA	dataset 1	0.45	0.65	0.78	0.57	0.71
	dataset 2	0.50	0.63	0.72	0.59	0.67
UMLS	dataset 1	0.48	0.69	0.82	0.61	0.75
	dataset 2	0.52	0.67	0.78	0.62	0.72

Table 4: Quality of normalization

Although we have not run additional expirements to confirm that the we have indeed made the right selection for our gene normalization system, the tests do corrobarate our results. The chosen ontology leads to a better result for both datasets that we used. Therefore, each dataset can be seen as a confirmation of the framework where we only used the other dataset.

5 Conclusions

In this paper we proposed a framework for selecting an appropriate ontology for a particular biomedical text mining application. The framework deals with ontology selection in three components, each of which considers different aspects of requirements of text mining applications on ontologies. Then we present an experiment in which we select a biomedical ontology for a gene normalization system, using the framework. Within the framework, evaluation results lead us to a relatively concrete choice of an ontology for our system.

In the future we want to evaluate our framework with more applications and ontologies. Further, currently there is no service to support ontology selection for biomedical text mining. Therefore, an important track for future work is to build or extend an existing portal with information about the ontologies and their use in text mining applications that is structured according to our framework. The information in such a portal will constitute valuable data and experiences regarding ontology selection that will be useful for future applications.

Acknowledgments

We thank Bo Servenius for comments on the project, Qiang Liu for generating part of the data and David Byers for proof-reading the paper. We also acknowledge the financial support of the Center for Industrial Information Technology.

References

Alani H, Brewster C. (2005) Ontology Ranking based on the Analysis of Concept Structures. *Proceedings of the 3th International Conference on Knowledge Capture*, 51-58.

Alias-i. 2008. LingPipe 3.7.0. http://alias-i.com/lingpipe

Ananiadou S, McNaught J. (editors) (2006) *Text Mining for Biology and Biomedicine*. Artech House.

Aronson AR. (2001) Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proceedings of the AMIA Symposium*, 17-21.

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics*, 25(1):25-9.

Bioportal. http://bioportal.bioontology.org/

Brewster C, Alani H, Dasmahapatra S, Wilks Y. (2004) Data Driven Ontology Evaluation. *International Conference on Language Resources and Evaluation*.

Buitelaar P, Eigner T, Declerck T. (2004) OntoSelect: A Dynamic Ontology Library with Support for Ontology Selection. *Proceedings of the Demo Session at the International Semantic Web Conference*.

Cohen AM, Hersh WR. (2005) A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 6:57-71.

Gangemi A, Catenacci C, Ciaramita M, Lehmann J. (2006) Modelling Ontology Evaluation and Validation. *Proceedings of the 2006 European Semantic Web Conference*.

Hristovski D, Peterlin B, Mitchell JA, Humphrey SM. (2005) Using literature-based discovery to identify disease candidate genes. *International Journal of Medical Informatics*, 74(2-4):289-298.

Kim J-D, Tomoko O, Yuka T, Tsujii J. (2003). GENIA corpus - a semantically annotated corpus for biotextmining. *Bioinformatics*, 19(suppl. 1):180-182.

Lambrix P, Tan H, Jakoniene V, Strömbäck L. (2007) Biological Ontologies. *Chapter 4 in Baker, Cheung (eds), Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences*, 85-99, Springer.

Lambrix P, Tan H. (2008) Ontology alignment and merging. *Chapter 6 in Burger, Davidson, Baldock, (eds), Anatomy Ontologies for Bioinformatics: Principles and Practice*, 133-150, Springer.

Lindberg DA, Humphreys BL, McCray AT.(1993) The Unified Medical Language System. *Methods of Information in Medicine*, 32:281 -91.

Lowe HJ, Barnett GO. (1994) Understanding and using the medical subject headings (MeSH) vocabulary to perform literature searches. *Journal of the American Medical Association*, 271(14):1103-1108.

Lozano-Tello A, Gómez-Pérez A. (2004) ONTOMETRIC: A Method to Choose the Appropriate Ontology. *Journal of Database Management. Special Issue on Ontological Analysis, Evaluation, and Engineering of Business Systems Analysis Methods*, 15(2):1-18.

Maglott D, Ostell J, Pruitt KD, Tatusova T. (2005) Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*. 33:D54-8.

Morgan AA, Lu Z, Wang X, Cohen AM, Fluck J, Ruch P, Divoli A, Fundel K, Leaman R, Hakenberg J, Sun C, Liu HH, Torres R, Krauthammer M, Lau WW, Liu H, Hsu CN, Schuemie M, Cohen KB, Hirschman L. (2008) Overview of BioCreative II gene normalization. *Genome Biology*, 9(Suppl 2):S3.

OBO. http://obofoundry.org/

Porzel R, Malaka R. (2004) A task-based approach for ontology evaluation. *ECAI 2004 Workshop Ontology Learning and Population* .

Rosse C, Mejino JL Jr. (2003) A reference ontology for biomedical informatics: the foundational model of anatomy. *Journal of Biomedical Informatics*, 36(6):478-500.

Sabou M, Lopez V, Motta E, Uren V. (2006) Ontology Selection: Ontology Evaluation on the Real Semantic Web. *Proceedings of the 4th International EON Workshop, Evaluation of Ontologies for the Web*.

Smith B. (2004) Beyond concepts: ontology as reality representation. *Proceedings of Formal Ontology and Information Systems*, 73-78.

Smith B, Ceusters W, Klagges B, Köhler J, Kumar A, Lomax J, Mungall C, Neuhaus F, Rector AL, Rosse C. (2005) Relations in biomedical ontologies. *Genome Biology*, 6(5):R46.

Spasic I, Ananiadou S, McNaught J, Kumar A. (2005) Text mining and ontologies in biomedicine: making sense of raw text. *Briefings in Bioinformatics*, 6(3):239-251.

Stenzhorn H, Schulz S, Beißwanger E. (2007) Towards a Top-Domain Ontology for Linking Biomedical Ontologies *Proceedings of the 12th World Congress on Health (Medical) Informatics*.

Tan H. (2008) Knowledge-based Gene Symbol Disambiguation. *CIKM: Proceedings of the 2nd international workshop on Data and text mining in bioinformatics*, 73-76.

Tan H, Jakoniene V, Lambrix P, Aberg J, Shahmehri N. (2006) Alignment of Biomedical Ontologies using Life Science Literature. *Proceedings of the International Workshop on Knowledge Discovery in Life Science Literature*, 1-17.

Investigation of Unsupervised Pattern Learning Techniques for Bootstrap Construction of a Medical Treatment Lexicon

Rong Xu, Alex Morgan, Amar K Das
Biomedical Informatics Program
Stanford University
Stanford, CA 94305, USA
xurong@stanford.edu

Alan Garber
Primary Care and Outcomes Research
Stanford University
Stanford, CA94305, USA

Abstract

Dictionaries of biomedical concepts (e.g. diseases, medical treatments) are critical source of background knowledge for systems doing biomedical information retrieval, extraction, and automated discovery. However, the rapid pace of biomedical research and the lack of constraints on usage ensure that such dictionaries are incomplete. Focusing on medical treatment concepts (e.g. drugs, medical procedures and medical devices), we have developed an unsupervised, iterative pattern learning approach for constructing a comprehensive dictionary of medical treatment terms from randomized clinical trial (RCT) abstracts. We have investigated different methods of seeding, either with a seed pattern or seed instances (terms), and have compared different ranking methods for ranking extracted context patterns and instances. When used to identify treatment concepts from 100 randomly chosen, manually annotated RCT abstracts, our medical treatment dictionary shows better performance (precision:0.40, recall: 0.92 and F-measure: 0.54) over the most widely used manually created medical treatment terminology (precision: 0.41, recall: 0.52 and F-measure: 0.42).

1 Introduction

Dictionary based natural language processing systems have been widely used in recognizing medical concepts from free text. For example, the MetaMap program is used to map medical text to concepts from the most widely used biomedical terminology, the Unified Medical Language System (UMLS) Metathesaurus (Aronson, 2000). It identifies various forms of UMLS concepts in text and returns them as a ranked list using a five-step process: identifying simple noun phrases (NP's), generating variants of each phrase, finding matched phrases, assigning scores to matched phrases and composing mappings. However, its performance largely depends on the quality of the underlying UMLS Metathesaurus and its manually created rules and variants. One study has shown that, of the medical concepts identified by human subjects, more than 40% were not in UMLS (Pratt, 2003). Other examples of mapping text to controlled biomedical terminologies include (Cohen, 2005) and (Fang, 2006). Many other systems make heavy use of biomedical terminologies directly such as the work of Blaschke, et al. (Blaschke, 2002) and Friedman et al. (Friedman, 2001).

Biomedical terminology is highly dynamic, both because biomedical research is itself highly dynamic, but also because there are essentially no constraints on the use of new terminological variants, making the terms used in free text quite different from the canonical forms listed in controlled terminologies. To contrast UMLS with actual text mentions, there are 150 different *chemotherapy* concepts in UMLS. The majority of these terms derive from the diseases they are used to treat. For example *cancer chemotherapy*, *AIDS chemotherapy*, *brain disorder chemotherapy*, and *alcoholism chemotherapy*. On the other hand, we have identified more than 1,000 different *chemotherapy* types mentioned in RCT (Randomized Clinical Trial) report abstracts, with most of the names derived

Proceedings of the Workshop on BioNLP, pages 63–70,
Boulder, Colorado, June 2009.

from the chemicals contained in the chemotherapy regimen, such as *platinum-based chemotherapy* or *fluorouracil-based chemotherapy*. There is little overlap between the *chemotherapy* terms in UMLS and the ones used in RCT abstracts. Even for simple drug names as *5-fluorouracil* and *tamoxifen*, there are many clinically distinct and important variants of these drugs which are absent in UMLS as distinct terms/concepts, such as *intralesional 5-fluorouracil*, *topical 5-fluorouracil*, *intrahepatic arterial 5-Fluorouracil*, *adjuvant sequential tamoxifen*, and *neoadjuvant tamoxifen*.

There has been considerable work on expanding the coverage of biomedical dictionaries through morphological variants, but these approaches require an initial term dictionary with reasonable extensive coverage. Examples include the approaches developed by Krauthammer and Nenadic (Krauthammer, 2004), Tsuruoka and Tsujii (Tsuruoka, 2004) & (Tsuruoka, 2003), Bodenreider, et al. (Bodenreider, 2002), and Mukherjea and colleagues (Mukherjea, 2004). An important shortcoming with static, human derived terminologies that cannot easily be addressed by looking for variants of existing terms is the fact that continual developments in medical therapies constantly gives rise to new terms. Examples include, *Apomab*, *Bapineuzumab*, *Bavituximab*, *Etaracizumab*, and *Figitumumab*. These all represent a new generation of targeted biological agents currently in clinical trials none of which appear in UMLS. Clearly we need to develop techniques to deal with this dynamic terminology landscape.

MEDLINE is the most extensive and authoritative source of biomedical information. Large quantities of biomedical text are available in MEDLINE's collection of RCT reports with over 500,000 abstracts available. RCT reports are a critical resource for information about diseases, their treatments, and treatment efficacy. These reports have the advantage of being highly redundant (a disease or treatment name is often reported in multiple RCT abstracts), medically related, coherent in writing style, trustworthy and freely available.

In our recent study (Xu, 2008), we have developed and evaluated an automated, unsupervised, iterative pattern learning approach for constructing a comprehensive disease dictionary from RCT abstracts. When used to identify disease concepts from 100 manually annotated clinical abstracts, the disease dictionary shows significant performance improvement (F1 increased by 35-88%) over UMLS and other disease terminologies. It remained to be demonstrated that these bootstrapping techniques are indeed rapidly retargetable and can be extended to other situations, and so we have extended our scope to investigate medical treatment names in addition to disease terms in this work.

Our approach is inspired by the framework adopted in several bootstrapping systems for learning term dictionaries, including (Brin, 1998), (**?**), and (Agichtein, 2000). These approaches are based on a set of surface patterns (Hearst , 1992), which are matched to the text collection and used to find instance-concept relations. Similar systems include that of Snow and colleagues (Snow, 2005), which integrates syntactic dependency structure into pattern representation and has been applied to the task of learning instance-of relations, and the approach developed of Caprosaso, et al. (Caprosaso, 2007) which focussed on learning text context patterns to identify mentions of point mutations.

All iterative learning systems suffer from the inevitable problem of spurious patterns and instances introduced in the iterative process. To analyze different approaches to addressing this issue, we have compared three different approaches to ranking extracted patterns and three different approaches to ranking extracted instances. Because such systems also depend on an initial seeding with either a seed pattern or term instance, an important question is whether these different starting points lead to different results. We investigated this issue by starting from each point separately and compared the final results.

2 Data and Methods

2.1 Data

509,308 RCT abstracts published in MEDLINE from 1965 to 2008 were parsed into 8,252,797 sentences. Each sentence was lexically parsed to generate a parse tree using the Stanford Parser. The Stanford Parser (Klein, 2003) is an unlexicalized natural language parser, trained on a non-medical document collection (Wall Street Journal). We used

the publicly available information retrieval library, Lucene, to create an index on sentences and their corresponding parse trees. For evaluation and comparison, 241,793 treatment terms with treatment related semantics types from UMLS were used.

2.2 Unsupervised Instance Extraction and Pattern Discovery

Figure 1 describes the bootstrapping algorithm used in learning instances of treatment and their associated text patterns. The algorithm can operate in two modes, either starting with a seed pattern p_0, which represents a typical way of writing about treatments, or a set of seed instances, (d_i). For example, the seed pattern we used was "*treated with NP*" (*NP*: noun phrase). The program loops over a procedure consisting of two steps: instance extraction and pattern discovery. In the instance extraction step, patterns are used as search queries to the local search engine. The parse trees with given patterns are retrieved and noun phrases (instances of treatments) following the pattern are matched from the parse trees. In the pattern discovery step, instances extracted from the previous iteration are used as search queries to the local search engine. Corresponding sentences containing instance mentions are retrieved and the bigrams (two words) in front of instances are extracted as patterns. When seeding with an initial pattern, only two iterations are typically needed, as experience shows that most of reliable patterns and instances have been discovered at this stage. The algorithm stops after a single iteration when seeding with a list of instances.

2.3 Selecting Seed Instances

Of the 241,793 treatment related terms in the UMLS, only about 22,000 (9%) of these have appeared in MEDLINE RCT reports. We randomly selected 500 drug terms and 500 medical procedure terms from the 22,000 terms as seed instances and used them in the pattern discovery system described above.

2.4 Pattern Ranking

A newly discovered pattern is scored on how similar its output (instances associated with the pattern) is to the output of the initial seed pattern. Intuitively, a reliable pattern is one that is both highly

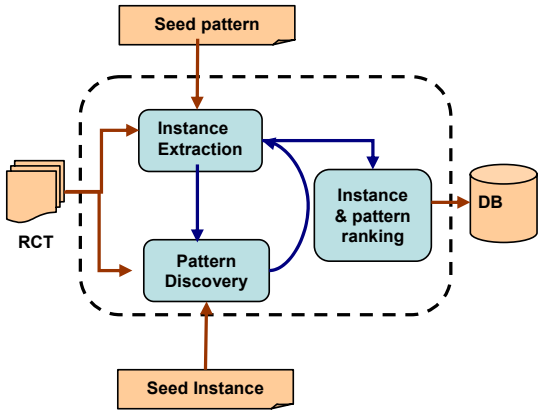


Figure 1: General scheme of the iterative method.

precise (high precision) and general (high recall). Using the output instances from the seed pattern p_0 as a comparison, we developed *Precision Based*, *Recall Based*, and *F1 Based* algorithms to rank patterns. We define $instances(p)$ to be the set of instances matched by pattern p, and the intersection $instances(p) \bigcap instances(p_0)$ as the set of instances matched by both pattern p and p_0.

1. *Precision Based* rank:

$$score1(p) = \frac{instances(p) \bigcap instances(p_0)}{instances(p)} \quad (1)$$

   The *precision based* ranking method favors specific patterns.

2. *Recall Based* rank:

$$score2(p) = \frac{instances(p) \bigcap instances(p_0)}{instances(p_0)} \quad (2)$$

   The *recall based* ranking method favors general patterns.

3. *F1 based* rank:

$$score3(p) = \frac{2 \times score1(p) \times score2(p)}{score1(p) + score2(p)} \quad (3)$$

   A combination of the *Precision Based* and the *Recall Based* evaluation methods is the *F1*

Based ranking method, which takes into account both pattern specificity and pattern generality. This method favors general patterns while penalizing overly specific patterns.

2.5 Instance Ranking

A reliable instance is one that is associated with a reliable pattern many times. We experimented with three ranking algorithms:

1. *Abundance Based* rank: A treatment instance(d) that is obtained multiple times is more likely to be a real treatment concept when compared with one that has only a single mention in the whole corpus. We define $scoreA(d)$ as the number of times where d appears in the corpus.

2. *Pattern Based* rank: A treatment instance obtained from multiple patterns is more likely to be a real treatment concept when compared with the one that was obtained by a single pattern (p). *Pattern Based* rank takes into account the number of patterns that generated the instance, score of those patterns, and the number of times that the instance is associated with each pattern ($count(p, d)$).

$$scoreB(d) = \sum_{i=0}^{n} \log score3(p_i) \times count(p_i, d) \quad (4)$$

3. *Best Pattern Based* rank: A treatment instance obtained from a highly ranked pattern is more likely to be a real treatment concept when compared with the one that was obtained from a poorly ranked pattern. First the instances are ranked by the best pattern (p_b) that generated the instances and then further ties are broken by the number of times the instance is associated with that pattern ($count(p, d)$) to provide $scoreC(d)$.

2.6 Comparison of Patterns Derived from Different Seed Types

The patterns extracted when starting with either seed instances or a seed pattern are ranked by the *recall based* method and *F1-based* method, then the overlaps at different cutoffs are measured to assess the similarity of the patterns discovered by starting with the different starting seed types.

2.7 Evaluation of Stanford Parser in Identifying Treatment Noun Phrase

An important question is how accurate the Stanford Parser is at identifying the relevant term boundaries. We used manually curated treatment names from UMLS to measure the accuracy of the Stanford Parser in identifying treatment noun phrases. With $NPcount(treatment)$ defined as number of times that the Stanford Parser identifies a treatment as noun phrase or part of a noun phrase in the data and $count(treatment)$ as number of times the treatment appears in the data.

$$accuracy = \frac{1}{n} \sum_{i=0}^{n} \left(\frac{NPcount(d_i)}{count(d_i)} \right) \quad (5)$$

2.8 Evaluation of the extracted treatment lexicon

We assessed the quality (precision and recall) of our lexicon by using it to identify treatment concepts in 100 randomly selected RCT abstracts where treatment names were manually identified. In addition, we also compared the performance of our lexicon with that of UMLS.

3 Results

3.1 Evaluation of Stanford Parser in Identifying Treatment Noun Phrases

Even though the Stanford Parser is trained on non-medical data, it is highly accurate in identifying treatments as noun phrases or parts of a noun phrase with accuracy of 0.95. The reason may be that medical treatments are indeed often noun phrases or parts of a noun phrase in RCT reports, and there are strong syntactical signals for their phrasal roles in the sentences. For example, treatments are often either the object of a preposition (e.g. *efficacy of fluorouracil* and *treated with fluorouracil*) or the subject of a sentence (e.g. *fluorouracil is effective in treating colon cancer*).

3.2 Comparison between Seed Types

There is considerable overlap in discovered patterns between starting with a single seed pattern and start-

ing with the 1,000 seed instances and little difference in overall performance. 12,241 patterns are found to be associated with the 1,000 seed treatment instances. However, only the most highly ranked patterns are relevant (see Evaluation of The Extracted Treatment Lexicon, below). Table 1 shows the intersection of the top ranked patterns between both seeding methods at different rank cut-offs. We find a very high level of intersection between the top ranked patterns from both initial seed types, for example eighteen of the top twenty patterns are identical. These results indicate that starting from either seed type leads to very similar results.

Rank	Recall Based	F1 Based
10	0.90	0.80
20	0.90	0.90
30	0.87	0.80
40	0.83	0.85
50	0.84	0.82
60	0.82	0.85
70	0.82	0.79
80	0.83	0.84
90	0.84	0.83
100	0.82	0.83

Table 1: : The ratio of overlap in the top ranking patterns discovered by different seed types

3.3 Pattern Ranking

Similar to the results observed in our previous study (Xu, 2008), the *Precision Based* metric assigns high scores to very specific but not generalizable patterns such as "*lornoxicam versus*" (Table 2), which appears only once in the data collection, while the top 10 patterns based on the *Recall Based* and *F1 Based* rankings are typical treatment related patterns. When a different seed pattern "*efficacy of*" was used, the top 10 patterns were the same with a different rank ordering.

3.4 Instance ranking

Table 3 shows the top 10 suggested treatment names when using "*treated with*" as the initial seed pattern. The rank of a proposed treatment instance is determined by the different ranking methods: *Abundance Based*, *Pattern Based*, or *Best Pattern Based* ranking

#	Precision based	Recall based	F1 based
1	beta-blockers nor	treated with	treated with
2	lornoxicam versus	treatment with	treatment with
3	piroxantrone and	effects of	efficacy of
4	heparin called	efficacy of	effects of
5	anesthetics containing	dose of	dose of
6	antioestrogens and	doses of	doses of
7	markedly adsorb	suggest that	suggest that
8	recover following	study of	safety of
9	Phisoderm and	response to	response to
10	MitoExtra and	effect of	effect of

Table 2: Top 10 patterns with "*treated with*" as seed pattern

algorithms. None of the top 10 extracted phrases on the basis of *Abundance Based* or *Pattern Based* are actual treatment names. These two ranking methods assign high ranks to common, non-specific phrases. The *Best Pattern Based* ranking method correctly identifies specific treatment mentions, mainly because it reduces the likelihood of selecting irrelevant patterns.

#	Abundance based	Pattern based	Best pattern based
1	patients	patients	placebo
2	treatments	the treatment	chemotherapy
3	the treatments	treatments	radiotherapy
4	children	the use	tamoxifen
5	the effect	children	antibiotics
6	no significant differences	surgery	insulin
7	placebo	the patients	interferon
8	surgery	changes	surgery
9	the effects	women	corticosteroids
10	the study	use	cisplatin

Table 3: Top 10 treatments when using "*treated with*" as the seed pattern

3.5 Evaluation of the Extracted Treatment Lexicon

Our dictionary derived from using "*treated with*" as the seed pattern with two bootstrapping itera-

Count	Cutoff	Precision	Recall	F1
17,683	1.0%	0.404	0.921	0.540
88,415	5%	0.127	1.0	0.22
132,623	7.5%	0.105	1.0	0.187
176,832	10%	0.088	1.0	0.160

Table 4: Precision, recall and F1 at 4 cutoff values

tions consists of 1,768,320 candidate instances and 78,037 patterns, each with an accompanying confidence score. The top 20 patterns are associated with more than 90% of the instances. We evaluated the quality of the dictionary by using it to identify treatment concepts in 100 randomly selected abstracts where treatment names were manually annotated. There were an average of three treatment names per test abstract. Table 4 shows the precision, recall and F1 values when instances are ranked by the *best pattern based* ranking method (ScoreC). The precision, recall and F1 values at each cut-off (percentage of all instances) were averaged across the 100 abstracts.

The precision, recall and F1 of the UMLS Metathesaurus in identifying treatment names from the test dataset are 0.41, 0.52 and 0.42 respectively. The performance using UMLS on this task is consistent with a previous study (Pratt, 2003). The low precision may due to the fact that UMLS often tags irrelevant names as treatment related names. For example, common, non-specific terms such as *drug*, *agent*, *treatment* and *procedure* appear in the dictionary derived from UMLS. However, we chose not to edit the lexicon derived from UMLS as it is unclear how to do so in a systematic matter without essentially creating a new version of UMLS, and we are interested in studying methods that do not rely on any human involvement (our Discussion describes the possible inclusion of human judgments). Also, the low recall of UMLS is not surprising given the fact that the names specified in UMLS are often not the terms authors use in writing. The performance of our dictionary (precision: 0.40, recall: 0.92, F1: 0.54) is a dramatic improvement over using UMLS. Our recall is high since all the terms are learned from the literature directly and exemplify the manner in which authors write RCT reports. However, the precision of our dictionary is still low (see Discussion).

4 Discussion

We have demonstrated an automated, unsupervised, iterative pattern learning approach for bootstrapping construction of a comprehensive treatment lexicon. We also compared different pattern and instance ranking methods and different initial seed types (instances or patterns). On the task of term identification, use of our boostrapped lexicon increased performance over using the most widely used manually curated terminology (UMLS). We have extended our previous work to the identification of new terminology types, demonstrating the versatility of this approach. Our approach may also be used with other data sources such as general health related web pages. However, there is still significant space in which to seek improvement in increasing the coverage of our lexicon and the quality of our patterns.

Although useful in demonstrating the proof of concept and allowing us to examine different ranking methods, focusing on bigrams that precede noun-phrases limited the space of patterns that we could potentially examine. More complex patterns might be involved. For example, in the sentence "*Pravastatin is an effective and safe drug*" (PMID 08339527), there is a distinctive treatment related pattern "*NP is an effective and safe drug*" that our technique does not capture. However, most key terms are mentioned in multiple contexts. For example, *Pravastatin* appears with the seed pattern *treatment with* more than 200 times. As our corpus of literature increases, redundancy will increase the likelihood of a treatment term being matched by the type of patterns we recognize. The rapid growth of biomedical knowledge and literature, which makes our automatically generated medical treatment vocabulary necessary, can also act to increase its coverage over time.

In order to keep our algorithm simple, we did not perform deep grammatical analysis. For example, in the sentence "*Treatment of the subjects with atorvastatin decreased the abundance of IL-12p35 mRNA in mononuclear cells*" (PMID 12492458), *atorvastatin* is associated with *treatment of*, not *subjects with*. Since our algorithms simply extracts the two words in front of treatment names, *subjects with* will be extracted as treatment related pattern. In fact, *subjects with* is a disease related pattern in RCT reports, for

example "*34 subjects with asthma*". But our pattern ranking algorithm will assign a low score to *subjects with* since the terms associated with this pattern are more disease related and have little overlap with the output of the seed pattern *treatment with*.

Our instance ranking assigns high confidence scores to common and non specific terms like *this drug*, *the treatment* or *this procedure* since they are often associated with highly ranked patterns many times. These anaphoric terms often refer to treatment names previously specified. There are at least two ways to address this problem. The first is to assign low scores to terms starting with a determiner such as *the* or *this*. Another way to improve the instance ranking algorithm is to take into account of the overall context of the term. For example, these anaphora often appear in specific sections of RCT reports such as the *result* section, and refer to terms from previous sections. Specific examples include "*Treatment with this drug should be attempted in intractable cases*" (PMID 09038009) and "*The efficacy of the treatment was 88 and 95% in group 1 and 2, respectively*" (PMID 14520944). The terms from *title*, *background* or *conclusion* sections could be assigned higher scores than the ones from *result* section. Beyond these simple heuristics, more sophisticated approaches might take advantage of the work in anaphora resolution, such as (Baldwin, 2001).

The lexicon consists of terms with mixed hierarchies, including general terms as *chemotherapy*, *surgery*, *corticosteroids*, *antibiotics*, and specific terms as *fluorouracil*, *oral or intravenous 5-Fluorouracil*, *cisplatin*, *nephrectomy*. In order to make this dictionary more useful, additional work is needed to organize the terms and build ontologies based on the lexicon.

Previous work has shown that learning multiple semantic types simultaneously can improve precision (Thelen, 2002) & (Curran, 2007), and it remains to be seen if that approach can be combined with the prioritization of pattern and extracted instance rankings here to give better overall performance. Other possible extensions and improvements include various approaches to slow the learning process and discover new patterns and instances more conservatively, at the expense of more iterations. Further improvements can be expected from integrating active learning approaches to include the involvement of a human judge in the process, analogous to the tag-a-little, learn-a-little method proposed as part of the Alembic Workbench (Day, 1997). Because our approach ranks both extracted patterns and instances, it is amenable to such techniques. Indeed, active learning has been found to provide considerable gains in corpus annotation (Tomanek, 2007) & (Buyko, 2007), and can be a model for semi-automated terminology compilation.

All the data and code are available on request from the author.

Acknowledgments

RX is supported by NLM training grant LM007033 and Stanford Medical School.

References

E. Agichtein, L Gravano. 2000. *Snowball: extracting relations from large plaintext collections*, *In Proc of the 5th ACM conference on Digital libraries* .

A.R. Aronson 2001. *Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc AMIA Symp*:17-21.

B. Baldwin 2001. *Text and knowledge mining for coreference resolution. Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*:1-8.

C. Blaschke, A. Valencia. 2002. *The frame-based module of the SUISEKI information extraction system*, *Intelligent Systems, IEEE,* 17; 2:14 - 20.

O. Bodenreider, T.C. Rindflesch, A. Burgun, 2002. *Unsupervised, corpus-based method for extending a biomedical terminology. Proc of the ACL-02 workshop on Natural language processing in the biomedical domain*: 53–60.

S. Brin 1998. *Extracting patterns and relations from the world wide web. WebDB Workshop at 6th International Conference on Extending Database Technology*

E. Buyko, S. Piao, Y. Tsuruoka, K. Tomanek , J.D. Kim, J. McNaught, U. Hahn, J. Su, and S. Ananiadou. 2007, *Bootstrep annotation scheme: Encoding information for text mining*, *Proc of the 4th Corpus Linguistics Conference*, Birmingham, July 27-30.

J. G. Caprosaso, W.A. Baumgartner, D.A. Randolph, K.B. Cohen, L. Hunter 2007. *Rapid pattern development for concept recognition systems: application to point mutations.*, *Journal of Bioinformatics and Computational Biology*, Vol. 5, No. 6, 12331259.

A. Cohen 2005. *Unsupervised gene/protein named entity normalization using automatically extracted dictionaries*. *Proc of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases*: 17-24.

M. Collins, Y. Singer 1999. *Unsupervised Models for Named Entity Classification*. *EMNLP*

J.R. Curran, T Murphy, B Scholz 2007. *Minimizing Semantic Drift With Mutual Exclusion Bootstrapping*, *Proc of the 10th Conference of PACL*:172-180.

D. Day, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson, M. Vilain 1997, *Mixed-initiative development of language processing systems*. *Proc of the 5th ACL Conference on Applied Natural Language Processing*

H. Fang, K. Murphy, Y. Jin, J.S. Kim, P.S. White 2006. *Human Gene Name Normalization using Text Matching with Automatically Extracted Synonym Dictionaries*. *Proc of the BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL 06*: 4148.

C. Friedman, P. Kra, H. Yu, M. Krauthammer, A. Rzhetsky. 2001. *GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles*. *Bioinformatics*, ;17 Suppl 1:S74-82.

M.A. Hearst 1992. *Automatic acquisition of hyponyms from large text corpora*, *Proc of the 14th conference on computational linguistics*.

D. Klein D, CD. Manning 2003. *Accurate Unlexicalized Parsing*, *Proc of the 41st Meeting of the Association for Computational Linguistics*, 2003; 423-30.

M. Krauthammer G. Nenadic 2004. *Term identification in the biomedical literature.*, *J Biomed Inform*, Dec;37(6):512-26.

S. Mukherjea, L.V. Subramaniam, G. Chanda, S. Sankararaman, R. Kothari, V.S. Batra, D.N. Bhardwaj, B.Srivastava 2004. *Enhancing a biomedical information extraction system with dictionary mining and context disambiguation*, *IBM Journal of Research and Development*, 48(5-6): 693-702

W. Pratt, M. Yetisgen-Yildiz 2003 *A Study of Biomedical Concept Identification: MetaMap vs. People*, *Proc AMIA Symp*, 529-533.

R. Snow, D. Jurafsky, A. Ng 2005. *Learning syntactic patterns for automatic hypernym discovery*, *Proc of the 17th Conference on Advances in Neural Information Processing Systems* MIT Press.

M. Thelen, E. Riloff 2002. *A Bootstrapping Method for Learning Semantic Lexicons Using Extraction Pattern Contexts*, *Proc of EMNLP*.

K. Tomanek, J Wermter, U Hahn. 2007. *An Approach to Text Corpus Construction which Cuts Annotation Costs and Maintains Reusability of Annotated Data*, *Proc of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*:486-495.

Y. Tsuruoka, J. Tsujii 2003, *Boosting Precision and Recall of Dictionary-Based Protein Name Recognition*, *Proc of the ACL 2003 Workshop on NLP in Biomedicine*:41-8.

Y. Tsuruoka, J. Tsujii 2004, *Improving the performance of dictionary-based approaches in protein name recognition*, *J of Biomed Inf* 37, 6; December: 461-470.

R. Xu, K. Supekar, A. Morgan, A.Das, A. Garber 2008. *Unsupervised Method for Automatic Construction of a Disease Dictionary from a Large Free Text Collection*, *Proc AMIA Symp*.

Disambiguation of Biomedical Abbreviations

Mark Stevenson[1], **Yikun Guo**[2], **Abdulaziz Al Amri**[3] and **Robert Gaizauskas**[4]
Department of Computer Science
University of Sheffield
Regent Court, 211 Portobello
Sheffield, S1 4DP
United Kingdom
[1,2,4]{initial.surname}@dcs.shef.ac.uk, [3]abdulazizmail@gmail.com

Abstract

Abbreviations are common in biomedical documents and many are ambiguous in the sense that they have several potential expansions. Identifying the correct expansion is necessary for language understanding and important for applications such as document retrieval. Identifying the correct expansion can be viewed as a Word Sense Disambiguation (WSD) problem. A WSD system that uses a variety of knowledge sources, including two types of information specific to the biomedical domain, is also described. This system was tested on a corpus of ambiguous abbreviations, created by automatically identifying the correct expansion in Medline abstracts, and found to identify the correct expansion with up to 99% accuracy.

1 Introduction

Many abbreviations are ambiguous in the sense that they have more than one possible expansion. For example, expansions for "NLP" include "Neuro-linguistic Programming" as well as "Natural Language Processing". Ambiguous abbreviations form a challenge to language understanding since identification of the correct expansion is often important. The query "NLP", for example, returns pages which refer to "Neuro-linguistic programming" for most web search engines, pages which are of limited value to those interested in Natural Language Processing. In some cases this problem could be obviated by altering the query terms, for example including "Natural", "Language" and "Processing". However, this will not help when the abbreviation's expansion does not occur within the document. Fred and Cheng (1999) point out that this is often the case in biomedical documents, in this domain ubiquitous abbreviations (such as DNA and mRNA) often appear without an expansion.

It has been reported that misinterpretation of abbreviations in biomedical documents has lead to medical practitioners making fatal errors (Fred and Cheng, 1999). However, identifying the correct expansion is not a straightforward task since an abbreviation may have several possible expansions. Chang et al. (2002) reported that abbreviations in biomedical journal articles consisting of six characters or less have an average of 4.61 possible meanings and Pustejovsky et al. (2002) mention that the simple abbreviation "AC" is associated with at least 10 strings in different biomedical documents including "atrioventricular connection", "anterior colporrhaphy procedure", "auditory cortex" and "atypical carcinoid".

The problem of identifying the correct expansion of an ambiguous abbreviation can be viewed as a Word Sense Disambiguation (WSD) task where the various expansions are the "senses" of the abbreviation. In this paper we approach the problem in this way by applying a WSD system which has previously been applied to biomedical text (Stevenson et al., 2008). The WSD system uses a variety of information sources, including those traditionally applied to the WSD problem in addition to two knowledge sources that are specific to the biomedical domain.

Evaluation of systems for disambiguating ambiguous abbreviations has been hindered by the fact

Proceedings of the Workshop on BioNLP, pages 71–79,
Boulder, Colorado, June 2009.

that there is no freely available benchmark corpus against which approaches can be compared. We describe a process whereby such a corpus can be created by automatically mining abstracts from Medline. This corpus is being made publicly available to encourage comparative research in this area. Our abbreviation disambiguation system was evaluated against this corpus and found to identify the correct abbreviation with up to 99% accuracy.

The remainder of this paper is organised as follows. The next section describes relevant previous work on disambiguation of abbreviations. Section 3 describes a supervised learning WSD system tailored specifically to the biomedical domain. Section 4 describes the automatic creation of a corpus of ambiguous abbreviations designed specifically for the training and evaluation of abbreviation disambiguation systems. Section 5 describes the evaluation of our system on this corpus. Our conclusions are presented in Section 6.

2 Previous Work

Gaudan et al. (2005) distinguish two types of abbreviations: global and local. *Global abbreviations* are those found in documents without the expansion explicitly stated, while *local abbreviations* are defined in the same document in which the abbreviation occurs. Our work is concerned with the problem of disambiguating global abbreviations. Gaudan et al. (2005) point out that global abbreviations are often ambiguous.

Various researchers have explored the problem of disambiguating global abbreviations in biomedical documents. Liu et al. (2001)(2002) used several domain-specific knowledge sources to identify terms which are semantically related to each possible expansion but which have only one sense themselves. Instances of these terms were identified in a corpus of biomedical journal abstracts and used as training data. Their learning algorithm uses a variety of features including all words in the abstract and collocations of the ambiguous abbreviation. They report an accuracy of 97% on a small set of abbreviations. Liu et al. (2004) present a fully supervised approach. They compared a variety of supervised machine learning algorithms and found that the best performance over a set of 15 ambiguous abbreviations, 98.6%, was obtained using Naive Bayes. Gaudan et al. (2005) use a Support Vector Machine trained on a bag-of-words model and report an accuracy of 98.5%. Yu et al. (2006) experimented with two supervised learning algorithms: Naive Bayes and Support Vector Machines. They extracted a corpus containing examples of 60 abbreviations from a set of biomedical journal articles which was split so that abstracts in which the abbreviations were defined were used as training data and those in which no definition is found as test data. Abbreviations in the test portion were manually disambiguated. They report 79% coverage and 80% precision using a Naive Bayes classifier. Pakhomov (2002) applied a maximum entropy model to identify the meanings of ambiguous abbreviations in 10,000 rheumatology notes with around 89% accuracy. Joshi et al. (2006) disambiguated abbreviations in clinical notes using three supervised learning algorithms (Naive Bayes, decision trees and Support Vector Machines). They used a range of features and found that the best performance was obtained when these were combined. Unfortunately direct comparison of these methods is made difficult by the fact that various researchers have evaluated their approaches on different data sets.

A variety of approaches have also been proposed for the problem of disambiguating local abbreviations in biomedical documents. This task is equivalent to identifying the abbreviation's expansion in the document. The problem is relatively straightforward for abbreviations which are created by selecting the first character from each word in the expansion, such as "angiotensin converting enzyme (ACE)", but is more difficult when this convention is not followed, for example "acetylchlinesterase (ACE)", "antisocial personality (ASP)" and "catalase (CAT)". Okazaki et al. (2008) recently proposed an approach to this problem based on discriminative alignment that has been shown to perform well. However, the most common solutions are based on heuristic approaches, for example Adar (2004) and Zhou et al. (2006). Pustejovsky et al. (2002) used hand-built regular expressions. Schwartz and Hearst (2003) describe an approach which starts by identifying the set of candidate expansions in the same sentence as an abbreviation. The most likely one is identified by searching for the

shortest candidate which contains all the characters in the abbreviation in the correct order.

3 Abbreviation Disambiguation System

Our abbreviation disambiguation system is based on a state-of-the-art WSD system that has been adapted to the biomedical domain by augmenting it with additional knowledge sources. The system on which our approach is based (Agirre and Martínez, 2004) participated in the Senseval-3 challenge (Mihalcea et al., 2004) with a performance close to the best system for the lexical sample tasks in two languages while the version adapted to the biomedical domain has achieved the best recorded results (Stevenson et al., 2008) on a standard test set consisting of ambiguous terms (Weeber et al., 2001).

This system is based on a supervised learning approach with features derived from text around the ambiguous word that are domain independent. We refer to these as *general* features. This feature set has been adapted for the disambiguation of biomedical text by adding further linguistic features and two different types of domain-specific features: CUIs (as used by McInnes et al. (2007)) and Medical Subject Heading (MeSH) terms. This set of features is more diverse than have been explored by previous approaches to abbreviation disambiguation.

3.1 Features

Our feature set contains a number of parameters (e.g. thresholds for unigram and CUI frequencies). These parameters were set to the same values that were used when the system was applied to general biomedical terms (Stevenson et al., 2008) since these were found to perform well. We also use the entire abstract as the context of the ambiguous term for relevant features rather than just the sentence containing the term. Effects of altering these variables are consistent with previous results (Liu et al., 2004; Joshi et al., 2005; McInnes et al., 2007) and are not reported here.

General features: The system uses a wide range of domain-independent features that are commonly employed for WSD.

- Local collocations: A total of 41 features which extensively describe the context of the ambiguous word and fall into two main types: (1) bigrams and trigrams containing the ambiguous word constructed from lemmas, word forms or PoS tags and (2) preceding/following lemma/word-form of the content words (adjective, adverb, noun and verb) in the same sentence as the ambiguous abbreviation. For example, consider the sentence below with the target abreviation *BSA*.

  > "Lean BSA was obtained from height and lean body weight ..."

  The features would include the following: left-content-word-lemma "*lean BSA*", right-function-word-lemma "*BSA be*", left-POS "JJ NNP", right-POS "NNP VBD", left-content-word-form "*Lean BSA*", right-function-word-form "*BSA was*", etc.

- Salient bigrams: Salient bigrams within the abstract with high log-likelihood scores, as described by Pedersen (2001).

- Unigrams: Lemmas of all content words in the abstract and words within a $\pm$4-word window around the target word, excluding those in a list of stopwords. In addition, the lemmas of any unigrams appearing at least twice in the entire corpus and which are found in the abstract are also included as features.

Concept Unique Identifiers (CUIs): We follow the approach presented by McInnes et al. (2007) to generate features based on UMLS Concept Unique Identifiers (CUIs). The MetaMap program (Aronson, 2001) identifies all words and terms in a text which could be mapped onto a UMLS CUI. MetaMap does not disambiguate the senses of the concepts, instead it enumerates likely candidate concepts. For example, MetaMap will segment the phrase "Lean BSA was obtained from height and lean body weight ..." into four chunks: "Lean BSA", "obtained", "from height" and "lean body weight". The first chunk will be mapped onto three CUIs: "C1261466: BSA (Body surface area)", "C1511233: BSA (NCI Board of Scientific Advisors)" and "C0036774: BSA (Serum Albumin, Bovine)". The chunk "lean body weight" is mapped onto two concepts: "C0005910: Body Weight"

and "C1305866: Body Weight (Weighing patient)"[1]. CUIs occurring more than twice in an abstract are included as features. CUIs have been used for various disambiguation tasks in the biomedical domain, including disambiguation of ambiguous general terms (McInnes et al., 2007) and gene symbol disambiguation (Xu et al., 2007), but not, to our knowledge, for abbreviation disambiguation.

Medical Subject Headings (MeSH): The final feature is also specific to the biomedical domain. Medical Subject Headings (MeSH) (Nelson et al., 2002) is a controlled vocabulary for indexing biomedical and health-related information and documents. MeSH terms are manually assigned to abstracts by human indexers. The latest version of MeSH (2009) contains over 25,000 terms organised into an 11 level hierarchy.

The MeSH terms assigned to the abstract in which each ambiguous word occurs are used as features. For example, the abstract containing our example phrase has been assigned 16 terms including "Body Surface Area", "Body Weight", "Humans" and "Organ Size" . MeSH terms have previously been used for abbreviation disambiguation by Yu et al. (2006).

3.2 Learning Algorithms

We compared three machine leaning algorithms which have previously been shown to be effective for WSD tasks.

The **Vector Space Model (VSM)** is a memory-based learning algorithm which was used by Agirre and Martínez (2004). Each occurrence of an ambiguous word is represented as a binary vector in which each position indicates the occurrence/absence of a feature. A single centroid vector is generated for each sense during training. These centroids are compared with the vectors that represent new examples using the cosine metric to compute similarity. The sense assigned to a new example is that of the closest centroid.

The **Naive Bayes (NB)** classifier is based on a probabilistic model which assumes conditional independence of features given the target classification. It calculates the posterior probability that an instance belongs to a particular class given the prior probabilities of the class and the conditional probability of each feature given the target class.

Support Vector Machines (SVM) have been widely used in classification tasks. SVMs map feature vectors onto a high dimensional space and construct a classifier by searching for the hyperplane that gives the greatest separation between the classes.

We used our own implementation of the Vector Space Model and Weka implementations (Witten and Frank, 2005) of the other two algorithms.

4 Evaluation Corpus

The most common method for generating corpora to train and test WSD systems is to manually annotate instances of ambiguous terms found in text with the appropriate meaning. However, this process is both time-consuming and difficult (Artstein and Poesio, 2008). An alternative to manual tagging is to find a way of automatically creating sense tagged corpora. For the translation of ambiguous English words Ng et al. (2003) made use of the fact that the various senses are often translated differently. For example when "bank" is used in the 'financial institution' sense it is translated to French as "banque" and "bord" when it is used to mean 'edge of river'. However, a disadvantage of this approach is that it relies on the existence of parallel text which may not be available. In the biomedical domain Liu et al. (2001)(2002) created a corpus using unambiguous related terms (see Section 2) although they found that it was not always possible to identify suitable related terms.

4.1 Corpus Creation

Liu et al. (2001) also made use of the fact that when abbreviations are introduced they are often accompanied by their expansion, for example "BSA (bovine serum albumin)". This phenomenon was exploited to automatically generate a corpus of abbreviations and associated definitions by replacing the abbreviation and expansion with the abbreviation alone. For example, the sentence *"The adsorption behavior of bovine serum albumin (BSA) on a Sepharose based hydrophobic interaction support has been studied."* becomes *"The adsorption behav-*

[1]The first of these, C0005910, refers to the weight of a patient as a property of that individual while the second, C1305866, refers to the process of weighing a patient as part of a diagnostic procedure.

"BSA" AND "body surface area" NOT "bovine serum albumin"
"BSA" AND "bovine serum albumin" NOT "body surface area"

Figure 1: Example queries for abbreviation "BSA"

ior of BSA on a Sepharose based hydrophobic interaction support has been studied."

We used this approach to create a corpus of sense tagged abbreviations in biomedical documents using a set of 21 three letter abbreviations used in previous research on abbreviation disambiguation (Liu et al., 2001; Liu et al., 2002; Liu et al., 2004). Possible expansions for the majority of these abbreviations were listed in these papers. For the few remaining ones possible expansions were taken from the Medstract database (Pustejovsky et al., 2002). We searched for instances of these abbreviations in Medline, a database containing more than 18 million abstracts from publications in biomedicine and the life sciences. For each abbreviation we queried Medline, using the Entrez interface, to identify documents containing one of its meanings. For example the abbreviation "BSA" has two possible expansions: "body surface area" and "bovine serum alumin". Medline is searched to identify documents that contain each possible expansion of the abbreviation using the queries shown in Figure 1. Each query matches documents containing the abbreviation and relevant expansion and no mentions of the other possible expansion(s).

The retrieved documents are then processed to remove the expansions of each abbreviation. The Schwartz and Hearst (2003) algorithm for identifying abbreviations and the relevant expansion (see Section 2) is then run over each of the retrieved abstracts to identify the correct expansion. The expansion is removed from the document and stored separately, effectively creating a sense tagged corpus. For convenience the abstracts are converted into a format similar to the one used for the NLM-WSD corpus (Weeber et al., 2001).

The resulting corpus consists of 55,655 documents. For each abbreviation Table 1 shows the number of abstracts retrieved from Medline (in the column labeled "Abstracts") and the number of expansions ("Count" column). The column labelled "Rare" lists the number of expansions that account for fewer than 1% of the occurrences of an abbreviation and "Frequent" lists the percentage of occurances represented by the most frequent expansion. It can be seen that there is a wide variation between the number of abstracts retrieved for each abbreviation. CSF occurs in 14,871 abstracts and ASP in just 71. There is also a wide variation between the frequency of the most common expansion with over 99% of the occurrences of "CSF" representing one expansion ("cerebrospinal fluid") while for "ASP" two of the five possible expansions ("antisocial personality" and "aspartate") each account for almost 34% of the documents. In addition, several abbreviations have expansions which occur only rarely. For example, two of the expansions of "APC" ("atrial pressure complexes" and "aphidicholin") each have only a single document and account for just 0.03% of the instances of that abbreviation.

4.2 Corpus Reduction

Given the diversity of the abbreviations which were downloaded from Medline, both in terms of number of documents and distribution of senses, subsets of this corpus that are more suitable for WSD experiments were created. Corpora containing 100, 200 and 300 randomly selected examples of each abbreviation were generated and these are referred to as Corpus.100, Corpus.200 and Corpus.300 respectively.

Some of the 21 abbreviations were not suitable for inclusion in these corpora. Abbreviations were not included in the relevant corpus if an insufficient number of examples were retrieved from Medline. For example, only 71 abstracts containing "ASP" were retrieved and it is is not included in any of the three corpora. Similarly, "ANA" and "FDP" are not included in Corpus.200 or Corpus.300 and "DIP" not included in Corpus.300. In addition, rare senses, those which represent fewer than 1% of the occurrences of an abbreviation in all retrieved abstracts, were discarded. Finally, two abbreviations ("ACE" and "CSF") have only one sense that is not "Rare"

		Expansions		
	Abstracts	Count	Rare	Frequent
ACE	3105	3	2	98.7
ANA	100	3	0	58.0
APC	3146	5	2	39.4
ASP	71	5	0	33.8
BPD	1841	3	0	46.7
BSA	5373	2	0	86.4
CAT	4636	3	1	55.2
CML	2234	4	2	91.7
CMV	7665	2	0	96.7
CSF	14871	3	2	99.1
DIP	209	2	0	75.1
EMG	2052	2	0	88.4
FDP	130	4	0	78.5
LAM	325	4	1	48.3
MAC	955	5	1	64.3
MCP	815	5	1	50.2
PCA	2442	5	1	68.9
PCP	1642	2	0	57.8
PEG	607	2	0	94.1
PVC	234	2	2	78.2
RSV	3202	2	0	76.7
Average	2650	3.2	0.6	70.8

Table 1: Properties of abbreviations corpus retrieved from Medline

(see Table 1) and these were also excluded from the reduced corpora.

Consequently, Corpus.100 contains 18 abbreviations ("ACE", "ASP" and "CSF" are excluded), Corpus.200 contains 16 ("ANA" and "FDP" are also excluded) and Corpus.300 contains 14 ("DIP" and "PVC" also excluded). Where an abbreviation is included in more than one corpus, all the examples in the smaller corpus are included in the larger one(s). For example, the 100 examples of "APC" in Corpus.100 are also included in Corpus.200 and Corpus.300.

5 Experiments

Various combinations of learning algorithms and features were applied to the three reduced corpora described in Section 4.2. Performance of the WSD system is measured in terms of the proportion of abbreviation instances for which the correct expansion is identified. 10-fold cross validation was used for all experiments and all quoted results refer to the average performance across the 10 folds. Results are shown in Table 2. The baseline figures, based on selecting the most frequent expansion for each abbreviation, are shown for each corpus. Note that these figures vary slightly across the three corpora because of the different abbreviations each contains (see Section 4.2).

A first observation is that performance of the WSD system is consistently better than the baseline for the relevant corpus and, with a few exceptions, above 90%. As might be expected, performance improves as additional training examples are added. However, even when the number of examples is relatively low, just 100, performance of the best configuration (VSM learning algorithm with all three types of feature) is 97.4%.

The best result, 99% (300 training examples, VSM learning algorithm with all feature types), exceeds reported performance of previous abbreviation disambiguation systems (see Section 2). Although these results are not directly comparable, since these studies used different evaluation corpora, the set of ambiguous abbreviations used in this study and methodology for corpus creation are similar to those used by Liu et al. (2001)(2002)(2004).

The best performance for each learning algorithm is obtained when all three types of features are combined. The difference between performance obtained using all three feature types and using only the MeSH or CUI features is statistically significant (Wilcoxon Signed Ranks test, $p < 0.01$) although the difference between this and performance using just the linguistic features is not.

The VSM learning algorithm generally performs better than either the SVM or Naive Bayes learning algorithms. The difference between performance of VSM and the other algorithms is statistically significant for Corpus.100 but not for the other two, suggesting that this learning algorithm is better able to cope with small number of training examples than Naive Bayes and Support Vector Machines. Strong performance of the VSM algorithm is consistent with previous work which has shown that this algorithm performs well on the disambiguation of ambiguous terms in both biomedical and general text (Agirre and Martínez, 2004; Stevenson et al., 2008).

Algorithm	Features						
	Linguistic	CUI	MeSH	Linguistic +CUI	Linguistic +MeSH	CUI+ MeSH	Linguistic+ MeSH+CUI
	Corpus.100 (Baseline = 69.0%)						
SVM	0.934	0.900	0.949	0.947	0.946	0.938	0.954
NB	0.940	0.917	0.949	0.951	0.947	0.944	0.958
VSM	0.968	0.937	0.888	0.970	0.971	0.939	**0.974**
	Corpus.200 (Baseline = 69.1%)						
SVM	0.957	0.911	0.964	0.964	0.964	0.947	0.965
NB	0.966	0.926	0.962	0.969	0.971	0.955	0.972
VSM	0.979	0.930	0.894	0.982	0.981	0.947	**0.984**
	Corpus.300 (Baseline = 68.7%)						
SVM	0.966	0.914	0.970	0.968	0.974	0.954	0.975
NB	0.971	0.933	0.960	0.971	0.976	0.960	0.978
VSM	0.981	0.938	0.894	0.987	0.985	0.957	**0.990**

Table 2: Performance of WSD system using various combinations of learning algorithms and features.

Performance of our system on this task is higher than would be expected for most WSD tasks suggesting that the problem of abbreviation disambiguation is simpler than the disambiguation of general terms. The most probable reason for this is that the various expansions of abbreviations in our corpus are more distinct and better defined than senses for general terms. For example, the three possible expansions for "ANA" in our corpus are a professional body ("American Nurses Association"), a type of medical test ("antinuclear") and a neurotransmitter ("Anandamide"). It is likely that these diverse meanings will tend to occur in very different contexts and in documents with different topics. On the other hand it is widely accepted that distinctions between possible meanings of words in natural language are often vague (Kilgarriff, 1993). It is likely that clearer distinctions between possible expansions of abbreviations make the task of identifying the correct one more straightforward than identifying meanings of ambiguous words. In addition, the creation of annotated data for WSD is often hampered by the difficulty in obtaining sufficient agreement between annotators (Artstein and Poesio, 2008; Weeber et al., 2001) and this problem does not apply to our automatically-generated corpus.

Results in Table 2 indicate that CUIs are useful features in the disambiguation of abbreviations. This is in contrast with previous experiments on ambiguous terms in biomedical documents (Stevenson et al., 2008) in which it was found that the best performance as obtained using only linguistic and MeSH features. It is likely that the clear distinction between expansions of abbreviations is the reason behind this difference. CUIs are assigned automatically by the MetaMap program (Aronson, 2001). However, this assignment is very noisy. It is likely that the various expansions of abbreviations are distinct enough for this noise to be tolerated by the learning algorithms while it causes problems when the meanings are closer together, such as in the case of ambiguous terms.

5.1 Performance of Individual Abbreviations

Table 3 shows the performance of the best WSD system (VSM learning algorithm with all features) for each abbreviation in the three subsets of our corpus. Our system performs well for all abbreviations. Accuracy is no lower than 92% for any abbreviation using Corpus.100 and no lower than 97% for Corpus.300, demonstrating that the approach is robust. In fact, the approach still performs well for abbreviations with low baseline scores, such as "APC", "BPD" and "LAM".

It is interesting to note that the abbreviations with the lowest performance tend to have expansions that are closely related. For example, the two expansions of "EMG" are 'electromyography' and 'electromyo-

	Corpus		
	100	200	300
ANA	0.980	-	-
APC	0.980	1.000	1.000
BPD	1.000	1.000	1.000
BSA	0.970	0.970	0.982
CAT	0.990	0.990	1.000
CML	0.960	0.963	0.978
CMV	0.970	0.970	0.970
DIP	1.000	1.000	-
EMG	0.920	0.960	0.980
FDP	0.970	-	-
LAM	0.960	0.980	0.980
MAC	0.970	0.990	0.989
MCP	0.980	0.978	1.000
PCA	0.960	0.987	0.992
PCP	0.990	1.000	1.000
PEG	0.980	0.982	1.000
PVC	0.990	1.000	-
RSV	0.960	0.972	0.978
Overall	0.974	0.984	0.990

Table 3: Performance of WSD system over individual abbreviations in three reduced corpora

gram' while for "LAM" one expansion ('Lymphangioleiomyomatosis') is a rare lung disease and the other ('Lipoarabinomannan') a molecule associated with another lung disease (tuberculosis). On the other hand, abbreviations that are more accurately disambiguated tend to have expansions with more distinct meanings. For example, "BPD" can be an acronym for 'borderline personality disorder' (a psychiatric diagnosis), 'bronchopulmonary dysplasia' (a lung disease) or 'biparietal diameter' (diameter of a foetus' head in an ultrasound) and the expansions of "DIP" are 'desquamative interstitial pneumonia' (a lung disease) and 'distal interphalangeal joints' (types of joints in the human hand and foot).

6 Conclusions

This paper has presented an approach to the disambiguation of ambiguous abbreviations in biomedical documents. We treat this problem as a form of WSD and apply a system that combines a wider range of features than have been previously applied, including those which are commonly used within WSD systems in addition to information from two domain-specific knowledge sources. The approach is evaluated using a corpus of abbreviations automatically mined from Medline and found to identify the correct expansion with accuracy of up to 99%. This figure is higher than previously reported results for abbreviation disambiguation systems, although direct comparison is difficult due to the use of different data sets. It was also found that best performance could be obtained using a simple machine learning algorithm and a diverse range of knowledge sources. Performance of our system is higher than is normally achieved by WSD systems when applied to general terms and we suggest that the reason for this is that the various expansions of abbreviations are better defined and more distinct than the senses of ambiguous words.

This study has been limited to the disambiguation of abbreviations consisting of exactly three letters. Possibilities for future work include experimenting with abbreviations of various lengths.

Data

The corpus described in Section 4 has been made freely available for research and may be obtained from `http://nlp.shef.ac.uk/BioWSD/downloads/abbreviationdata/`.

Acknowledgments

We are grateful to the anonymous reviewers of this paper for their valuable feedback.

References

E. Adar. 2004. SaRAD: A simple and robust abbreviation dictionary. *Bioinformatics*, 20(4):527–533.

E. Agirre and D. Martínez. 2004. The Basque Country University system: English and Basque tasks. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 44–48, Barcelona, Spain, July.

A. Aronson. 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In *Proceedings of the American Medical Informatics Association (AMIA)*, pages 17–21.

R. Artstein and M. Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

J. Chang, H. Schütze, and R. Altman. 2002. Creating an Online Dictionary of Abbreviations from MEDLINE. *The Journal of the American Medical Informatics Association*, 9(6):612–620.

H. Fred and T. Cheng. 1999. Acronymesis: the exploding misuse of acronyms. *Texas Heart Institute Journal*, 30:255–257.

S. Gaudan, H. Kirsch, and D. Rebholz-Schuhmann. 2005. Resolving abbreviations to their senses in Medline. *Bioinformatics*, 21(18):3658–3664.

M. Joshi, T. Pedersen, and R. Maclin. 2005. A Comparative Study of Support Vector Machines Applied to the Word Sense Disambiguation Problem for the Medical Domain. In *Proceedings of the Second Indian Conference on Artificial Intelligence (IICAI-05)*, pages 3449–3468, Pune, India.

M. Joshi, S. Pakhomov, T. Pedersen, and C. Chute. 2006. A comparative study of supervised learning as applied to acronym expansion in clinical reports. In *Proceedings of the Annual Symposium of the American Medical Informatics Association*, pages 399–403, Washington, DC.

A. Kilgarriff. 1993. Dictionary word sense distinctions: An enquiry into their nature. *Computers and the Humanities*, 26:356–387.

H. Liu, Y. Lussier, and C. Friedman. 2001. Disambiguating ambiguous biomedical terms in biomedical narrative text: An unsupervised method. *Journal of Biomedical Informatics*, 34:249–261.

H. Liu, S. Johnson, and C. Friedman. 2002. Automatic Resolution of Ambiguous Terms Based on Machine Learning and Conceptual Relations in the UMLS. *Journal of the American Medical Informatics Association*, 9(6):621–636.

H. Liu, V. Teller, and C. Friedman. 2004. A Multi-aspect Comparison Study of Supervised Word Sense Disambiguation. *Journal of the American Medical Informatics Association*, 11(4):320–331.

B. McInnes, T. Pedersen, and J. Carlis. 2007. Using UMLS Concept Unique Identifiers (CUIs) for Word Sense Disambiguation in the Biomedical Domain. In *Proceedings of the Annual Symposium of the American Medical Informatics Association*, pages 533–537, Chicago, IL.

R. Mihalcea, T. Chklovski, and A. Kilgarriff. 2004. The Senseval-3 English lexical sample task. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain.

S. Nelson, T. Powell, and B. Humphreys. 2002. The Unified Medical Language System (UMLS) Project. In Allen Kent and Carolyn M. Hall, editors, *Encyclopedia of Library and Information Science*. Marcel Dekker, Inc.

H. Ng, B. Wang, and S. Chan. 2003. Exploiting Parallel Texts for Word Sense Disambiguation: an Empirical Study. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, pages 455–462, Sapporo, Japan.

N. Okazaki, S. Ananiadou, and J. Tsujii. 2008. A discriminative alignment model for abbreviation recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 657–664, Manchester, UK.

S. Pakhomov. 2002. Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical texts. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Philadelphia, PA.

T. Pedersen. 2001. A Decision Tree of Bigrams is an Accurate Predictor of Word Sense. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-01)*, pages 79–86, Pittsburgh, PA.

J. Pustejovsky, J. Castano, R. Saur, A. Rumshisky, J. Zhang, and W. Luo. 2002. Medstract: Creating Large-scale Information Servers for Biomedical Libraries. In *ACL 2002 Workshop on Natural Language Processing in the Biomedical Domain*.

A. Schwartz and M. Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Proceedings of the Pacific Symposium on Biocomputing*, Kauai.

M. Stevenson, Y. Guo, R. Gaizauskas, and D. Martinez. 2008. Disambiguation of biomedical text using diverse sources of information. *BMC Bioinformatics*, 9(Suppl 11):S7.

M. Weeber, J. Mork, and A. Aronson. 2001. Developing a Test Collection for Biomedical Word Sense Disambiguation. In *Proceedings of AMAI Symposium*, pages 746–50, Washington, DC.

I. Witten and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco.

H. Xu, J. Fan, G. Hripcsak, E. Mendonça, Markatou M., and Friedman C. 2007. Gene symbol disambiguation using knowledge-based profiles. *Bioinformatics*, 23(8):1015–22.

H. Yu, W. Kim, V. Hatzivassiloglou, and J. Wilbur. 2006. A large scale, corpus-based approach for automatically disambigutaing biomedical abbreviations. *ACM Transactions on Information Systems*, 24(3):380–404.

W. Zhou, I. Vetle, and N. Smalheiser. 2006. ADAM: another database of abbreviations in MEDLINE. *Bioinformatics*, 22(22):2813–2818.

TX Task:
Automatic Detection of Focus Organisms in Biomedical Publications

Thomas Kappeler, Kaarel Kaljurand, Fabio Rinaldi*
Institute of Computational Linguistics, University of Zurich
kappeler@bluewin.ch, kalju@cl.uzh.ch, rinaldi@cl.uzh.ch

Abstract

In biomedical information extraction (IE), a central problem is the disambiguation of ambiguous names for domain specific entities, such as proteins, genes, etc. One important dimension of ambiguity is the organism to which the entities belong: in order to disambiguate an ambiguous entity name (e.g. a protein), it is often necessary to identify the specific organism to which it refers.

In this paper we present an approach to the detection and disambiguation of the focus organism(s), i.e. the organism(s) which are the subject of the research described in scientific papers, which can then be used for the disambiguation of other entities.

The results are evaluated against a gold standard derived from IntAct annotations. The evaluation suggests that the results may already be useful within a curation environment and are certainly a baseline for more complex approaches.

1 Introduction

The task of identifying the organisms which are involved in research described in biomedical articles is extremely important for the field of biomedical information extraction (IE), both in itself and in connection with other tasks. In itself, because the concept of biological taxonomy is basic for every researcher: organisms and their taxonomic classification can be used very effectively in various contexts, for example to restrict searches, a classical information retrieval (IR) task. At the same time, any biomedical text mining system would be incomplete without the possibility to use organisms as concepts, e.g. in finding (statistical) associations, which can then be used to form hypotheses about causal relations.

The necessity of identifying organisms is even more evident as part of other important entity recognition tasks in biomedical information extraction (IE), e.g. identification and disambiguation of proteins mentioned in the literature. For example, within the PPI task (identification of protein-protein interactions) of Biocreative II (Krallinger et al., 2008), the identification of the focus organism was seen by many participants as an essential subtask in order to properly disambiguate protein names. Protein interactions are fundamental for most biological processes, therefore they are at the focus of a huge and fast growing number of biomedical papers. As these cannot all be read or even inspected by the researchers, databases such as IntAct (Kerrien et al., 2006) or MINT (Zanzoni et al., 2002) try to create a reliable catalogue of experimentally detected interactions by extracting them "manually" from the literature through the usage of human experts. This is known as "curation", a costly and time-consuming process, which could be speeded up much by efficient, robust and precise extraction tools.

One of the most important obstacles for efficient automatic identification of proteins is the extreme ambiguity of the commonly used protein names in the literature. The fragmentation of the biomedical scientific community into lots of extremely specialized sub-communities seems to be the main reason for this ambiguity. In most cases, the ambiguity is between homologous proteins of different species. Any human reader belonging to the sub-community concerned can, in general, disambiguate an ambiguous protein name like "goat" (which can refer to proteins found in four different organisms: human, rat, mouse and zebrafish), as the species is obvious to them from the context. However, this ambiguity

*Corresponding author

Proceedings of the Workshop on BioNLP, pages 80–88,
Boulder, Colorado, June 2009.

remains problematic for IE systems (and even for curators in some cases) and needs to be solved before more complex tasks, such as protein interaction detection, can be effectively tackled (Rinaldi et al., 2008).

Our goal is to be able to identify automatically the focus organisms, i.e. the organisms that are mentioned in the paper as the hosts of the experiments described, or as the sources of the entities involved. This information can then be used for tagging papers for more efficient organism-based information retrieval, or, more commonly, for the disambiguation of other entities mentioned in the same paper. Since organism recognition is normally performed with reference to a taxonomical organization (of Linnean origin) of all known organisms (in our case, the NCBI taxonomy) this task is often referred to as "TX task".

In the rest of this paper we describe in section 2 the resources used and the approach followed in order to extract and rank candidate organisms. In section 3 we present our results and propose a more fine grained interpretation of the task, which we again evaluate. Finally in section 4 we compare our approach to previous work and discuss its limitations.

2 Methods

Our approach can be described briefly as (1) find all explicit mentions of organisms either by their scientific or "common" names; (2) count these mentions and combine the resulting numbers with a simple use of statistics to arrive at a ranked list or a simple set of organisms which can be used, among other things, to disambiguate ambiguous protein names in the article under investigation.

2.1 Resources Used

The first step for this approach was to choose a widely accepted taxonomy which not just includes unambiguous identifiers for all known organisms, but also provides a sufficiently large list of names for them. The taxonomy selected for this was the NCBI Taxonomy[1].

As most of these organism are unlikely to ever occur in biomedical literature, we decided to restrict our interest to the organisms for which a UniProt organism mnemonic identifier exists. UniProt (UniProt Consortium, 2007) is a database containing detailed information about known proteins, obtained by a process of curation of the biomedical literature. For every protein, a "mnemonic" identifier is defined (e.g. HBA_HUMAN for "Human Hemoglobin A") which is composed by a shorthand for the protein name and a simple unique identifier for the organism. Within the UniProt entry for the protein, the organism is also referred to by its NCBI identifier, allowing the construction of a mapping from the mnemonic identifiers for the organisms used by UniProt to their equivalent NCBI identifiers.

The set of organism that have a UniProt mnemonic identifier (11,444 organisms) probably covers the near totality of organisms that have been subject to research in molecular biology. In the NCBI taxonomy 31,733 names are defined for that subset of organisms. Although several classes of names are defined by NCBI, for the purpose of this work we distinguish only between "scientific names" and the other classes (pooled together as "common names").[2]

As an additional source of information, we used the IntAct database of protein interactions[3] for two different purposes:

- to derive statistical measures used later by the program, most importantly the frequency of each focus organism in papers curated by IntAct (using the IntAct annotations as the sources of the 'focus').
- to derive a gold standard against which our programs could be tested

IntAct provides an annotated set of protein interactions. Each interaction is enriched with detailed information about the two proteins involved

[1] Available as archive taxdmp.zip from ftp://ftp.ncbi.nih.gov/pub/taxonomy/. We worked with a version downloaded on July 10th 2008. The file nodes.dmp contains the taxonomy as a set of 443,299 nodes for the taxa and immediate-dominance-relations between them. The file names.dmp connects one or several names (619,325) of different nameclasses (such as "scientific" or "common") to each node. The nodes (taxa) are referred to by numeric identifiers.

[2] While there are no ambiguous "scientific names" in this taxonomy, there are several ambiguous "common names", but only very few of these occurred in our sample, e.g. "mink", "barley", "green monkey", and they are very rare.

[3] Version of May 2008, downloaded from http://www.ebi.ac.uk/intact/site/contents/downloads.jsf

(from which the reference organisms can be recovered), and with the identifier of the paper from which the interaction was originally derived in the curation process. This allows to build a gold standard by associating each paper to its focus organisms.

The sample used in our experiments is a set of 621 PubMed-indexed full text articles, dating from 1995 to 2007, for which IntAct annotations are available.[4]

2.2 First Experiments and Normalization

As an initial experiment, we performed a simple lexical lookup of the names of the 11,444 organisms under consideration. In previous applications of IE techniques for biomedical literature (Kappeler et al., 2008; Rinaldi et al., 2008) we found that simple techniques for the generation of variants of the known names significantly benefited the recall of the application. For example, multiword protein names can be subject to a number of minor variants, such as the introduction of hyphens or the separation of compound words, which make automatic recognition more challenging. In the case of organism names, although our initial expectations were similar, we found the benefit (in terms of additional recall) of such variants to be extremely limited, possibly because names of species are used more consistently than the names of proteins or genes.

Therefore it was possible to implement a simpler approach to recognition of organism names, based on lexical lookup against a database containing all names of interest, coupled with a simple normalization step which removes trivial orthographic differences (such as hyphens) between the key word in the database and the lookup word from the document (for details see (Kaljurand et al., 2009)). The inclusion of other biomedical NE's (such as protein names, method names, cell line names) in the database together with a strict implementation of the "longest match" principle leads to better precision by eliminating false positives caused by matching organism names with a fragment of a multiword term for another entity (such as the method "yeast two-hybrid").

As mentioned, the names provided by the NCBI taxonomy have been classified into "scientific names" or "common names". Using only "scientific names" appeared as an effective way to obtain better precision, but we soon discovered that precision of the common names suffered most by a few very bad names, such as "Li", which is a "common name" for LIV (Louping ill virus) in the taxonomy, but appears only (and very frequently) as Chinese surname in the texts. By eliminating about 25 of similar misleading "common names" the results of this class rose to the same level as the "scientific names", so there was no reason to exclude the whole class (as that would have harmed recall).

Since the bibliography might contain spurious mentions of other organisms, we automatically removed it from the main text. However, contrary to expectations, this did not lead to better results for this task (at least after the elimination of the misleading "common names" mentioned above), but was not reversed because of its effects on other tasks. An intuition from other tasks was to use the abstracts instead of the full text of the articles, because that would tend to exclude accidental mentions of organisms leading to false positives. But a main problem of this approach is that many abstracts do not yield any organism mentions. Whenever they do though, their precision is high. So there is a strong case for giving the mentions there a higher weight, but obviously the rest of the article plays an important role as well. We experimentally found that counting an "abstract mention" as equivalent to 25 "fulltext mentions" worked best.

2.3 Measures Improving Recall

An experiment using all names provided by NCBI and considering all mentions of those names in the fulltext version of each article led to a recall of 83%, leading us to conclude that either the taxonomy does not contain all names used, or some organisms are suggested to the human reader by the context and/or his anticipations. The first of these problems was adressed by adding some generated names to the termbase, the second by the use of a default.

Several possible ways of generating new names automatically from the names in the database were considered, but only two were applied successfully, as described below. One of them was the automatic generation of additional names from the nameclass

[4]The reason of this particular choice is that the same subset was used for experiments related to the automatic detection of experimental methods, also using IntAct annotations as a gold standard, described in (Kappeler et al., 2008).

"scientific name" (for organisms of species or subspecies level) by the process of replacing the first word (which would be the genus name in the classical Linnean binomial nomenclature) by its first letter and a dot. The resulting names, such as "E. coli", are widely used, but not included in the taxonomy. A seemingly large disadvantage of this approach is its potential for ambiguity: 338 of the resulting names refer to more than one organism. But the test on our sample showed that of these only 4 occurred at all, only 1 more than once: "C. elegans" (potentially referring to the organisms identified in UniProt as CAEEL, CENEL, CESEL and CUNEL) which always stood for CAEEL, i.e. "Caenorhabditis elegans". So excluding the other options for "C. elegans" eliminated the ambiguity (at least in our sample). We observed that this type of name is in frequent use only for few species and in this case the unabbreviated name is often used first, so the addition of this generated nameclass added little to recall.

The other type of name missing from the taxonomy is the use of the (Linnean) genus name for a very frequent species, e.g. "Arabidopsis" used for "Arabidopsis thaliana". Experiments showed that this type could not be reliably generated automatically from the "scientific names", as this nameclass includes many names which do not follow the rules of Linnean binomial nomenclature, mostly virus names such as "Human papillomavirus type me180" where the first word is generally not a genus name, but a host name. So the problem of (potentially huge) ambiguity in this type of names was not even researched, instead the names of this type for the most frequent organisms were generated manually and those which improved the results were included into the termbase (*Saccharomyces, Arabidopsis, Drosophila, Escherichia, Xenopus* and *Synechocystis*). The addition of this generated nameclass did not add much to recall for the same reason as for the first group: in most cases the unabbreviated name appears in the paper as well. Together both groups improved recall by about 3.4%.

As HUMAN is the most frequent organism in this context, it was obvious that a default HUMAN would take care of many cases where human readers disambiguate ambiguous protein names even without any explicit mentions of this species. As there are no cases (with the current termbase and sample) of articles with no organism mentions in the full text, we chose to have a default triggered by no findings in the abstract. Experiments showed that — contrary to intuition — a weight of the default proportional to the total number of mentions (just adding a percentage to HUMAN) would lead to worse results than an absolute value for the default.[5]

Table 1: Most frequent organisms in IntAct (derived from interactor proteins and host organisms)

ORG	freq
HUMAN	0.281
YEAST	0.272
MOUSE	0.091
ARATH	0.056
CERAE	0.037
RAT	0.033
DROME	0.028
SCHPO	0.023
ECOLX	0.020
ECOLI	0.013

2.4 Measures Improving Precision

The simple approach of considering every mention of each organism (after excluding the misleading common names, as described above), leads to a precision of only 27.6%, therefore the list of organism identifiers obtained in this way has to be considered as a "candidates list" from which a selection has to be made.

Candidates can be of course ranked according to number of mentions in each article. A ranking based on the mention counts, taking into account the correction factor of 25 for mentions in the abstract (as described in section 2.2), was still far from optimal, so we multiplied the mentions with the relative frequencies of the organisms in a micro-averaged frequency table (table 1) computed over all of IntAct (not just our sample, to avoid overfitting) and smoothed roughly by attributing 1% of the probability mass to all unseen organisms (over 11,000). This ranking did far better than expected and after nor-

[5] A tentative explanation: In a small paper, the effect of accidental mentions of "wrong" organisms is much larger than in big papers (where the important organisms are mentioned again and again). This detrimental effect may be counterbalanced by a relatively stronger default.

malizing the whole list to 1, a minimal threshold for the score could be set up to maximize the f-score by improving precision at the cost of recall. The actual value of the threshold (currently 0.04) is of course arbitrary, depending on what measure one wants to maximize.

Another problem to be tackled is that different papers will have different numbers of focus organisms, ranging from one (in about 70% of the cases), to several hundreds (in a few very infrequent cases). It could be assumed that being able to correctly guess the number of focus organisms would lead to improvement in the TX task, as we could pick only as many candidate organisms (in their ranking order) as the expected number for the paper. However, an experiment using the gold standard as an oracle to predict the number of organisms to be returned as a result, instead of using a threshold in the ranking, did not perform much better (recall was about 1.7% higher), so we decided not to spend any energy on exploring ways to predict the number of organisms as the effect would be minimal, even with perfect prediction.

Further experiments, such as giving different weights to mentions of names of different name-classes, did not lead to better results. Including information about the precision or recall of the names encountered in our test set (or the organisms predicted by them) in the formula for the weights[6] did not lead to better results either.

3 Evaluation and analysis of results

So finally the program in its current form considers all organism mentions, as delivered by the termbase search, eliminates the problematic common names, counts the mentions for each organism in fulltext and abstracts, multiplies the latter by 25 and adds them to the fulltext mentions. In case of no abstract mentions, a default of 28 fulltext mentions is added to HUMAN (equivalent to about one abstract mention).

The result for each organism is multiplied by the relative frequency of the organism in IntAct and divided by the sum of the results over all organisms to normalize the sum of the values to 1 (100%). All organisms under the threshold of 0.04 (or 4%) are then eliminated from the list.

[6] An idea suggested by its successful use in the detection of experimental methods in (Kappeler et al., 2008) and (Rinaldi et al., 2008).

Table 2: Most frequent false positives for the best results with our sample

ORG	freq
HUMAN	121
YEAST	104
MOUSE	68
ECOLX	18
DROME	13
ARATH	11
RAT	9

Table 3: Most frequent false negatives for the best results with our sample

ORG	freq
CERAE	73
MOUSE	59
RAT	40
YEAST	21
BOVIN	14
ECOLI	13
ECOLX	13

Our best results (max. f-score) for the task of finding all organisms in the gold standard combining organisms of interacting proteins and host organisms are: precision: 0.742; recall: 0.738; f-score: 0.740.

An analysis of the most frequent false positives is reported in table 2. The ranking is more or less identical with the frequency table (table 1), which is what we would expect. Manual inspection of some of the papers causing these false positives gave the following results:

- Some names of experimental methods containing organism names (which could avoid false positives if recognized as methods) were not yet included in the termbase.
- Some organisms (or their proteins respectively) are discussed in the paper, but not as results of the authors own experiments, so they do not appear in the gold standard. Obviously the curators consider only the novel findings reported in the paper, and all background information is ignored.

Table 4: Most frequent organisms in IntAct (derived from interactor proteins only)

ORG	freq
HUMAN	0.380
MOUSE	0.123
YEAST	0.108
ARATH	0.080
RAT	0.047
DROME	0.040
SCHPO	0.032
ECOLI	0.019
BOVIN	0.016
CAEEL	0.014

- While in some cases the annotators seem to decide that an organism is just used as part of the method and does not merit an inclusion, in other cases the annotators do not seem to treat the problem the same way.

An analysis of the most frequent false negatives is reported in table 3. The ranking is certainly not identical with the frequency table (table 1), which was unexpected. Manual inspection of some of the papers causing these false negatives gave the following results:

- Some common names such as "mice", and adjectives such as "murine", were absent from the taxonomy (while "transgenic mice" e.g. was present).
- There are probably more hints to recognize ECOLI (Escherichia coli K12) than just the presence of the string "K12" (or "K-12"). Our program tends to attribute all mentions of "Escherichia coli" without this string to ECOLX, generating false negatives for ECOLI and false positives for ECOLX.
- The extremely high false negative rate for CERAE (Chlorocebus aethiops, also known as Cercopithecus aethiops) is a consequence of its very different frequencies as source of interactor proteins and as a host organism.

The problem with CERAE suggests that it might be necessary to consider separately organisms in their roles as sources of the interactor proteins and as hosts for the experiments. CERAE is only frequent as a host organism, but in this role it does not appear in the papers by any of the organism names given by the taxonomy (such as "Chlorocebus aethiops", "Cercopithecus aethiops", "African green monkey", "grivet", "savanah monkey" or "vervet monkey"). The reason is that often only the names of cell lines (e.g. "Vero") derived from the organism appear in the paper.[7] To a lesser degree, this is true as well for papers where YEAST appears in this role.

A first step to deal with this problem consisted in creating different frequency tables for organisms as source of interactor proteins and as hosts of the experiment (tables 4 and 5). As these frequency tables are very different from each other and from the combined one (table 1) and as the combined task of identifying "protein organisms" and "host organisms" seems to be artificial in any case, we decided to split the problem accordingly: (a) identify organisms from which interacting proteins are derived; (b) identify host organisms. The results for each of these new tasks are not yet as good as the result for the combined task we described above, but as the information we are looking for now is more specific, this was to be expected.

3.1 Identification of "Interactor Organisms"

In order to obtain a solution for this more specific task, we just kept the formula as for the original task, but replaced the frequency table for "interactor and host organisms" (table 1) by a new one for "interactors only" (table 4). At the same time we raised the threshold to 18%: as the new freqency tables tended to nearly eliminate several typical host organisms, the remaining candidates for "interactor organisms" profited by this, so the threshold had to be raised to maximize f-score. The rest of the parameters remained identical.

Obviously, a new gold standard for "interactors only" had to be derived from IntAct. Our best results for this new task are: precision: 0.697; recall: 0.693; f-score: 0.695.

3.2 Identification of "Host Organisms"

For this alternative task we also had to improve the input, not just the formula, as we noticed that of-

[7] The Vero lineage is a very popular cell line isolated from kidney epithelial cells extracted from an African green monkey ("Cercopithecus aethiops").

Table 5: Most frequent organisms in IntAct (host organisms only), freq* is computed excluding "in vitro"

ORG	freq	freq*
"in vitro"	0.363	-
YEAST	0.262	0.412
HUMAN	0.167	0.264
CERAE	0.036	0.057
MOUSE	0.035	0.055
ARATH	0.021	0.034
DROME	0.021	0.034
SCHPO	0.020	0.031
ECOLX	0.017	0.027
RAT	0.010	0.015

ten species which were given as hosts by IntAct were not mentioned by any of their names (most importantly CERAE). So we decided to include another category of biological named entities in our termbase, namely cell line names. These were derived from one of the largest collections of cell lines information: the Cell Lines Knowledge Base (CLKB, (Sarntivijai et al., 2008)). However, a few cell line names which are type-ambiguous with other types of NE's in our termbase (normally proteins) had to be ignored to avoid conflicts. Another new input to the formula was the mention of "in vitro", contained in our termbase as a method, but used by the IntAct annotators as annotation for the "host organism".

The following adaptations to the ranking formula were necessary. The frequency table for "interactor and host organisms" (table 1) was replaced by a new one for "hosts only", including "in vitro" (table 5). At the same time the default had to be changed to "in vitro" and was given a nearly identical weight of 30 fulltext mentions (instead of 28), the threshold remained at 4% and the abstract mentions were given a weight of 35 fulltext mentions. The new cell line mentions were given a weight of 3 fulltext mentions for their respective organisms. Of course, a new gold standard for "interactors only" was derived from IntAct also in this case. Our best results yet for this new task are: precision: 0.689; recall: 0.737; f-score: 0.712.

4 Related Work and Discussion

The task of organism recognition is only recently starting to emerge as an independent subtask in biomedical IE. For example, the latest BioCreative competitive evaluation of text mining system for biology[8] included a task of protein-protein interaction detection (Krallinger et al., 2008). Although organism recognition was not officially evaluated, many participants found that it was an indispensable step in order to perform accurate protein recognition and disambiguation. As a consequence, the BioCreative meta-server (Leitner et al., 2008), offers organism recognition as one of its services (called "TX task").

(Wang and Matthews, 2008) is perhaps the most comprehensive study to date dealing with species disambiguation for term disambiguation. They combine a rule-based species disambiguation approach with a maximum entropy classifier based on contextual features of the term to be disambiguated. They evaluate in detail the contribution of both approaches over two separate corpora. While previous work has shown the benefits of using species information for term disambiguation (Alex et al., 2008; Rinaldi et al., 2008), this is perhaps the first study which also provides a separate evaluation of species disambiguation in itself. Since their purpose is to use the organism mentions to disambiguate entities, they evaluate how far their system can identify the organisms associated with each entity mention in the document. They report a level of accuracy that reaches 74.24% on one of their test corpora.

Since our results are for whole articles, not single entity mentions, they are not directly comparable. The advantage of our approach resides in its simplicity, since it does not require a specifically designed training set, being based only on publicly available standard databases. This reduces not only the cost compared to building own resources, but also ensures that their quality is monitored.

In this paper we have not discussed how our results can be used in the disambiguation of entities. As long as only one organism is selected as the focus of a given research publication, this is a rather trivial task. However, as mentioned already in section 2.4, it is often the case that multiple organisms are considered within the same publication. In that

[8] http://www.biocreative.org/

case, organism mentions would need to be 'localized' within the article in order to serve for disambiguation purposes, as done in (Wang and Matthews, 2008). Our own approach to this problem is presented and discussed in (Kaljurand et al., 2009).

One important limitation of our approach is its reliance on explicit mentions of organisms by their names as stored in the termbase (or minor variants thereof). Using all the names available to us (including cell lines) and their variants we could so far achieve only a maximal value of 88% recall, which means that 12% of the organisms are not referred to by any name in our resources. This may be due to either missing names in the termbase (the organisms are mentioned, but by different names) or because they are identified by human readers through other contextual hints which may consist of any sort of information,[9] and may presuppose massive amounts of background knowledge. The first problem might be adressed by adding other sources of names to our termbase. The second problem might be adressed by using a machine learning approach, which however brings with it a whole set of new problems, such as selection and representation of the features relevant for training, as well as the fact that a sufficiently large training corpus needs to be available.

Another limitation of our approach is the fact that its development and testing rests on its application to the identification of either organisms of protein interactors or host organisms. The original formulation of the goal that motivated this work was "to identify automatically the organisms forming part of the subject matter of scientific papers". This leaves open the question of the application of the results, and is deliberately vague in the wording "part of the subject matter", which includes but is not confined to the cases mentioned above. This formulation was motivated by a desire to keep the task as generic as possible, so that the resulting application could not only be used as a module for the protein disambiguation task, but also for other tasks of NE disambiguation with respect to organisms, as well as for organism identification as an independent task. Additionally, the ranked list of candidate organisms delivered by our program could also be presented to human users, who might want to use them in novel ways, for example in an assisted curation environment.

However, the gold standard by which we test our results is tailored to its application as a protein disambiguation module, just as the frequency tables we use. Even apart from this, the appropriateness of the gold standard is partly questionable, as it does not only prefer organisms involved in protein interactions to those that are not, but also "new" knowledge to "old" knowledge, etc. Our approach, based on "correcting" simple counts of organism mentions using frequency tables, can only be successful as long as there is a gold standard for the specific application that is being pursued. We can derive from IntAct useful gold standards for organisms from which protein interactors are derived or host organisms, but we have no gold standard for "organism identification" as an independent task.

5 Conclusion

In this paper we discussed an approach to the problem of "organism identification" as an independent task, based only on standard resources. While the initial results were interesting, the experimental setup led us to identify more specific aspects of the problem, and in particular to distinguish organisms mentioned in their roles as sources of the interacting proteins and as hosts of the experiments. We have shown that a clear identification of the different functional roles played by organism mentions can lead to more accurate results.

Although a fully automated disambiguation process based on organism mentions is not within immediate reach, the results described in this paper appear already potentially useful for protein name disambiguation in a curation environment. Another possible application would be in biomedical curation-based databases, for the semi-automatic tagging of publications with their focus organisms.

Acknowledgements

This research is partially funded by the Swiss National Science Foundation (grant 100014-118396/1). Additional support is provided by Novartis Pharma AG, NITAS, Text Mining Services, CH-4002, Basel, Switzerland. We thank the anonymous reviewers for their insightful comments.

[9] A trivial example would be a publication in a journal which specializes in research on a single organism.

References

[Alex et al.2008] Beatrice Alex, Claire Grover, Barry Haddow, Mijail Kabadjov, Ewan Klein, Michael Matthews, Richard Tobin, and Xinglong Wang. 2008. Automating curation using a natural language processing pipeline. *Genome Biology*, 9(Suppl 2):S10.

[Kaljurand et al.2009] Kaarel Kaljurand, Fabio Rinaldi, Thomas Kappeler, and Gerold Schneider. 2009. Using existing biomedical resources to detect and ground terms in biomedical literature. In *12th Conference on Artificial Intelligence in Medicine (AIME'09)*, Verona, Italy, 18–22 July.

[Kappeler et al.2008] Thomas Kappeler, Simon Clematide, Kaarel Kaljurand, Gerold Schneider, and Fabio Rinaldi. 2008. Towards automatic detection of experimental methods from biomedical literature. In *Third International Symposium on Semantic Mining in Biomedicine (SMBM)*.

[Kerrien et al.2006] S. Kerrien, Y. Alam-Faruque, B. Aranda, I. Bancarz, A. Bridge, C. Derow, E. Dimmer, M. Feuermann, A. Friedrichsen, R. Huntley, C. Kohler, J. Khadake, C. Leroy, A. Liban, C. Lieftink, L. Montecchi-Palazzi, S. Orchard, J. Risse, K. Robbe, B. Roechert, D. Thorneycroft, Y. Zhang, R. Apweiler, and H. Hermjakob. 2006. IntAct — Open Source Resource for Molecular Interaction Data. *Nucleic Acids Research*.

[Krallinger et al.2008] Martin Krallinger, Florian Leitner, Carlos Rodriguez-Penagos, and Alfonso Valencia. 2008. Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome Biology*, 9(Suppl 2):S4.

[Leitner et al.2008] Florian Leitner, Martin Krallinger, Carlos Rodriguez-Penagos, Jörg Hakenberg, Conrad Plake, Cheng-Ju Kuo, Chun-Nan Hsu, Richard Tzong-Han Tsai, Hsi-Chuan Hung, William W. Lau, Calvin A. Johnson, Rune Saetre, Kazuhiro Yoshida, Yan Hua Chen, Sun Kim, Soo-Yong Shin, Byoung-Tak Zhang, William A. Baumgartner, Lawrence Hunter, Barry Haddow, Michael Matthews, Xinglong Wang, Patrick Ruch, Frédéric Ehrler, Arzucan Özgür, Günes Erkan, Dragomir R. Radev, Michael Krauthammer, ThaiBinh Luong, Robert Hoffmann, Chris Sander, and Alfonso Valencia. 2008. Introducing meta-services for biomedical information extraction. *Genome Biology*, 9(Suppl 2):S6.

[Rinaldi et al.2008] Fabio Rinaldi, Thomas Kappeler, Kaarel Kaljurand, Gerold Schneider, Manfred Klenner, Simon Clematide, Michael Hess, Jean-Marc von Allmen, Pierre Parisot, Martin Romacker, and Therese Vachon. 2008. OntoGene in BioCreative II. *Genome Biology*, 9(Suppl 2):S13.

[Sarntivijai et al.2008] Sirarat Sarntivijai, Alexander S. Ade, Brian D. Athey, and David J. States. 2008. A bioinformatics analysis of the cell line nomenclature. *Bioinformatics*, 24(23):2760–2766.

[UniProt Consortium2007] UniProt Consortium. 2007. The universal protein resource (UniProt). *Nucleic Acids Research*, 35:D193–7.

[Wang and Matthews2008] Xinglong Wang and Michael Matthews. 2008. Distinguishing the species of biomedical named entities for term identification. *BMC Bioinformatics*, 9(Suppl 11):S6.

[Zanzoni et al.2002] A. Zanzoni, L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni. 2002. MINT: a Molecular INTeraction database. *FEBS Letters*, 513(1):135–140.

Exploring graph structure for detection of reliability zones within synonym resources: Experiment with the *Gene Ontology*

Thierry Hamon
LIPN – UMR 7030
Université Paris 13 – CNRS
99 av. J-B Clément
F-93430 Villetaneuse, France
thierry.hamon@lipn.univ-paris13.fr

Natalia Grabar
Centre de Recherche des Cordeliers
Université Paris Descartes, UMR_S 872
INSERM, U872
HEGP AP-HP, 20 rue Leblanc
Paris, France
natalia.grabar@spim.jussieu.fr

Abstract

Computing the semantic similarity between terms relies on existence and usage of semantic resources. However, these resources, often composed of equivalent units, or synonyms, must be first analyzed and weighted in order to define within them the reliability zones where the semantic cohesiveness is stronger. We propose an original method for acquisition of elementary synonyms based on exploitation of structured terminologies, analysis of syntactic structure of complex (multi-unit) terms and their compositionality. The acquired synonyms are then profiled thanks to endogenous lexical and linguistic indicators (other types of relations, lexical inclusions, productivity), which are automatically inferred within the same terminologies. Additionally, synonymy relations are observed within graph, and its structure is analyzed. Particularly, we explore the usefulness of the graph theory notions such as connected component, clique, density, bridge, articulation vertex, and centrality of vertices.

1 Introduction

In various tasks and applications of natural language processing and of biomedical informatics (*i.e.*, query expansions, information retrieval, text mining, information extraction or terminology matching), it is important to be able to decide whether two terms (*i.e.*, *acetone anabolism* and *acetone biosynthesis*, *replication of mitochondrial DNA* and *mtDNA replication*) convey the same or different meaning. This is particularly important for deciphering and computing semantic similarity between words and terms. Lexicon of specific resources (synonym, morphological or orthographic variants) can be used for detection of semantic similarity. However, depending on languages and domains, such resources are not equally well described. Morphological description is the most complete for both general (Burnage, 1990; Hathout et al., 2001) and biomedical (NLM, 2007; Schulz et al., 1999; Zweigenbaum et al., 2003) languages. But the situation is not as successful at the semantic level: little synonym resources can be found. If WordNet (Fellbaum, 1998) proposes general language synonym relations for English, the corresponding resources for other languages are not freely available. Moreover, the initiative for fitting WordNet to the biomedical area (Smith and Fellbaum, 2004) seems to have been abandoned, although there is a huge need for this kind of resources.

In our previous work, we proposed to use the existing biomedical terminologies (*i.e.*, *Gene Ontology* (Gene Ontology Consortium, 2001), Snomed (Côté et al., 1997), UMLS (NLM, 2007)), wich provide complex terms, and to acquire from them lexical resources of synonyms. Indeed, the use of complex biomedical terms seems to be less suitable and generalizable as compared to lexical resources (Poprat et al., 2008). Within the biological area, we proposed to exploit the *Gene Ontology* (*GO*), and more specifically to exploit compositional structure of its terms (Hamon and Grabar, 2008). However, with the acquisition of synonymy we faced two problems: (1) contextual character of these relations (Cruse, 1986), *i.e.*, two terms or words are considered as synonyms if they can occur within the

Proceedings of the Workshop on BioNLP, pages 89–96,
Boulder, Colorado, June 2009.

same context, which makes this relation more or less broad depending on the usage; (2) ability of automatic tools to detect and characterize these relations, *i.e.*, two terms or words taken out of their context can convey different relations than the one expected. Because we aim at acquiring synonymy resources which could be used by various applications and on various corpora, we need to profile them and possibly to detect the reliability zones. We proposed to do this profiling through lexical and linguistic indicators generated within the same terminology (Grabar et al., 2008), such as productivity, cooccurence with other types of relations (`is-a`, `part-of`) and with lexical inclusion. These indicators on reliability zones will be used for defining the synonymity degree of terms and for preparing the validation of the acquired synonym resources. In the current work, we continue profiling the acquired synonyms, but rely on the form of the graph built from pairs of synonyms. We exploit for this some notions of the graph theory (Diestel, 2005). In the following of this paper, we first present our material (sec. 2) and methods (sec. 3), we then present and discuss results (sec. 4) and conclude with some perspectives (sec. 5).

2 Material

We use the *Gene Ontology* (*GO*) as the original resource from which synonym lexicon (or elementary synonym relations) are induced. The goal of the *GO* is to produce a structured, common, controlled vocabulary for describing the roles of genes and their products in any organism. *GO* terms convey three types of biological meanings: biological processes, molecular functions and cellular components. Terms are structured through four types of relationships: subsumption `is-a`, meronymy `part-of`, synonymy and `regulates`. The version, we used in the current work, was downloaded in February 2008[1]. It provides 26,057 concepts and their 79,994 terms. When we create pairs of terms, which we exploit with our methods, we obtain 260,399 `is-a`, 29,573 `part-of` and 459,834 synonymy relations. There are very few `regulates` relations, therefore we don't exploit them in our work.

[1]Our previous work has been performed with an anterior version of the *GO*.

3 Methods

GO terms present compositional structure, like within the concept GO:0009073, where compositionality can be observed through the substitution of one of the components (underlined):

aromatic amino acid family biosynthesis
aromatic amino acid family anabolism
aromatic amino acid family formation
aromatic amino acid family synthesis

Compositionality of the *GO* terms has been exploited previously, for instance (Verspoor et al., 2003) propose to derive simple graphs from relations between complex *GO* terms, (Mungall, 2004) exploits the compositionality as a mean for consistency checking of the *GO*, (Ogren et al., 2005) use it for enriching the *GO* with missing synonym terms. We propose to exploit the compositionality for induction of synonym lexical resources (*i.e.*, *biosynthesis*, *anabolism*, *formation*, *synthesis* in the given example). While the cited works are based on the string matching within *GO* terms, our approach aims at exploiting the syntactic analysis of terms, which makes it independent from the graphical form of the analyzed terms (like examples on fig. 1). Our method has several steps: linguistic preprocessing of the *GO* terms (sec. 3.1), induction of elementary semantic lexicon (sec. 3.2), and then the profiling the synonymy lexicon through the lexical and linguistic indicators (sec. 3.3), and through the analysis of connected components built from the induced synonym pairs (sec. 3.4). Steps 3.1 to 3.3 have been already described in our previous work: we mention here the main notions for the sake of clarity.

3.1 Preprocessing the *GO* terms: Ogmios NLP platform

The aim of terminology preprocessing step is to provide syntactic analysis of terms for computing their syntactic dependency relations. We use the `Ogmios` platform[2] and perform: segmentation into words and sentences; POS-tagging and lemmatization (Schmid, 1994); and syntactic analysis[3]. Syntactic dependencies between term components are

[2]http://search.cpan.org/∼thhamon/Alvis-NLPPlatform/
[3]http://search.cpan.org/∼thhamon/Lingua-YaTeA/

Figure 1: Parsing tree of the terms *replication of mitochondrial DNA* and *mtDNA replication*.

computed according to assigned POS tags and shallow parsing rules. Each term is considered as a syntactic binary tree composed of two elements: head component and expansion component. For instance, *replication* is the head component of the two terms analyzed on figure 1.

3.2 Acquiring the elementary semantic relations

The notion of compositionality assumes that the meaning of a complex expression is fully determined by its syntactic structure, the meaning of its parts and the composition function (Partee, 1984). On the basis of syntactically analysed terms, we apply a set of compositional rules: if the meaning $\mathcal{M}$ of two complex terms $A\ rel\ B$ and $A'\ rel\ B$, where A is its head and B its expansion components, is given as following:

$$\mathcal{M}(A\ rel\ B) = f(\mathcal{M}(A), \mathcal{M}(B), \mathcal{M}(rel))$$

$$\mathcal{M}(A'\ rel\ B) = f(\mathcal{M}(A'), \mathcal{M}(B), \mathcal{M}(rel))$$

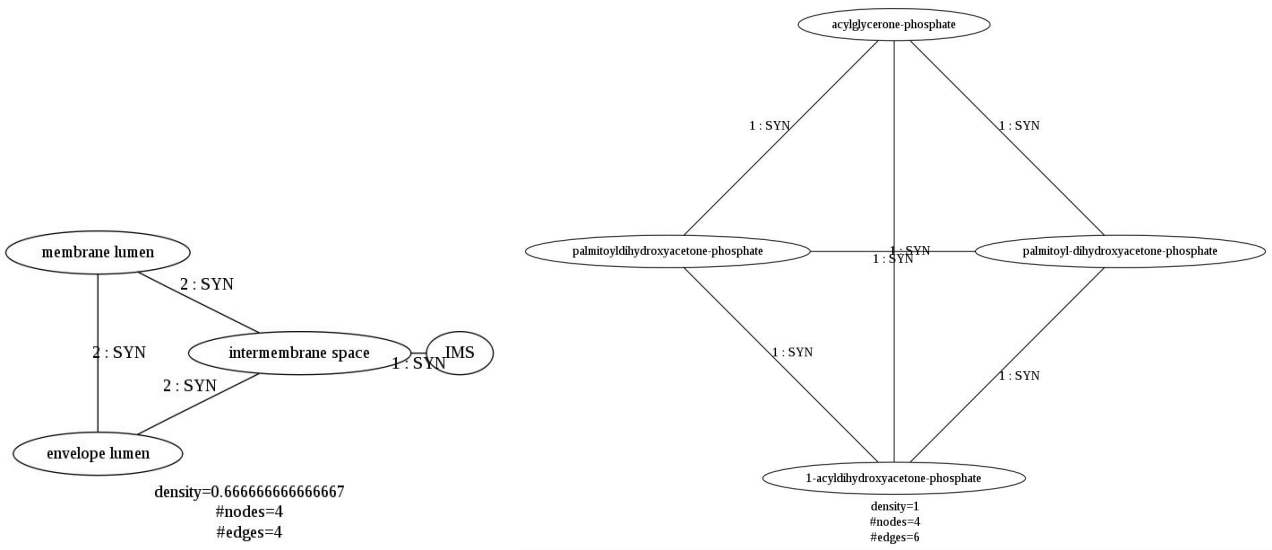
for a given composition function f, if $A\ rel\ B$ and $A'\ rel\ B$ are complex synonym terms and if B components are identical (such as *acetone* within *acetone catabolism* and *acetone breakdown*), then the synonymy relation between components A and A' {*catabolism*, *breakdown*} can be induced. The modification is also accepted on expansion component B: from terms *replication of mitochondrial DNA* and *mtDNA replication* (fig. 1), we can induce synonymy between *mitochondrial DNA* and *mtDNA*. Finally, the modification is also accepted for both components $A\ rel\ B$ and $A'\ rel\ B'$, such as in *nicotinamide adenine dinucleotide catabolism* and *NAD breakdown*, where one pair, *i.e.* {*catabolism*, *breakdown*}, can be known from previously processed synonyms and allow to induce the new pair {*nicotinamide adenine dinucleotide*, *NAD*}. The method is recursive and each induced elementary synonym relation can then be propagated in order to induce new elementary relations, which allows to generate a more exhaustive lexicon of synonyms.

This method is not specific to the synonymy. As it works at the syntactic level of terms, it therefore can be applied to other relationships: relationship between elementary terms is inherited from the relationship between complex terms. If we exploit complex terms related with `part-of` relations and if the compositionality rules can be applied, then we can induce elementary `part-of` relations. For instance, complex terms *cerebral cortex development* GO:0021987 and *cerebral cortex regionalization* GO:0021796 have a `part-of` relation between them, and we can induce the elementary `part-of` relation between their components *development* and *regionalization*. Similarly, on the basis of two *GO* terms that have `is-a` relation between them, *cell activation* GO:0001775 and *astrocyte activation* GO:0048143, we can induce the elementary `is-a` relation between *cell* and *astrocyte*.

3.3 Exploiting lexical and linguistic indicators

Several endogenously generated indicators are used for profiling the induced lexicon of synonyms:

- Elementary `is-a` relations;
- Elementary `part-of` relations;
- Lexical inclusion: terms within each induced synonymy pair are controlled for the lexical inclusion. If the test is positive, like in {*DNA binding*, *binding*}, this would suggest that the analyzed terms may convey a hierarchical relation: indeed, lexical subsumption marks often a hierarchical subsumption (Kleiber and Tamba, 1990), which can be either `is-a` or `part-of` relations;
- Productivity: number of original *GO* pairs from which this elementary relation is inferred. For instance, synonymy relations {*binding*, *DNA*



(a) Connected component of synonyms

(b) Clique of synonyms

Figure 2: Connected components formed with pairs of elementary synonym relations.

binding$\}$ and $\{$*cell*, *lymphocyte*$\}$ are inferred from only one original pair of *GO* synonyms, while the pair $\{$*T-cell*, *T-lymphocyte*$\}$ is supported by eight original *GO* synonym pairs.

Factors that would weaken synonymy relations and make them less reliable are their co-occurrence with lexical inclusions, `is-a` or `part-of` relations, and their low productivity.

3.4 Exploiting the graph theory notions

Pairs of induced synonyms are observed through the connected components they form: lexical entries are *nodes or vertices* and relations between them are *edges or paths*. For instance, connected component 2(a) contains four pairs of synonyms: $\{$*membrane lumen*, *envelope lumen*$\}$, $\{$*membrane lumen*, *intermembrane space*$\}$, $\{$*envelope lumen*, *intermembrane space*$\}$ and $\{$*intermembrane space*, *IMS*$\}$. On each edge, we projected information associated with the relation corresponding to this edge. For instance, $\{$*membrane lumen*, *intermembrane space*$\}$ relation is labelled as synonymy SYN and shows 2 as productivity value (it has been acquired from two original pairs of synonyms within *GO*). If other relationships ($INCL$, PAR, $HIER$) are associated to a given synonymy relation, they are also indicated together with their productivity.

As a matter of fact, figure 2 presents two typical examples of connected components we can obtain (in these examples, both of them have four nodes):

- *Connected component* (fig. 2(a)) is a graph in which any two vertices are connected to each other by edges. Connected components have not orphan vertices, which would remain not connected to any other vertex.

- *Clique*, also called *block* (fig. 2(b)) is a particular case of connected components: clique is a maximally connected component. In such graphs, all the vertices are interconnected between them.

We propose to exploit four more notions of the graph theory, which we assume can be useful for further profiling of the acquired synonymy relations:

- *Density* of a connected component is the ratio between the number of its edges and the number of edges of the corresponding clique. For instance, the connected component on figure 2(a) has 4 edges while the corresponding clique would have 6 edges. In that respect, this connected component has the dentisty of 0.67. Besides, the clique on figure 2(b) shows the maximum density (*i.e.*, 1). (For all the fig-

ures, we indicate their density, together with the number of vertices and edges).

- *Bridge* is defined as an edge which removal would increase the number of connected components. For instance, within connected component 2(a), removing the edge {*intermembrane space*, *IMS*} would lead to the creation of two new connected components: (1) single-vertex component *IMS*, and (2) connected component with three vertices *intermembrane space*, *membrane lumen* and *envelope lumen*. Consequently *articulation vertices* are defined as vertices which removal would increase the number of connected components. At figure 2(a), the articulation vertex is *intermembrane space*.

- The *centrality of a vertex* is defined as the number of shortest paths passing through it. For instance, on figure 2(a), *intermembrane space*'s centrality is 4, while the centrality of other vertices is null.

4 Results and Discussion

4.1 Acquiring the elementary synonymy relations and their lexical and linguistic profiling

79 994 *GO* terms have been fully analyzed through the Ogmios platform. Compositional rules (sec. 3.2) have been applied and allowed to induce 9,085 semantic relations among which: 3,019 synonyms, 3,243 `is-a` and 1,205 `part-of`. 876 lexical inclusions have discovered within all these elementary pairs. 2,533 synonymy pairs are free of the lexical profiling indicators. However, 486 synonymy relations (16%) cooccur with other relations, and the details of this cooccurrence is showed in table 1. We can observe for instance that 142 synonym pairs are also labelled as `is-a` relations, and 34 as `part-of` relations. Productivity of the induced synonyms is between 1 and 422 original complex *GO* terms.

Connected component on figure 3 illustrates coocurrence of synonymy relations with other types of relations: the pair {*import*, *ion import*} shows synonym and inclusion relations; the pair {*import*, *uptake*} shows synonym and hierarchical relations, both acquired on seven original pairs of *GO* terms.

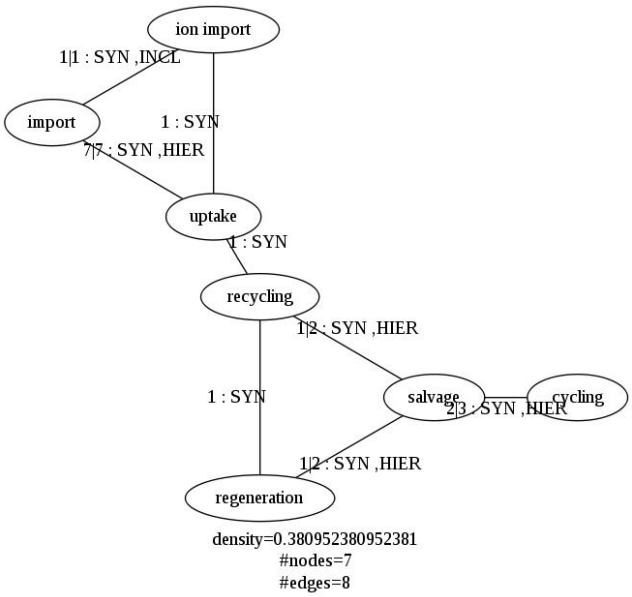


Figure 3: Connected component where synonymy relations cooccur with other relations.

Synonymy and other relations	Number
syno ∩ is-a	142
syno ∩ par	34
syno ∩ incl	309
syno ∩ par ∩ is-a	14
syno ∩ incl ∩ is-a \ par	40
syno ∩ incl ∩ par \ is-a	2
syno ∩ incl ∩ is-a ∩ par	1

Table 1: Number of synonymy relations which cooccur with other relations (`is-a`, `part-of` and lexical inclusions `incl`).

4.2 Analysing the induced synonym pairs through the graph theory

3,019 induced synonym pairs have been grouped into 1,018 connected components. These components contain 2 to 69 nodes, related among them by 1 to 132 edges. Analyses of the connected components have been performed with Perl package `Graph` and additionnal Perl scripts. Among the studied connected components, we have 914 cliques composed of 2 (n=708), 3 (n=66), 4 (n=88), 5 (n=44) or 6 (n=8) nodes. The remaining 104 connected components are less dense with edges. The density of the connected components is between

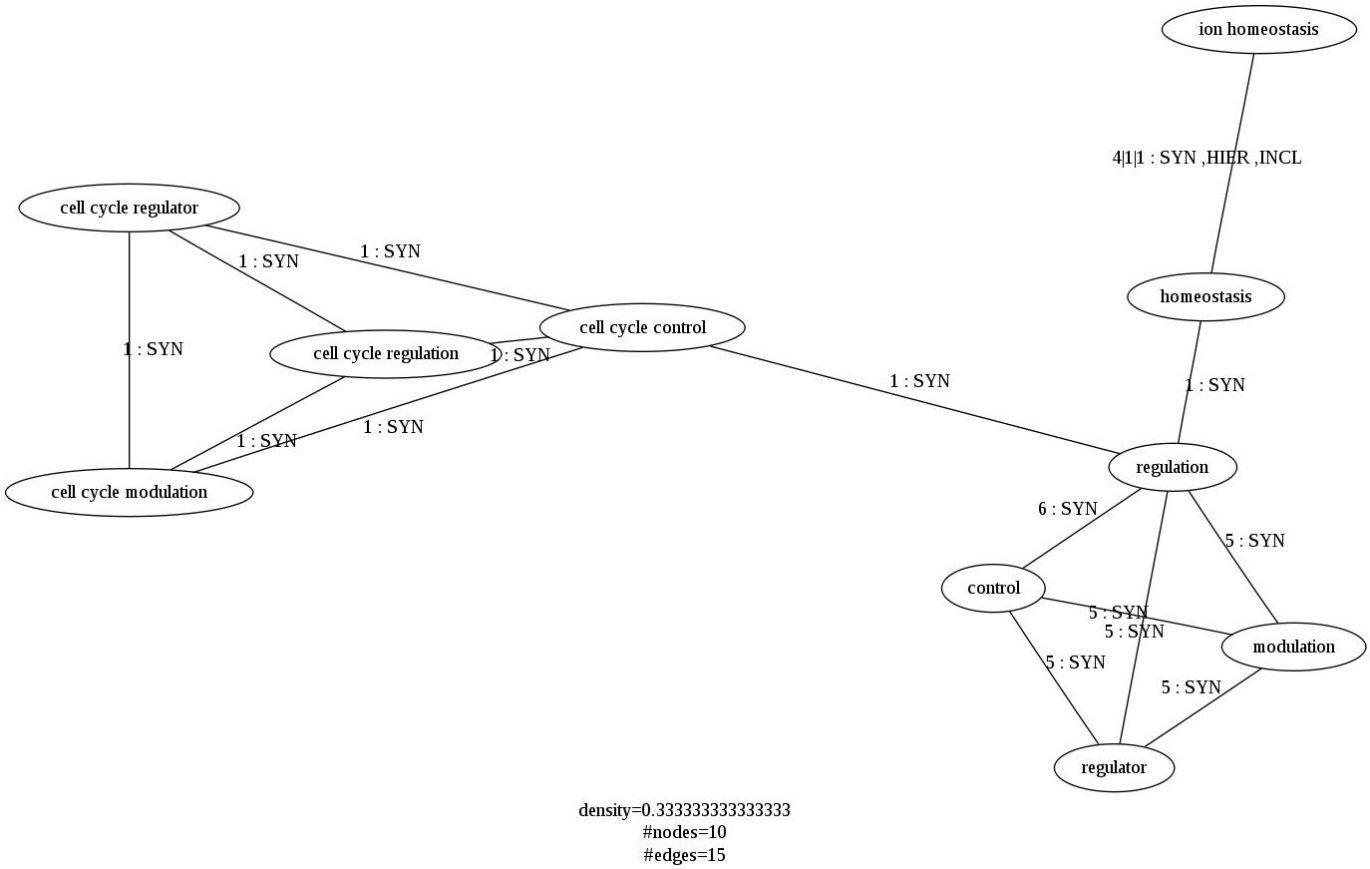


Figure 4: Connected component with three bridges: {*ion homeostasis*, *homeostasis*}, {*homeostasis*, *regulation*} and {*cell cycle control*, *regulation*}.

0.0467 and 1 (in case of cliques). Among the 104 connected components, which are not cliques, we detected 249 bridges: 0 to 35 depending on connected components. In order to propose a general approach exploiting graph theory notions for synonym profiling we analyse the structure of three representative connected components.

Density of the connected component 2(a) is 0.67. It contains one bridge: {*intermembrane space*, *IMS*}. This edge corresponds to the acronym and its expanded form, which can cause its contextual character. Moreover, *intermembrane space* is the central node of this connected component.

Connected component 3 (density=0.38) contains two bridges {*uptake*, *recycling*} and {*salvage*, *cycling*}, and three articulation vertices *uptake*, *recicling* and *salvage* with the measures of centrality 16, 18 and 10 respectively. Indeed, the majority of shortest paths pass by *uptake* and *recicling* nodes. Otherwise, edges around the *salvage* vertex are weakened because of the cooccurrence of synonymy and hierarchical relations. As we have already noticed, the edge {*import*, *uptake*} shows the cooccurrence of synonymy and hierarchical relations, but its productivity is rather high (seven for each relation), which stregthens this edge.

Finally, connected component 4 (density=0.33) contains three bridges {*ion homeostasis*, *homeostasis*}, {*homeostasis*, *regulation*} and {*cell cycle control*, *regulation*} and three articulation vertices: *regulation*, *cell cycle control* and *homeostasis* with the measures of centrality 52, 37 and 16 respectively. The bridge {*ion homeostasis*, *homeostasis*} is weakened by the cooccurrence of synonymy, hierarchical and lexical inclusion relations. Otherwise, other edges seem to convey non ambiguous synonymy.

From the analyzed examples, we can see that the graph theory may have several implications on profiling of synonyms. However, these implications must still be formalized and, possibly, expressed as a single reliability indicator, alone or combined with the lexical and linguistic clues.

First, within a connected component, with a given number of nodes, higher the number of edges, higher will be its density and closer it will be to a clique (fig. 2(b)). Consequently, within a clique, the semantic cohesion is more strong. Indeed, in these cases, terms are far more strongly related between them. But when the density value decreases the semantic cohesiveness of connected components decreases as well. In other words, density is an indication on the semantic cohesiveness between terms within connected components. As for bridges, we assume that they indicate breaking points within connected components, such as {*cell cycle control*, *regulation*} within figure 4. The weak character of these points can increased when the synonymy relation co-occurs with other relationships (`is-a`, `part-of`, lexical inclusion). Consequently, removal of bridges can create connected components with higher density and therefore with stronger synonymy relations. Finally, the centrality of vertices measure may be useful for identification of polysemic words or terms.

The connected components analysis can also indicate the missing relations. For instance, if a connected component, which is not a clique, has no bridges but its density is not maximal, this would indicate that it misses some correct synonymy relations which can be easily induced.

5 Conclusion and Perspectives

In this paper, we propose an original method for inducing synonym lexicon from structured terminologies. This method exploits the compositionality principle and three rules based on syntactic dependency analysis of terms. More specifically, we explore various indicators for profiling the acquired synonym relations, which is motivated by the fact that synonymy is a contextual relation and its validity and universality is not guaranteed. We assume the semantic cohesiveness of synonymy relations should be qualified and quantified. Thus, we propose several indicators for profiling the inferred synonymy relations and for detecting possible weak and strong points. First, lexical and linguistic clues are generated endogenously within the same terminology: other types of elementary semantic relations (`is-a` and `part-of`), lexical inclusions and productivity of the acquired semantic relations. Then, more specifically, this work is dedicated to exploring of the usefulness of notions of the graph theory. We propose to study the form and specificities of connected components formed by synonymy relations. We exploited the following notions from the graph theory: distinction between connected components and cliques, their density, bridges and articulation vertices within connected components, and the centrality of their vertices. We observed that the lexical indicators as well as connected components characteristics are helpful for profiling the acquired synonymy relations. These clues are intended to be used for preparing the validation of this lexicon by experts and also for its weighting in order to control and guarantee the specificity of lexicon during its use by automatic tools.

Currently, we study separately the endogeneous lexical indicators, and the characteristics of the connected components. However, in the future, these two types of clues should be combined. For this, these indicators should be modelized in order to provide a weight of each edge. This weight can be used for profiling of connected component through the detection of strong and weak points. Notice that the current version of the `Graph` package cannot take into account this additional information on edges and should be modified. Another perspective is the better exploitation of the *Gene Ontology* and taking into account the nature of synonymy relations as they are labelled by thier creators: *exact*, *broad*, *narrow* or *related*. Additionnally, for a more precise profiling, the four relationships of *GO* (synonymy, `is-a`, `part-of` and `regulates`) can be cross-validated, while currently, we perform the validation of synonymy relations through `is-a` and `part-of` (and other indicators). We plan also to use the induced relations and propagate them through corpora and discover some of the missing synonyms (Hole and Srinivasan, 2000). In this way, applying the same compositionality principle, we can enrich and extend the *Gene Ontology*: new synonyms of *GO*

terms and even other relations between *GO* terms and terms from corpora can be detected. As noticed, this method can be applied to other terminologies and languages as far as structured terminological resources and NLP tools exist. For instance, within the context of search of clinical documents, we successfully tested this method on the French part of the UMLS (Grabar et al., 2009). From a more ontological perspective, our method can be used for consistency checking of a terminologies, like in (Mungall, 2004). Moreover, as this method performs syntactic analysis of terms and their decomposition into semantically independent components, it can be used for the transformation of a pre-coordinated terminology into a post-coordinated one.

References

G. Burnage. 1990. *CELEX - A Guide for Users*. Centre for Lexical Information, University of Nijmegen.

Roger A. Côté, Louise Brochu, and Lyne Cabana, 1997. *SNOMED Internationale – Répertoire d'anatomie pathologique*. Secrétariat francophone international de nomenclature médicale, Sherbrooke, Québec.

David A. Cruse. 1986. *Lexical Semantics*. Cambridge University Press, Cambridge.

Reinhard Diestel. 2005. *Graph Theory*. Springer-Verlag Heidelberg, New-York.

Christian Fellbaum. 1998. A semantic network of english: the mother of all WordNets. *Computers and Humanities. EuroWordNet: a multilingual database with lexical semantic network*, 32(2-3):209–220.

Gene Ontology Consortium. 2001. Creating the Gene Ontology resource: design and implementation. *Genome Research*, 11:1425–1433.

Natalia Grabar, Marie-Christine Jaulent, and Thierry Hamon. 2008. Combination of endogenous clues for profiling inferred semantic relations: experiments with gene ontology. In *JAMIA (AMIA 2008)*, pages 252–6, Washington, USA.

Natalia Grabar, Paul-Christophe Varoutas, Philippe Rizand, Alain Livartowski, and Thierry Hamon. 2009. Automatic acquisition of synonym ressources and assessment of their impact on the enhanced search in ehrs. *Methods of Information in Medicine*, 48(2):149–154. PMID 19283312.

Thierry Hamon and Natalia Grabar. 2008. Acquisition of elementary synonym relations from biological structured terminology. In *Computational Linguistics and Intelligent Text Processing (5th International Conference on NLP, 2006)*, number 4919 in LNCS, pages 40–51. Springer.

Nabil Hathout, Fiammetta Namer, and Georgette Dal. 2001. An experimental constructional database: the MorTAL project. In P. Boucher, editor, *Morphology book*. Cascadilla Press, Cambridge, MA.

WT Hole and S Srinivasan. 2000. Discovering missed synonymy in a large concept-oriented metathesaurus. In *AMIA 2000*, pages 354–8.

Georges Kleiber and Irène Tamba. 1990. L'hyperonymie revisitée : inclusion et hiérarchie. *Langages*, 98:7–32, juin. L'hyponymie et l'hyperonymie (dir. Marie-Françoise Mortureux).

CJ Mungall. 2004. Obol: integrating language and meaning in bio-ontologies. *Comparative and Functional Genomics*, 5(6-7):509–520.

NLM, 2007. *UMLS Knowledge Sources Manual*. National Library of Medicine, Bethesda, Maryland. www.nlm.nih.gov/research/umls/.

PV Ogren, KB Cohen, and L Hunter. 2005. Implications of compositionality in the Gene Ontology for its curation and usage. In *Pacific Symposium of Biocomputing*, pages 174–185.

Barbara H Partee, 1984. *Compositionality*. F Landman and F Veltman.

Michael Poprat, Elena Beisswanger, and Udo Hahn. 2008. Building a biowordnet using wordnet data structures and wordnet's software infrastructure - a failure story. In *ACL 2008 workshop "Software Engineering, Testing, and Quality Assurance for Natural Language Processing"*, pages 31–9.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Stefan Schulz, Martin Romacker, Pius Franz, Albrecht Zaiss, Rdiger Klar, and Udo Hahn. 1999. Towards a multilingual morpheme thesaurus for medical free-text retrieval. In *Medical Informatics in Europe (MIE)*.

Barry Smith and Christian Fellbaum. 2004. Medical wordnet: a new methodology for the construction and validation of information. In *Proc of 20th CoLing*, pages 371–382, Geneva, Switzerland.

Cornelia M Verspoor, Cliff Joslyn, and George J Papcun. 2003. The gene ontology as a source of lexical semantic knowledge for a biological natural language processing application. In *SIGIR workshop on Text Analysis and Search for Bioinformatics*, pages 51–56.

Pierre Zweigenbaum, Robert Baud, Anita Burgun, Fiammetta Namer, Éric Jarrousse, Natalia Grabar, Patrick Ruch, Franck Le Duff, Benot Thirion, and Stéfan Darmoni. 2003. Towards a Unified Medical Lexicon for French. In *Medical Informatics in Europe (MIE)*.

Towards Automatic Generation of Gene Summary

Feng Jin
Dept. Computer Science and Technology
Tsinghua University
Beijing 100084, China
jinfengfeng@gmail.com

Minlie Huang
Dept. Computer Science and Technology
Tsinghua University
Beijing 100084, China
aihuang@tsinghua.edu.cn

Zhiyong Lu
National Center for Biotechnology Information
National Library of Medicine
Bethesda, 20894, USA
luzh@ncbi.nlm.nih.gov

Xiaoyan Zhu
Dept. Computer Science and Technology
Tsinghua University
Beijing 100084, China
zxy-dcs@tsinghua.edu.cn

Abstract

In this paper we present an extractive system that automatically generates gene summaries from the biomedical literature. The proposed text summarization system selects and ranks sentences from multiple MEDLINE abstracts by exploiting gene-specific information and similarity relationships between sentences. We evaluate our system on a large dataset of 7,294 human genes and 187,628 MEDLINE abstracts using Recall-Oriented Understudy for Gisting Evaluation (ROUGE), a widely used automatic evaluation metric in the text summarization community. Two baseline methods are used for comparison. Experimental results show that our system significantly outperforms the other two methods with regard to all ROUGE metrics. A demo website of our system is freely accessible at http://60.195.250.72/onbires/summary.jsp.

1 Introduction

Entrez Gene is a database for gene-centric information maintained at the National Center for Biotechnology Information (NCBI). It includes genes from completely sequenced genomes (e.g. Homo sapiens). An important part of a gene record is the summary field (shown in Table 1), which is a small piece of text that provides a quick synopsis of what is known about the gene, the function of its encoded protein or RNA products, disease associations, genetic interactions, etc. The summary field, when available, can help biologists to understand the target gene quickly by compressing a huge amount of knowledge from many papers to a small piece of text. At present, gene summaries are generated manually by the National Library of Medicine (NLM) curators, a time- and labor-intensive process. A previous study has concluded that manual curation is not sufficient for annotation of genomic databases (Baumgartner et al., 2007). Indeed, of the 5 million genes currently in Entrez Gene, only about 20,000 genes have a corresponding summary. Even in humans, arguably the most important species, the coverage is modest: only 26% of human genes are curated in this regard. The goal of this work is to develop and evaluate computational techniques towards automatic generation of gene summaries.

To this end, we developed a text summarization system that takes as input MEDLINE documents related to a given target gene and outputs a small set of genic information rich sentences. Specifically, it first preprocesses and filters sentences that do

Proceedings of the Workshop on BioNLP, pages 97–105,
Boulder, Colorado, June 2009.

Gene	*Number of Abstracts*	*GO terms*	*Human-writtenSummary*
EFEMP1	26	calcium ion binding protein binding extracellular region proteinaceous extracellular matrix	This gene spans approximately 18 kb of genomic DNA and consists of 12 exons. Alternative splice patterns in the 5\' UTR result in three transcript variants encoding the same extracellular matrix protein. Mutations in this gene are associated with Doyne honeycomb retinal dystrophy.
IL20RA	15	blood coagulation receptor activity integral to membrane membrane	The protein encoded by this gene is a receptor for interleukin 20 (IL20), a cytokine that may be involved in epidermal function. The receptor of IL20 is a heterodimeric receptor complex consisting of this protein and interleukin 20 receptor beta (IL20B). This gene and IL20B are highly expressed in skin. The expression of both genes is found to be upregulated in Psoriasis.

Table1. Two examples of human-written gene summaries

not include enough informative words for gene summaries. Next, the remaining sentences are ranked by the sum of two individual scores: a) an authority score from a lexical PageRank algorithm (Erkan and Radev, 2004) and b) a similarity score between the sentence and the Gene Ontology (GO) terms with which the gene is annotated (To date, over 190,000 genes have two or more associated GO terms). Finally, redundant sentences are removed and top ranked sentences are nominated for the target gene.

In order to evaluate our system, we assembled a gold standard dataset consisting of handwritten summaries for 7,294 human genes and conducted an *intrinsic* evaluation by measuring the amount of overlap between the machine-selected sentences and human-written summaries. Our metric for the evaluation was ROUGE[1], a widely used intrinsic summarization evaluation metric.

2 Related Work

Summarization systems aim to extract salient text fragments, especially sentences, from the original documents to form a summary. A number of methods for sentence scoring and ranking have been developed. Approaches based on sentence position (Edmundson, 1969), cue phrase (McKeown and Radev, 1995), word frequency (Teufel and Moens, 1997), and discourse segmentation (Boguraev and Kennedy, 1997) have been reported. Radev et al. (Radev et al., 2004) developed an extractive multi-document summarizer, MEAD, which extracts a summary from multiple documents based on the document cluster centroid, position and first-sentence overlap. Recently, graph-based ranking methods, such as LexPageRank (Erkan and Radev, 2004) and TextRank (Mihalcea and Tarau, 2004), have been proposed for multi-document summarization. Similar to the original PageRank algorithm, these methods make use of similarity relationships between sentences and then rank sentences according to the "votes" or "recommendations" from their neighboring sentences.

Lin and Hovy (2000) first introduced topic signatures which are topic relevant terms for summarization. Afterwards, this technique was successfully used in a number of summarization systems (Hickl et al., 2007, Gupta and Nenkova et al., 2007). In order to improve sentence selection, we adopted the idea in a similar way to identify terms that tend to appear frequently in gene summaries and subsequently filter sentences that include none or few such terms.

Compared with newswire document summarization, much less attention has been paid to summarizing MEDLINE documents for genic information. Ling et al. (Ling et al., 2006 and 2007) presented an automatic gene summary generation system that constructs a summary based on six aspects of a gene, such as gene products, mutant phenotype, etc. In their system, sentences were ranked according to a) the relevance to each category (namely the aspect), b) the relevance to the document where they are from; and c) the position where sentences are located. Although the system performed well on a small group of genes (10~20 genes) from Flybase, their method relied heavily on high-quality training data that is often hard to obtain in practice.

Yang et al. reported a system (Yang et al., 2007 and 2009) that produces gene summaries by focusing on gene sets from microarray experiments. Their system first clustered gene set into functional related groups based on free text, Medical Subject Headings (MeSH®) and Gene Ontology (GO) features. Then, an extractive summary was generated for each gene following the Edmundson paradigm

[1] http://haydn.isi.edu/ROUGE/

(Edmundson, 1969). Yang et al. also presented evaluation results based on human ratings of eight gene summaries.

Another related work is the second task of Text REtrieval Conference [2] (TREC) 2003 Genomics Track. Participants in the track were required to extract GeneRIFs from MEDLINE abstracts (Hersh and Bhupatiraju, 2003). Many teams approached the task as a sentence classification problem using GeneRIFs in the Entrez database as training data (Bhalotia et al., 2003; Jelier et al., 2003). This task has also been approached as a single document summarization problem (Lu et al., 2006).

The gene summarization work presented here differs from the TREC task in that it deals with multiple documents. In contrast to the previously described systems for gene summarization, our approach has three novel features. First, we are able to summarize all aspects of gene-specific information as opposed to a limited number of predetermined aspects. Second, we exploit a lexical PageRank algorithm to establish similarity relationships between sentences. The importance of a sentence is based not only on the sentence itself, but also on its neighbors in a graph representation. Finally, we conducted an intrinsic evaluation on a large publicly available dataset. The gold standard assembled in this work makes it possible for comparisons between different gene summarization systems without human judgments.

3 Method

To determine if a sentence is extract worthy, we consider three different aspects: (1) the number of salient or informative words that are frequently used by human curators for writing gene summaries; (2) the relative importance of a sentence to be included in a gene summary; (3) the gene-specific information that is unique between different genes.

Specifically, we look for signature terms in handwritten summaries for the first aspect. Ideally, computer generated summaries should resemble handwritten summaries. Thus the terms used by human curators should also occur frequently in automatically generated summaries. In this regard, we use a method similar to Lin and Hovy (2000) to identify signature terms and subsequently use them to discard sentences that contain none or few such terms. For the second aspect, we adopt a lexical PageRank method to compute the sentence importance with a graph representation. For the last aspect, we treat each gene as having its own properties that distinguish it from others. To reflect such individual differences in the machine-generated summaries, we exploit a gene's GO annotations as a surrogate for its unique properties and look for their occurrence in abstract sentences.

Our gene summarization system consists of three components: a preprocessing module, a sentence ranking module, and a redundancy removal and summary generation module. Given a target gene, the preprocessing module retrieves corresponding MEDLINE abstracts and GO terms according to the gene2pubmed and gene2go data provided by Entrez Gene. Then the abstracts are split into sentences by the MEDLINE sentence splitter in the LingPipe[3] toolkit. The sentence ranking module takes these as input and first filters out some non-informative sentences. The remaining sentences are then scored according to a linear combination of the PageRank score and GO relevance score. Finally, a gene summary is generated after redundant sentences are removed. The system is illustrated in Figure 1 and is described in more detail in the following sections.

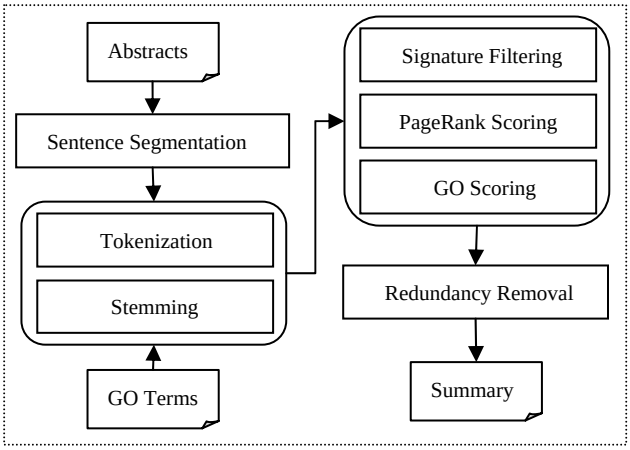


Figure 1. System overview

3.1 Signature Terms Extraction

There are signature terms for different topic texts (Lin and Hovy, 2000). For example, terms such as *eat*, *menu* and *fork* that occur frequently in a corpus may signify that the corpus is likely to be

[2] http://ir.ohsu.edu/genomics/

[3] http://alias-i.com/lingpipe/

about cooking or restaurants. Similarly, there are signature terms for gene summaries.

We use the Pearson's chi-square test (Manning and Schütze, 1999) to extract topic signature terms from a set of handwritten summaries by comparing the occurrence of terms in the handwritten summaries with that of randomly selected MEDLINE abstracts. Let R denote the set of handwritten summaries and $\tilde{R}$ denote the set of randomly selected abstracts from MEDLINE. The null hypothesis and alternative hypothesis are as follows:

$$H_0: \quad P(t_i \mid R) = p = P(t_i \mid \tilde{R})$$

$$H_1: \quad P(t_i \mid R) = p_1 \neq p_2 = P(t_i \mid \tilde{R})$$

The null hypothesis says that the term t_i appears in R and in $\tilde{R}$ with an equal probability and t_i is independent from R. In contrast, the alternative hypothesis says that the term t_i is correlated with R. We construct the following 2-by-2 contingency table:

	R	$\tilde{R}$
t_i	O_{11}	O_{12}
$\tilde{t}_i$	O_{21}	O_{22}

Table 2. Contingency table for the chi-square test.

where

O_{11}: the frequency of term t_i occurring in R;

O_{12}: the frequency of t_i occurring in $\tilde{R}$;

O_{21}: the frequency of term $\tilde{t}_i \neq t_i$ occurring in R;

O_{22}: the frequency of $\tilde{t}_i$ in $\tilde{R}$.

Then the Pearson's chi-square statistic is computed by

$$X^2 = \sum_{i,j=1}^{2} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

where O_{ij} is the observed frequency and E_{ij} is the expected frequency.

In our experiments, the significance level is set to 0.001, thus the corresponding chi-square value is 10.83. Terms with X^2 value above 10.83 would be selected as signature terms. In total, we obtained 1,169 unigram terms. The top ranked (by X^2 value) signature terms are listed in Table 3. Given the set of signature terms, sentences containing less than 3 signature terms are discarded. This parameter was determined empirically during the system development.

protein	member	receptor
gene	variant	isoform
encode	domain	alternative
family	splice	bind
transcription	subunit	involve

Table 3. A sample of unigram topic signature terms.

3.2 Lexical PageRank Scoring

The lexical PageRank algorithm makes use of the similarity between sentences and ranks them by how similar a sentence is to all other sentences. It originates from the original PageRank algorithm (Page et al., 1998) that is based on the following two hypotheses:

(1) A web page is important if it is linked by many other pages.

(2) A web page is important if it is linked by important pages.

The algorithm views the entire internet as a large graph in which a web page is a vertex and a directed edge is connected according to the linkage. The salience of a vertex can be computed by a random walk on the graph. Such graph-based methods have been widely adapted to such Natural Language Processing (NLP) problems as text summarization and word sense disambiguation. The advantage of such graph-based methods is obvious: the importance of a vertex is not only decided by itself, but also by its neighbors in a graph representation. The random walk on a graph can imply more global dependence than other methods. Our PageRank scoring method consists of two steps: constructing the sentence graph and computing the salience score for each vertex of the graph.

Let $S = \{s_i \mid 1 \leq i \leq N\}$ be the sentence collection containing all the sentences to be summarized. According to the vector space model (Salton et al., 1975), each sentence s_i can be represented by a vector $\vec{s}_i$ with each component being the weight of a term in s_i. The weight associated with a term w is calculated by $tf(w) * isf(w)$, where $tf(w)$ is the frequency of the term w in sentence s_i and $isf(w)$

is the inverse sentence frequency[4] of term w: $isf(w) = 1 + \log(N / n_w)$, where N is the total number of sentences in S and n_w is the number of sentences containing w. The similarity score between two sentences is computed using the inner product of the corresponding sentence vectors, as follows:

$$sim(s_i, s_j) = \frac{\vec{s}_i \cdot \vec{s}_j}{\| \vec{s}_i \| \cdot \| \vec{s}_j \|}$$

Taking each sentence as a vertex, and the similarity score as the weight of the edge between two sentences, a sentence graph is constructed. The graph is fully connected and undirected because the similarity score is symmetric.

The sentence graph can be modeled by an adjacency matrix $\mathbf{M}$, in which each element corresponds to the weight of an edge in the graph. Thus $\mathbf{M} = [M_{ij}]_{N \times N}$ is defined as:

$$M_{ij} = \begin{cases} \dfrac{\vec{s}_i \cdot \vec{s}_j}{\| \vec{s}_i \| \cdot \| \vec{s}_j \|}, & if\ i \neq j \\ 0, & otherwise \end{cases}$$

We normalize the row sum of matrix $\mathbf{M}$ in order to assure it is a stochastic matrix such that the PageRank iteration algorithm is applicable. The normalized matrix is:

$$\tilde{M}_{ij} = \begin{cases} M_{ij} \Big/ \sum_{j=1}^{N} M_{ij}, & if \sum_{j=1}^{N} M_{ij} \neq 0 \\ 0, & otherwise \end{cases}.$$

Using the normalized adjacency matrix, the salience score of a sentence s_i is computed in an iterative manner:

$$score(s_i) = d \cdot \sum_{j=1}^{N} score(s_j) \cdot \tilde{M}_{ji} + \frac{(1-d)}{N}$$

where d is a damping factor that is typically between 0.8 and 0.9 (Page et al., 1998).

If we use a column vector p to denote the salience scores of all the sentences in S, the above equation can be written in a matrix form as follows:

$$p = [d \cdot \mathbf{M}^T + (1-d) \cdot \mathbf{U}] \cdot p$$

where $\mathbf{U}$ is a square matrix with all elements being equal to $1/N$. The component $(1-d) \cdot \mathbf{U}$ can be considered as a smoothing term which adds a small probability for a random walker to jump from the current vertex to any vertex in the graph. This guarantees that the stochastic transition matrix for iteration is irreducible and aperiodic. Therefore the iteration can converge to a stable state.

In our implementation, the damping factor d is set to 0.85 as in the PageRank algorithm (Page et al., 1998). The column vector $\boldsymbol{p}$ is initialized with random values between 0 and 1. After the algorithm converges, each component in the column vector $\boldsymbol{p}$ corresponds to the salience score of the corresponding sentence. This score is combined with the GO relevance score to rank sentences.

3.3 GO Relevance Scoring

Up to this point, our system considers only gene-independent features, in both sentence filtering and PageRank-based sentence scoring. These features are universal across different genes. However, each gene is unique because of its own functional and structural properties. Thus we seek to include gene-specific features in this next step.

The GO annotations provide one kind of gene-specific information and have been shown to be useful for selecting GeneRIF candidates (Lu et al., 2006). A gene's GO annotations include descriptions in three aspects: molecular function; biological process; and cellular component. For example, the human gene AANAT (gene ID 15 in Entrez Gene) is annotated with the GO terms in Table 4.

GO ID	*GO term*
GO:0004059	aralkylamine N-acetyltransferase activity
GO:0007623	circadian rhythm
GO:0008152	metabolic process
GO:0008415	acyltransferase activity
GO:0016740	transferase activity

Table 4. GO terms for gene AANAT

The GO relevance score is computed as follows: first, the GO terms and the sentences are both stemmed and stopwords are removed. For example, the GO terms in Table 4 are processed into a set of stemmed words: *aralkylamin, N, acetyltransferas, activ, circadian, rhythm, metabol, process, acyltransferas* and *transferas*.

[4] *Isf* is equivalent to *idf* if we view each sentence as a document.

Second, the total number of occurrence of the GO terms appearing in a sentence is counted. Finally, the GO relevance score is computed as the ratio of the total occurrence to the sentence length. The entire process can be illustrated by the following pseudo codes:

```
1 tokenize and stem the GO terms;
2 tokenize and stem all the sentences, remove stop
  words;
3 for each sentence s_i , i = 1,...,N
GOScore(s_i) = 0
for each word w in s_i
if w in the GO term set
GOScore(s_i) ++
end if
end for
    GOScore(s_i) = GOScore(s_i) / length(s_i)
end for
```

where $length(s_i)$ is the number of distinct non-stop words in s_i. For each sentence s_i, the GO relevance score is combined with the PageRank score to get the overall score (α is a weight parameter between 0 and 1; see Section 4.2 for discussion):

$$score(s_i) = \alpha \cdot PRScore(s_i) + (1-\alpha) \cdot GOScore(s_i)$$

3.4 Redundancy Removal

A good summary contains as much diverse information as possible for a gene, while with as little redundancy as possible. For many well-studied genes, there are thousands of relevant papers and much information is redundant. Hence it is necessary to remove redundant sentences before producing a final summary.

We adopt the diversity penalty method (Zhang et al., 2005; Wan and Xiao, 2007) for redundancy removal. The idea is to penalize the candidate sentences according to their similarity to the ones already selected. The process is as follows:

(1) Initialize two sets, $A = \phi$, $B = \{s_i \mid i = 1, 2, ..., K\}$ containing all the extracted sentences;

(2) Sort the sentences in B by their scores in descending order;

(3) Suppose s_i is the top ranked sentence in B, move it from B to A. Then we penalize the remaining sentences in B as follows:

For each sentence s_j in B, $j \neq i$

$$Score(s_j) = Score(s_j) - \omega \cdot sim(s_j, s_i) \cdot Score(s_i)$$

where $\omega > 0$ is the penalty degree factor, $sim(s_j, s_i)$ is the similarity between s_i and s_j.

(4) Repeat steps 2 and 3 until enough sentences have been selected.

4 Results and Discussion

4.1 Evaluation Metrics

Unlike the newswire summarization, there are no gold-standard test collections available for evaluating gene summarization systems. The two previous studies mentioned in Section 2 both conducted *extrinsic* evaluations by asking human experts to rate system outputs. Although it is important to collect direct feedback from the users, involving human experts makes it difficult to compare different summarization systems and to conduct large-scale evaluations (both studies evaluated nothing but a small number of genes). In contrast, we evaluated our system intrinsically on a much larger dataset consisting of 7,294 human genes, each with a pre-existing handwritten summary downloaded from the NCBI's FTP site[5].

The handwritten summaries were used as reference summaries (i.e. a gold standard) to compare with the automatically generated summaries. Although the length of reference summaries varies, the majority of these summaries contain 80 to 120 words. To produce a summary of similar length, we decided to select five sentences consisting of about 100 words.

For the intrinsic evaluation of a large number of summaries, we made use of the ROUGE metrics that has been widely used in automatic evaluation of summarization systems (Lin and Hovy, 2003; Hickl et al., 2007). It provides a set of evaluation metrics to measure the quality of a summary by counting overlapping units such as n-grams or word sequences between the generated summary and its reference summary.

[5] ftp://ftp.ncbi.nih.gov/gene/DATA/ASN_BINARY/

We computed three ROUGE measures for each summary, namely ROUGE-1 (unigram based), ROUGE-2 (bigram based) and ROUGE-SU4 (skip-bigram and unigram) (Lin and Hovy, 2003). Among them, ROUGE-1 has been shown to agree most with human judgments (Lin and Hovy, 2003). However, as biomedical concepts usually contain more than one word (e.g. transcription factor), ROUGE-2 and ROUGE-SU4 scores are also important for assessing gene summaries.

4.2 Determining parameters for best performance

The two important parameters in our system – the linear coefficient α for the combination of PageRank and GO scores and the diversity penalty degree factor ω in redundancy removal – are investigated in detail on a collection of 100 randomly selected genes. First, by setting α to values from 0 to 1 with an increment of 0.1 while holding ω steady at 0.7, we observed the highest ROUGE-1score when α was 0.8 (Figure 2). This suggests that the two scores (i.e. PageRank and GO score) complement to each other and that the PageRank score plays a more dominating role in the summed score. Next, we varied ω gradually from 0 to 5 with an increment of 0.25 while holding α steady at 0.75.The highest ROUGE-1 score was achieved when ω was 1.3 (Figure 3). For ROURE-2, the best performance was obtained when α was 0.7 and ω was 0.5. In order to balance ROUGE-1 and ROUGE-2 scores, we set α to 0.75 and ω to 0.7 for the remaining experiments.

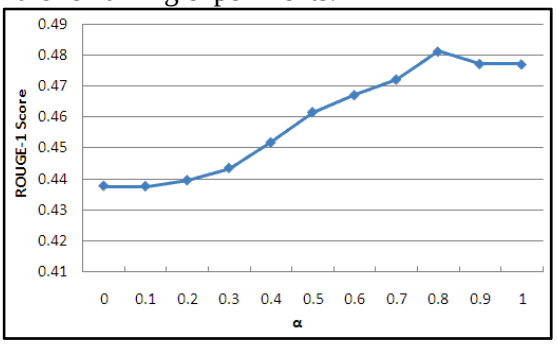


Figure 2. The blue line represents the changes in ROUGE-1 scores with different values of α while ω is held at 0.7.

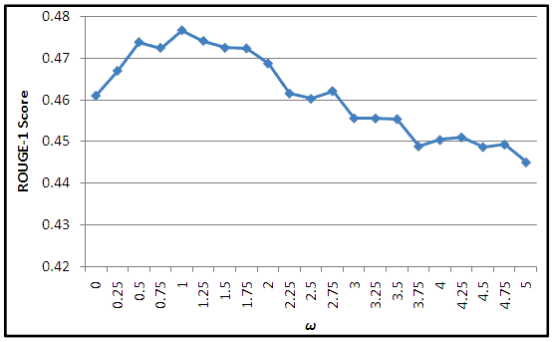


Figure 3. The blue line represents the changes in ROUGE-1 scores with different values of ω while α is held at 0.75.

4.3 Comparison with other methods

Because there are no publicly available gene summarization systems, we compared our system with two baseline methods. The first is a well known publicly available summarizer - MEAD (Radev et al., 2004). We adopted the latest version of MEAD 3.11 and used the default setting in MEAD that extracts sentences according to three features: centroid, position and length. The second baseline extracts different sentences randomly from abstracts. Comparison results are shown in the following table:

System	*ROUGE-1*	*ROUGE-2*	*ROUGE-SU4*
Our System	0.4725	0.1247	0.1828
MEAD	0.3890	0.0961	0.1449
Random	0.3434	0.0577	0.1091

Table 5. Systems comparison on 7,294 genes.

As shown in Table 5, our system significantly outperformed the two baseline systems in all three ROUGE measures. Furthermore, larger performance gains are observed in ROUGE-2 and ROUGE-SU4 than in ROUGE-1. This is because many background words (e.g. *gene*, *protein* and *enzyme*) also appeared frequently as unigrams in randomly selected summaries.

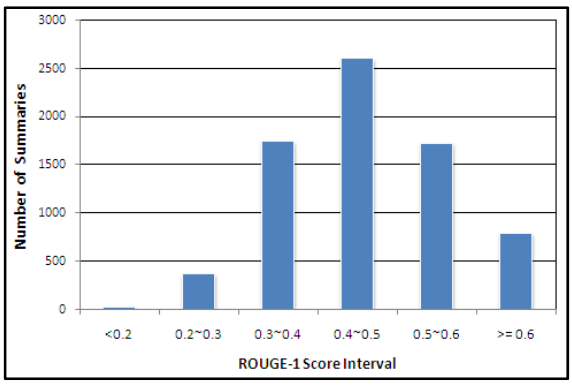


Figure 4. ROUGE-1 score distribution

In Figure 4, we show that the majority of the summaries have a ROUGE-1 score greater than 0.4. Our further analysis revealed that almost half summaries with a low score (smaller than 0.3) either lacked sufficient relevant abstracts, or the reference summary was too short or too long. In either case, only few overlapping words can be found when comparing the generated gene summary with the reference. The statistics for low ROUGE-1 score are listed in Table 6. We also note that almost half of the summaries that have low ROUGE-1 scores were due to other causes: mostly, machine generated summaries differ from human summaries in that they describe different functional aspects of the same gene product. Take the gene TOP2A (ID: 7153) for example. While both summaries (handwritten and machine generated) focus on its encoded protein *DNA topoisomerase*, the handwritten summary describes the chromosome location of the gene whereas our algorithm selects statements about its gene expression when treated with a chemotherapy agent. We plan to investigate such differences further in our future work.

Causes for Low Score	*Number of genes*
Few (⩽10) related abstracts	106
Short reference summary (< 40 words)	27
Long reference summary (> 150 words)	76
Other	198
Total	407

Table 6. Statistics for low ROUGE-1 scores (<0.3)

4.4 Results on various summary length

Figure 5 shows the variations of ROUGE scores as the summary length increases. At all lengths and for both ROUGE-1 and ROUGE-2 measures, our proposed method performed better than the two baseline methods. By investigating the scores of different summary lengths, it can be seen that the advantage of our method is greater when the summary is short. This is of great importance for a summarization system as ordinary users typically prefer short content for summaries.

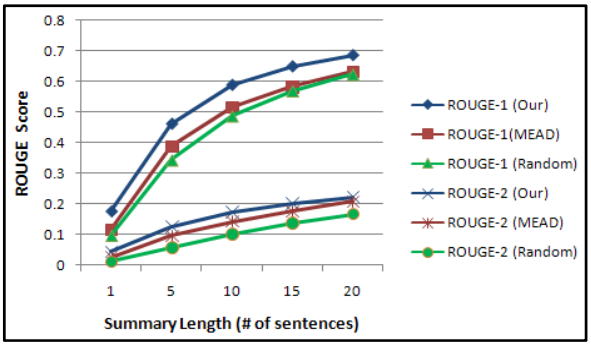


Figure 5. Score variation for different summary length

5 Conclusions and Future Work

In this paper we have presented a system for generating gene summaries by automatically finding extract-worthy sentences from the biomedical literature. By using the state-of-the-art summarization techniques and incorporating gene specific annotations, our system is able to generate gene summaries more accurately than the baseline methods. Note that we only evaluated our system for human genes in this work. More summaries are available for human genes than other organisms, but our method is organism-independent and can be applied to any other species.

This research has implications for real-world applications such as assisting manual database curation or updating existing gene records. The ROUGE scores in our evaluation show comparable performance to those in the newswire summarization (Hickl et al., 2007). Nonetheless, there are further steps necessary before making our system output readily usable by human curators. For instance, human curators are generally in favor of sentences presented in a coherent order. Thus, information-ordering algorithms in multi-document summarization need to be investigated. We also plan to study the guidelines and scope of the curation process, which may provide additional important heuristics to further refine our system output.

Acknowledgments

The work is supported by NSFC project No. 60803075, Chinese 973 Project No. 2007CB311003. ZL is supported by the Intramural Program of the National Institutes of Health. The authors are grateful to W. John Wilbur and G. Craig Murray for their help on the early version of this manuscript.

References

W. A. Baumgartner, B. K. Cohen, L. M. Fox, G. Acquaah-Mensah, L. Hunter. 2007. Manual Curation Is Not Sufficient for Annotation of Genomic Databases. Bioinformatics, Vol. 23, No. 13. (July 2007), pp. i41-48.

G. Bhalotia, P. I. Nakov, A. S. Schwartz and M. A. Hearst, BioText Team Report for the TREC 2003 Genomics Track. In Proceedings of TREC 2003.

B. Boguraev and C. Kennedy. 1997. Salience-based Content Characterization of Text Documents. In Proceedings of Workshop on Intelligent Scalable Text Summarization (ACL97/EACL97), pp. 2-9.

J. Carbonell and J. Goldstein. 1998. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In ACM SIGIR, pages 335–336, August.

H. P. Edmundson. 1969. New Methods in Automatic Extracting. Journal of the ACM (JACM) archive Volume 16, Issue 2 (April 1969) Pages: 264 – 285.

G. Erkan and D. R. Radev. 2004. LexPageRank: Prestige in Multi-Document Text Summarization. In Proceedings of 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain.

S. Gupta, A.Nenkova and D.Jurafsky. 2007. Measuring Importance and Query Relevance in Topic-focused Multi-document Summarization. Proceedings of ACL 2007 short papers, Prague, Czech Republic.

W. Hersh and R. T. Bhupatiraju. 2003. TREC Genomics track Overview. In Proceedings of TheTwelfth Text REtrieval Conference, 2003.

A. Hickl, K. Roberts and F. Lacatusu. 2007. LCC's GISTexter at DUC 2007: Machine Reading for Update Summarization.

R. Jelier, M. Schuemie, C. Eijk, M. Weeber, E. Mulligen, B. Schijvenaars, B. Mons, J. Kors. Searching for geneRIFs: Concept-based Query Expansion and Bayes Classification. In Proceedings of TREC 2003.

C. Lin and E. Hovy. 2000. The Automated Acquisition of Topic Signatures for Text Summarization. In Proceedings of the COLING Conference.

C. Lin and E. Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics. In HLT-NAACL, pages 71–78.

X. Ling, J. Jiang, X. He, Q. Mei, C. Zhai and B. Schatz. 2006. Automatically Generating Gene Summaries from Biomedical Literature. Proceedings of the Pacific Symposium on Biocomputing 2006.

X. Ling, J. Jiang, X. He, Q. Mei, C. Zhai and B. Schatz. 2007. Generating Gene Summaries from Biomedical Literature: A Study of Semi-Structured Summarization. Information Processing and Management 43, 2007, 1777-1791.

Z. Lu, K. B. Cohen and L. Hunter. 2006. Finding GeneRIFs via Gene Ontology Annotations. Pac Symp-Biocomput. 2006:52-63.

C. Manning and H. Schütze. 1999. Foundations of Statistical Natural Language Processing. Chapter 5, MIT Press. Cambridge, MA: May 1999.

K. R. McKeown and D. R. Radev. 1995. Generating Summaries of Multiple News Articles. In Proceedings, ACM Conference on Research and Development in Information Retrieval SIGIR'95, pages 74–82.

R. Mihalcea and P. Tarau. TextRank: Bringing Order into Texts, in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain, July 2004.

M. Newman. 2003. The Structure and Function of Complex Networks. SIAM Review 45.167–256 (2003).

L. Page, S. Brin, R. Motwani and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford University, Stanford, CA, 1998.

D. R. Radev, H. Jing, M. Stys and D. Tam. 2004. Centroid-based Summarization of Multiple Documents. Information Processing and Management, 40:919–938.

G. Salton, A. Wong, and C. S. Yang. 1975. A Vector Space Model for Automatic Indexing. Communications of the ACM, vol. 18, nr.11, pages 613–620.

S. Teufel and M. Moens. 1997. Sentence Extraction as a Classification Task. Workshop 'Intelligent and scalable Text summarization', ACL/EACL 1997.

X. Wan and J. Xiao. 2007. Towards a Unified Approach Based on Affinity Graph to Various Multi-document Summarizations. ECDL 2007: 297-308.

J. Yang, A. M. Cohen, W. Hersh. Automatic Summarization of Mouse Gene Information by Clustering and Sentence Extraction from MEDLINE Abstracts. AMIA 2007 Annual Meeting. Nov. 2007 Chicago, IL.

J. Yang, A. M. Cohen, W. Hersh. 2008. Evaluation of a Gene Information Summarization System by Users During the Analysis Process of Microarray Datasets. In BMC Bioinformatics 2009 10(Suppl 2):S5.

B. Zhang, H. Li, Y. Liu, L. Ji, W. Xi, W. Fan, Z. Chen, W. Ma. 2005. Improving Web Search Results Using Affinity Graph. The 28th Annual International ACM SIGIR Conference (SIGIR'2005), August 2005.

Incorporating GENETAG-style annotation to GENIA corpus

Tomoko Ohta[*] and **Jin-Dong Kim**[*] and **Sampo Pyysalo**[*] and **Yue Wang**[*] and **Jun'ichi Tsujii**[*†‡]
[*]Department of Computer Science, University of Tokyo, Tokyo, Japan
[†]School of Computer Science, University of Manchester, Manchester, UK
[‡]National Centre for Text Mining, University of Manchester, Manchester, UK
{okap,jdkim,smp,wangyue,tsujii}@is.s.u-tokyo.ac.jp

1 Introduction

Proteins and genes are the most important entities in molecular biology, and their automated recognition in text is the most widely studied task in biomedical information extraction (IE). Several corpora containing annotation for these entities have been introduced, GENIA (Kim et al., 2003; Kim et al., 2008) and GENETAG (Tanabe et al., 2005) being the most prominent and widely applied. While both aim to address protein/gene annotation, their annotation principles differ notably. One key difference is that GENETAG annotates the conceptual entity, gene, which is often associated with a function, while GENIA concentrates on the physical forms of gene, i.e. protein, DNA and RNA. The difference has caused serious problems relating to the compatibility and comparability of the annotations. In this work, we present an extension of GENIA annotation which integrates GENETAG-style gene annotation. The new version of the GENIA corpus is the first to bring together these two types of entity annotation.

2 GGP Annotation

Gene is the basic unit of heredity, which is encoded in the coding region of *DNA*. Its physical manifestations as *RNA* and Protein are often called its products. In our view of these four entity types, gene is taken as an abstract entity whereas protein, DNA and RNA are physical entities. While the three physical entity types are disjoint, the abstract concept, gene, is defined from a different perspective and is realized in, not disjoint from, the physical entity types.

The latest public version of GENIA corpus (hereafter "old corpus") contains annotations for gene-related entities, but they are classified into only physical entity types: Protein, DNA and RNA. The corpus revisions described in this work are two-fold. First, annotation for the abstract entity, gene, were added (Table 1, GGP). To emphasize the characteristics of the new entity type, which does not distinguish a gene and its products, we call it GGP (gene or gene product). Second, the addition of GGP annotation triggered large-scale removal of Protein, DNA and RNA annotation instances for cases where the physical form of the gene was not referred to (Due to space limitations, we omit RNA from now on). The time cost involved with this revision was approximately 500 person-hours.

	Protein	DNA	RNA	GGP
Old Annotation	21,489	8,653	876	N/A
New Annotation	15,452	7,872	863	12,272

Table 1: Statistics on annotation for gene-related entities

3 Quality Assessment

To measure the effect of revision, we performed NER experiments with old and new annotation (Tables 2 and 3). We split the corpus into disjoint 90% and 10% parts for use in training and test, respectively. We used the BANNER (Leaman and Gonzalez, 2008) NE tagger and created a separate single-class NER problem for each entity type.

In the old annotation, consistency is moderate for protein (77.70%), while DNA is problematic (58.03%). The new GGP annotation has been achieved in a fairly consistent way (81.44%). However, the removal of annotation for entities previously marked as protein or DNA had opposite effects on the two: better performance for DNA (64.06%),

Proceedings of the Workshop on BioNLP, pages 106–107,
Boulder, Colorado, June 2009.

	Precision	Recall	F-score
Protein	80.78	74.84	77.70
DNA	64.90	52.48	58.03

Table 2: NER performance before GGP annotation

	Precision	Recall	F-score
Protein	71.20	56.61	63.08
DNA	69.59	59.35	64.06
GGP	86.86	76.65	81.44
Protein+	83.22	78.20	80.63

Table 3: NER performance after GGP annotation

Phosphorylation		Gene_expression	
GGP_in_protein	70%	GGP_abstract	34%
Protein	25%	Protein	24%
GGP_abstract	3%	GGP_in_Protein	17%
Peptide	1%	GGP_in_DNA	9%

Table 4: Distribution of theme entity types in GENIA

implying annotation consistency improved with the removals, but worse for Protein (63.08%).

We find the primary explanation for this effect in the statistics in Table 1: in the revision, a large number of protein annotations (6,037) but only a small number of DNA annotations (780) were replaced with GGP. To distinguish such GGPs from those embedded in Protein or DNA annotations, we call them "abstract" GGPs, as they appear in text without information on their physical form. Nevertheless, in the old annotation, they had to be annotated as either protein or DNA, which might have caused inconsistent annotation. However, the statistics show a clear preference for choosing Protein over DNA. The radical drop of performance in protein recognition can then be explained in part as a result of removing this systematic preference.

Aside from the discussion on whether the preference is general or specific, we interpret the preference as a need for "potential" proteins to be retrieved together with "real" proteins, which was answered by the old protein annotation. To reproduce this class in the new annotation, we added abstract GGPs to the Protein annotation and performed an NER experiment. The result (Table 3, Protein+) shows a clear improvement over the comparable result for the old protein annotation.

In conclusion, we argue, the revision of the GENIA annotation, in addition to introducing a new entity class, has led to a significant improvement of overall consistency.

4 Discussion

Although there are already corpora such as GENETAG with annotation similar to GGPs, we expect this newly introduced class of annotation to support existing annotations of GENIA, such as event and co-reference annotation, opening up new possibilities for application. The quality of entity annotation should be closely related to that of other semantic annotation, e.g. events. For example, the event type Phosphorylation is about a change on physical entities, e.g. proteins and peptides, and as such, it is expected that themes of these events would be physical entities. On the other hand, the event type Gene_expression is about the manifestation of an abstract entity (gene) as a physical entity (protein) and would thus be expected to involve both abstract and physical entities. Statistics from GENIA (Table 4) show that the theme selection made in event annotation well reflects these characteristics of the two event types. The observation suggests that there is a good likelihood that improvement of the entity annotation can be further transferred to other semantic annotation, which is open for future work.

Acknowledgments

This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan) and Genome Network Project (MEXT, Japan).

References

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl. 1):i180–i182.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from lterature. *BMC Bioinformatics*, 9(1):10.

R. Leaman and G. Gonzalez. 2008. Banner: an executable survey of advances in biomedical named entity recognition. *Pacific Symposium on Biocomputing*, pages 652–663.

Lorraine Tanabe, Natalie Xie, Lynne Thom, Wayne Matten, and W John Wilbur. 2005. Genetag: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, 6(Suppl 1):S3.

User-Driven Development of Text Mining Resources for Cancer Risk Assessment

Lin Sun, Anna Korhonen
University of Cambridge
Computer Laboratory
15 JJ Thomson Avenue
Cambridge CB3 0GD, UK
ls418,alk23@cl.cam.ac.uk

Ilona Silins, Ulla Stenius
Institute of Environmental Medicine
Karolinska Institutet
S-17177, Stockholm
Sweden
ilona.silins,ulla.stenius@ki.se

Abstract

One of the most neglected areas of biomedical Text Mining (TM) is the development of systems based on carefully assessed user needs. We investigate the needs of an important task yet to be tackled by TM — Cancer Risk Assessment (CRA) — and take the first step towards the development of TM for the task: identifying and organizing the scientific evidence required for CRA in a taxonomy. The taxonomy is based on expert annotation of 1297 MEDLINE abstracts. We report promising results with inter-annotator agreement tests and automatic classification experiments, and a user test which demonstrates that the resources we have built are well-defined, accurate, and applicable to a real-world CRA scenario. We discuss extending and refining the taxonomy further via manual and machine learning approaches, and the subsequent steps required to develop TM for the needs of CRA.

1 Introduction

Biomedical Text Mining (TM) has become increasingly popular due to the pressing need to provide access to the tremendous body of texts available in biomedical sciences. Considerable progress has been made in the development of basic resources (e.g. ontologies, annotated corpora) and techniques (e.g. Information Retrieval (IR), Information Extraction (IE)) in this area, and research has began to focus on increasingly challenging tasks, e.g. summarization and the discovery of novel information in biomedical literature (Hunter and Cohen 2006, Ananiadou et al. 2006, Zweigenbaum et al. 2007).

In recent past, there has been an increasing demand for research which is driven by actual user needs rather than technical developments (Zweigenbaum et al. 2007). Shared tasks (e.g. BioCreative and the TREC Genomics track) targeting the workflow of biomedical researchers have appeared along with studies exploring the TM needs of specific tasks (Karamanis et al. 2008, Demaine et al. 2006). However, the understanding of user needs is still one of the neglected areas of BIO-TM, and further user-centered evaluations and systems grounded in real-life tasks are required to determine which tools and services are useful (Cohen et al. 2008).

We investigate the user needs of a challenging task yet to be tackled by TM but identified as an important potential application for it (Lewin et al. 2008): Cancer Risk Assessment (CRA). Over the past years, CRA has become increasingly important as the link between environmental chemicals and cancer has become evident. It involves examining published evidence to determine the relationship between exposure to a chemical and the likelihood of developing cancer from that exposure (EPA, 2005). Performed manually by experts in health related institutions worldwide, CRA requires searching, locating and interpreting information in biomedical journal articles. It can be extremely time-consuming because the data for a single carcinogen may be scattered across thousands of articles.

Given the exponentially growing volume of biomedical literature and the rapid development of molecular biology techniques, the task is now getting too challenging to manage via manual means. From the perspective of BIO-TM, CRA is an excellent example of real-world task which could greatly benefit from a dedicated TM tool. However, the development of a truly useful tool requires careful investigation of risk assessors needs.

Proceedings of the Workshop on BioNLP, pages 108–116,
Boulder, Colorado, June 2009.

This paper reports our investigation of the user needs of CRA and the creation of basic TM resources for the task. Expanding on our preliminary experiments (Lewin et al. 2008), we present a taxonomy which specifies the scientific evidence needed for CRA at the level of detail required for TM. The taxonomy is based on expert annotation of a corpus of 1297 MEDLINE abstracts. We report promising results with inter-annotator agreement tests, automatic classification of corpus data into taxonomy classes, and a user test in a near real-world CRA scenario which shows that the taxonomy is highly accurate and useful for practical CRA. We discuss refining and extending it further via manual and machine learning approaches, and the subsequent steps required to develop TM for the needs of CRA.

2 User Needs of Cancer Risk Assessment

We interviewed 14 experienced risk assessors working for a number of authorities in Sweden[1] asking a range of questions related to different aspects of their work. The risk assessors described the following steps of CRA: (1) identifying the journal articles relevant for CRA of the chemical in question, (2) identifying the scientific evidence in these articles which help to determine whether/how the chemical causes cancer, (3) classifying and analysing the resulting (partly conflicting) evidence to build the toxicological profile for the chemical, and (4) preparing the risk assessment report. These steps are conducted manually, relying only on standard literature search engines (e.g. PubMed) and word processors.

The average time required for CRA of a single chemical was reported to be two years when done (as usual) on a part time basis. Risk assessors were unanimous about the need to increase productivity to meet the current CRA demand. They reported that locating and classifying the scientific evidence in literature is the most time consuming part of their work and that a tool capable of assisting it and ensuring that all the potentially relevant evidence is found would be particularly helpful.

It became clear that a prerequisite for the development of such a tool would be an extensive specification of the scientific evidence used for CRA. This evidence — which forms the basis of all the subsequent steps of CRA — is described in the guideline documents of major international CRA agencies, e.g. European Chemicals Agency (ECHA, 2008) and the United States Environmental Protection Agency (EPA, 2005). However, although these documents constitute the main reference material in CRA, they cover the main types of evidence only, do not specify the evidence at the level of detail required for comprehensive data gathering, and are not updated regularly (i.e. do not incorporate the latest developments in biomedical sciences). The risk assessors admitted that rather than relying on these documents, they rely on their experience and expert knowledge when looking for the evidence. We decided that our starting point should be to compose a more adequate specification of the scientific evidence needed for CRA.

3 Cancer Risk Assessment Taxonomy

We recruited three experienced risk assessors to help construct the resources described in sections below: (i) a representative corpus of CRA literature for parts of hazard identification (i.e. the assessment of whether a chemical is capable of causing cancer), (ii) a tool for expert annotation of the corpus, (iii) an annotated corpus, and (iv) a taxonomy which classifies and organizes the scientific evidence discovered in the corpus.

3.1 CRA corpus

Various human, animal (*in vivo*), cellular (*in vitro*) and other mechanistic data provide evidence for hazard identification and the assessment of the Mode of Action (MOA) (i.e. the sequence of key events that result in cancer formation, e.g. mutagenesis and increased cell proliferation) in CRA. The experts selected eight chemicals which are (i) well-researched using a range of scientific tests and (ii) represent the two most frequently used MOAs – *genotoxic* and *non-genotoxic*[2]. 15 journals were identified which are used frequently for CRA and jointly provide a good coverage of relevant scientific evidence (e.g. Cancer Research, Chemico-biological Interaction, Mutagenesis, Toxicological Sciences). From these

[1] Institute of Environmental Medicine at Karolinska Institutet, Swedish Chemical Inspectorate, Scientific Committee on Occupational Exposure Limits (EU), Swedish Criteria Group.

[2] Chemicals acting by a genotoxic MOA interact with DNA, while chemicals acting by a nongenotoxic MOA induce cancer without interfering directly with DNA.

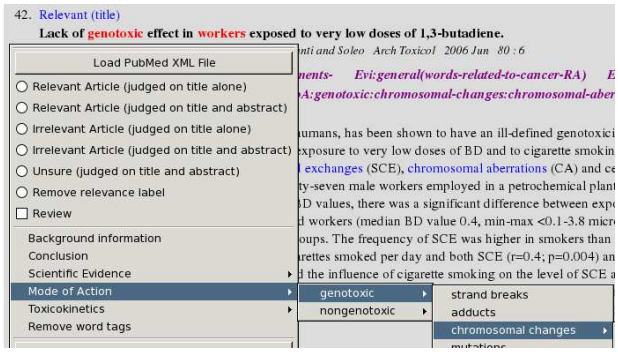


Figure 1: Screenshot of the annotation tool

journals, all the PubMed abstracts from 1998-2008 which include one of the 8 chemicals were downloaded. The resulting corpus of 1297 abstracts is distributed per chemical as shown in Table 1.

3.2 Annotation tool

Risk assessors typically (i) read each abstract retrieved by PubMed to determine its relevance for CRA, and (ii) classify each relevant abstract based on the type of evidence it provides for CRA. We extended the tool designed for expert annotation of abstracts in our earlier work (Lewin et al. 2008) so that imitates this process as closely as possible.

The tool provides two types of functionality. The first enables the experts to classify abstracts as relevant, irrelevant or unsure. The second enables them to annotate such keywords (words or phrases) in abstracts and their titles which indicate the scientific evidence relevant for the task. Keyword annotation was chosen because the experts found it intuitive, it did not require linguistic training, and it specifies the scientific evidence more precisely than larger spans of text.

Initially a very shallow taxonomy (including only human, animal, and cellular data) and the two types of MOA was integrated inside the tool. This was gradually extended as the annotation progressed. The tool permits annotating any number of relevant keywords in the abstracts, attaching them to any class in the taxonomy, and classifying the same text in more than one way. It was implemented inside the familiar Mozilla Firefox browser using its extension facility. A screenshot illustrating the tool is provided in Figure 1.

3.3 Annotation

Given a set of initial guidelines agreed by the experts, one of the experts annotated a subset of the corpus, the other two evaluated the result, disagreements were then discussed, and the guidelines were improved where needed. This process (crucial for maintaining quality) was repeated several times. The guidelines described below are the final result of this work.

3.3.1 Relevance annotation

An abstract is classified as (i) relevant when it (or its title) contains evidence relevant for CRA and (ii) irrelevant when it (or its title) contains no evidence or contains ”negative” evidence (e.g. diseases or endpoints unrelated to cancer). Abstracts containing vague, conflicting or complex evidence (e.g. studies on chemicals in complex mixtures) or evidence whose association with cancer is currently unclear were dealt on case by case basis. All the potentially relevant abstracts were included for further assessment as not to lose data valuable for CRA.

The experts annotated the 1297 abstracts in the corpus. 89.4% were classified as relevant, 10.1% as irrelevant, and 0.5% as unsure. We used the Kappa statistics (Cohen 1960) to measure inter-annotator agreement on unseen data which two experts annotated independently. 208 abstracts were selected randomly from the 15 journals and from 16 journals likely to be irrelevant for CRA. The latter were included to make the task harder as the proportion of relevant abstracts was high in our corpus. Our Kappa result is 0.68 — a figure which indicates substantial agreement (Landis and G.Koch 1977).

The experts disagreed on 24 (11.5% of the) abstracts. Half of the disagreements are due to one of the annotators failing to notice relevant evidence. Such cases are likely to decrease when annotators gain more experience. The other half are caused by vague or conflicting evidence. Many of these could be addressed by further development of guidelines.

3.3.2 Keyword annotation

Keyword annotation focussed on the types of scientific evidence experts typically look for in CRA: **carcinogenic activity** (human, animal, cellular, and other mechanistic data), **Mode of Action (MOA)** (data for a specific MOA type — genotoxic or non-

Chemical	Retrieved	Relevant
1,3-butadiene	195	187
phenobarbital	270	240
diethylnitrosamine	221	214
diethylstilbestrol	145	110
benzoapyrene	201	192
fumonisin	80	70
chloroform	96	84
styrene	162	132
Total	**1297**	**1164**

Table 1: Total of abstracts per chemical

genotoxic), and relevant parts of **toxicokinetics** (e.g. metabolic activation). The experts annotated the keywords which they considered as the most important and which jointly identify the types of scientific data offered by the abstract. They focussed on new (rather than previously published) data on the chemical in question.

All the 1164 abstracts deemed relevant were annotated. A total of 1742 unique keywords were identified, both simple nouns and complex nominals / phrases. Figure 1 shows an example of an annotated abstract where the keyword *chromosomal aberrations* is identified as evidence for genotoxic MOA. Since the experts were not required to annotate every relevant keyword, calculating inter-annotator agreement was not meaningful. However, the keyword annotation was evaluated jointly with taxonomy classification (the following section).

3.4 The taxonomy and the resulting corpus

During keyword annotation, the initial taxonomy was extended and refined with new classes and class members. The resulting taxonomy relies solely on expert knowledge. Experts were merely advised on the main principles of taxonomy creation: the classes should be conceptually coherent and their hierarchical organization should be in terms of coherent sub- and superordinate relations.

The taxonomy contains three top level classes: 1) Carcinogenic activity (*CA*), 2) Mode of Action (*MOA*) and 3) Toxicokinetics (*TOX*). 1) and 2) are organized by TYPE-OF relations (*leukemia* is a type of carcinogenic evidence) and 3) by PART-OF relations (*biodegradation* is a part of Metabolism). Each top level class divides into sub-classes. Figure 2 shows *CA* taxonomy with three keyword examples per class. The taxonomy has 48 classes in total; half of them under *CA*. Table 6 shows the total number of abstracts and keywords per class: 82.4% of the abstracts include keywords for *CA*, and 50.3% and 28.1% for *MOA* and *TOX*, respectively.

We calculated inter-annotator agreement for assigning abstracts to taxonomy classes. For each of the 8 chemicals, 10 abstracts were randomly chosen from the 15 journals. The average agreement between two annotators is the highest with *CA* and *MOA* (78%) and the lowest with *TOX* (62%). The overall agreement is 76%. This result is good, particularly considering the high number of classes and the chance agreement of 1.5%. The disagreements are mostly due to one of the experts annotating as many keywords as possible, and the other one annotating only the ones that classify each abstract as precisely as possible. This was not a serious problem for us, but it demonstrates the importance of detailed guidelines. Also, some of the classes were too imprecise to yield unique distinctions. Future work should focus on refining them further.

4 Automatic classification

To examine whether the classification created by experts provides a good representation of the corpus data and is machine learnable, we conducted a series of abstract classification experiments.

4.1 Methods

4.1.1 Feature extraction

The first step of text categorization (TC) is to transform documents into a feature vector representation. We experimented with two document representation techniques. The first one is the simple 'bag of words' approach (*BOW*) which considers each word in the document as a separate feature. *BOW* was evaluated using three methods which have proved useful in previous TC work: (i) stemming (using the Porter (1980) stemmer) which removes affixes from words, (ii) the TFIDF weighting (Kibriya et al. 2004), and (iii) stop word removal.

The second technique is the recent 'bag of substrings' (*BOS*) method by (Wang et al. 2008) which considers the whole abstract as a string and extracts from it all the length p substrings without affix removal. BOS has proved promising in biomedical TC (Han et al. 2006, Wang et al. 2008) and unlike a traditional grammatical stemmer, does not re-

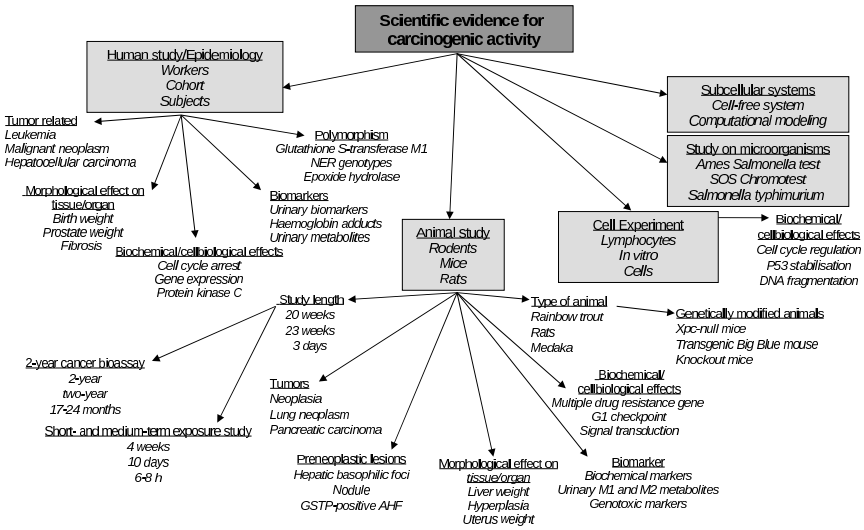


Figure 2: Taxonomy of Carcinogenic Activity

quire domain tuning for optimal performance. Because *BOS* generates substrings with fixed length p, a word shorter than $p-2$ can get obscured by its context[3]. For example, 'mice' would be transformed to '_mice_a', '_mice_b', ..., which is less informative than the original word form. Therefore, we enriched *BOS* features with word forms shorter than $p-2$.

4.1.2 Feature selection

We employed two feature selection methods for dimensionality reduction. The first is Information Gain (*IG*) which has proved useful in TC (Yang and Pedersen 1997). Given a feature's distribution X and class label distribution Y, $IG(X) = H(Y) - H(Y|X)$, $H(X)$ is the entropy of X. The second method $fscore$ optimises the number of features (N). Features are first ranked using the simple $fscore$ criterion (Chen and Lin 2006), and N is selected based on the performance of the SVM classifier using the N features.

4.1.3 Classification

Three classifiers were used: Naive Multinomial Bayesian (*NMB*), Complement Naive Bayesian (*CNB*) (Rennie and Karger 2003) and Linear Support Vector Machines (*L-SVM*) (Vapnik 1995).

NMB is a widely used classifier in TC (Kibriya et al. 2004). It selects the class C with the maximum probability given the document d: $argmax_c\ Pr(C)\prod_{w\in d} Pr(X=w|C)$. $Pr(C)$ can be estimated from the frequency of documents in C. $Pr(X=w|C)$ is estimated as the fraction of tokens in documents of class C that contain w.

CNB extends *NMB* by addressing the problems it has e.g. with imbalanced data and weight magnitude error. The class c of a document is: $argmax_c[logp(\theta_c) - \sum_i f_i log\frac{N_{\tilde{c}i}+\alpha_i}{N_{\tilde{c}}+\alpha}]$. $N_{\tilde{c}i}$ is the number of times term i occurs in classes other than c. α and α_i are the smoothing parameters. $p(\theta_c)$ is the prior distribution of class c.

L-SVM is the basic type of SVM which produces a hyperplane that separates two-class samples with a maximum margin. It handles high dimensional data efficiently, and has shown to perform well in TC (Yang and Liu 1999). Given the data set $X = (\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)$ $y_i \in \{-1, +1\}$, *L-SVM* requires a solution $\mathbf{w}$ to the following unconstrained optimisation problem: $min(\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\max(1 - y_i\mathbf{w^T}\mathbf{x_i}, 0)^2$. Cost parameter C was estimated within range 2^2,..., 2^5 on training data using cross validation. The C of the positive class was weighted by class population ratio $r = \frac{\text{negative population}}{\text{positive population}}$.

4.1.4 Evaluation

We used the standard measures of recall (R), precision (P) and F measure (F) for evaluation. These are defined as follows:

$$R = \frac{TP}{TP+FN} \quad P = \frac{TP}{TP+FP} \quad F = \frac{2\times R\times P}{R+P}$$

Our random baseline is $\frac{P_+}{N+P_+}$.

[3]Minus 2 because of space characters.

P_+/N: positive/negative population TP: truth positive; FN: false negative, FP: false positive

4.2 Experimental evaluation

4.2.1 Data

Our data was the expert annotated CRA corpus.

4.2.2 Document preprocessing

We first evaluated the *BOW* preprocessing technique with and without the use of (i) the Porter (1980) stemmer, (ii) TFIDF, (iii) stop word removal, and (iv) their combinations. The evaluation was done in the context of the binary relevance classification of abstracts (not in the context of the main taxonomic classification task to avoid overfitting preprocessing techniques to the taxonomy). Only (iii) improved all the classifiers and was thus adopted for the main experiments. The poor performance of (i) demonstrates that a standard stemmer is not optimal for our data. As highlighted by (Han et al. 2006, Wang et al. 2008), semantically related biological terms sharing the same stem are not always reducible to the stem form.

4.2.3 Feature selection

We evaluated the feature selection methods on two taxonomy classes: the most balanced class 'Animal study' (positive/negative 1:1.4) and an imbalanced class 'Adducts' (positive/negative 1:6.5). *IG* was used for the fixed *N* setting and *fscore* for the *dynamic N* setting. Each combination of classifiers (*NMB/CNB/SVM*), document representations (*BOW*, *BOS*) and settings for *N* (dynamic, ..., 83098) was evaluated. The results show that the *dynamic* setting yields consistent improvement on all the setups (although the impact on *SVM*'s is not big). Also the optimal *N* varies by the data and the classifier. Thus, we used the *dynamic* feature selection in the taxonomic classification.

4.2.4 Taxonomic classification

Experimental setup We ran two sets of experiments on the corpus, using 1) *BOW* and 2) *BOS* for feature extraction. Without feature selection, *BOW* had c. 9000 features and *BOS* c. 83000. Features were selected using *fscore*. For each class with more than 20 abstracts (37 in total)[4], three "one against other" classifiers (*NMB*, *CNB* and *L-SVM*) were trained and tested using 10-fold cross validation.

[4]The classes with less than 20 abstracts may have less than 2 positive abstracts in each fold of 10 fold CV, which is not representative for the class population.

Method	Feature Set	P	R	F
NMB	BOW	0.59	0.75	0.66
NMB	BOS	0.62	0.82	0.70
CNB	BOW	0.52	0.74	0.60
CNB	BOS	0.57	0.76	0.64
SVM	BOW	0.68	0.76	0.71
SVM	BOS	0.71	0.77	0.74

Table 2: Performance of classifiers with *BOS*/*BOW*

Class	Method	P	R	F
CA	NMB	0.94	0.89	0.91
CA	CNB	0.92	0.94	0.93
CA	SVM	0.93	0.93	0.93
MOA	NMB	0.88	0.81	0.84
MOA	CNB	0.84	0.82	0.83
MOA	SVM	0.92	0.80	0.86
TOX	NMB	0.66	0.83	0.74
TOX	CNB	0.70	0.80	0.75
TOX	SVM	0.76	0.79	0.78

Table 3: Result for the top level classes

Results Table 2 shows the average performance for the whole taxonomy. The performance of *BOS* is better than that of *BOW* according to all the three measures. On average, *BOS* outperforms *BOW* by 4% in P and F, and 3% in R. *SVM* yields the best overall P and F (*0.71* and *0.74*) with *BOS*. Surprisingly, *NMB* outperforms *CNB* with all the settings. *NMB* yields the best overall R with *BOS* (0.82) but its P is notably lower than that of *SVM*.

Table 3 shows the average P, R and F for the top level classes using the best performing feature set *BOS* with the three classifiers. *CA* has the best F (0.93). Its positive population is the highest (positive/negative: 5:1). *TOX* with a lower positive population (1:2.6) has still good F (0.78). R and P are balanced with an average difference of 0.06.

Table 4 shows the distribution of F across the taxonomy. There is a clear correlation between

No. of abstracts(f)	Classes	F	Random
$f > 300$	9	0.80	0.38
$100 < f \leq 300$	12	0.73	0.13
$20 < f \leq 100$	16	0.68	0.04

Table 4: Mean F and random baseline for taxonomic classes in three frequency ranges.

frequency and performance: the average F decreases with descending frequency range, revealing increased classification difficulty. Classes with more than 300 abstracts have the highest average F (0.80 with standard deviation (*SD*) 0.08). Classes with 20-100 abstracts have the average F 0.68 (*SD* 0.11), which is lower but still fairly good. No class has F lower than 0.46, which is much higher than the average random baseline of 0.11.

5 User Test

A user test was carried out to examine the practical usefulness of the automatic classification in a near real-world scenario. The *L-SVM+BOS* classifier was applied to the PubMed abstract data (from 1998-2008) of five unseen chemicals representing genotoxic (geno) and non-genotoxic (non) MOAs (see table 5). The results were displayed to two experts in a friendly web interface. The experts were invited to imagine that they have submitted a query to a system, the system has returned the classification of relevant abstracts for each chemical, and the task is to judge whether it is correct. The top 500 *BOS* features per class were shown to aid the judgement.

Results were evaluated using precision (P) (recall could not be calculated as not all of the positive polulation was known). Table 5 shows the average P for chemicals and top level classes. The results are impressive: the only chemical with P lower than 0.90 is polychlorinated biphenyls (PCB). As PCB has a well-known neuro-behavioural effect, the data includes many abstracts irrelevant for CRA. Most other errors are due to the lack of training data for low frequency classes. For example, the CRA corpus had only 27 abstracts in "DNA repair (damage)" class, while the new corpus has many abstracts on DNA damage some of which are irrelevant for CRA.

The experts found the tool easy to use and felt that if such a tool was available to support real-world CRA, it could significantly increase their productivity and also lead to more consistent and thorough CRA. Such a wide range of scientific evidence is difficult to gather via manual means, and chemical carcinogenesis is such a complex process that even the most experienced risk assessor is incapable of memorizing the full range of relevant evidence without the support of a thorough specification / taxonomy.

Name	MOA	Σ	P	Class	P
Aflatoxin B1	geno	189	0.95	CA	0.94
Benzene	geno	461	0.99	MOA	0.95
PCB	non	761	0.89	TOX	0.99
Tamoxifen	non	382	0.96		
TCDD	non	641	0.96		

Table 5: Chemicals and the results of the user test

6 Conclusion and Future Work

The results of our inter-annotator agreement tests, automatic classification experiments and the user test demonstrate that the taxonomy created by risk assessors is accurate, well-defined, and can be useful in a real-world CRA scenario. This is particularly encouraging considering that the taxonomy is based on biomedical annotation. As highlighted by (Kim et al. 2008), expert annotation is more challenging and prone to inter-annotator disagreement than better-constrained linguistic annotation. We believe that we obtained promising results because we worked in collaboration with risk assessors and developed technology which imitates their current practices as closely as possible.

Most related work focuses on binary classification, e.g. BioCreative II had a subtask (Krallinger et al. 2008) on the relevance classification of abstracts for protein interactions. The few works that have attempted multi-classification include e.g. that of Aphinyanaphongs et al. (2005) who applied *NMB*, SVM and AdaBoost to classify abstracts of internal medicine into four categories, and that of Han et al. (2006) who used *BOS* and *NMB*/*L-SVM* to classify abstracts in five categories of protein post-translational modifications.

In the future, we plan to refine the taxonomy further by careful analysis of keyword types found in the data and the taxonomic relationships defined by experts. This will help to transform the taxonomy into a better-developed knowledge resource. We also need to extend the taxonomy. Although our results show that the current taxonomy provides a good basis for the classification of CRA literature, it is not comprehensive: more data is required especially for low frequency classes, and the taxonomy needs to be extended to cover more specific MOA types (e.g. further subtypes of non-genotoxic chemicals).

The taxonomy can be extended by manual annota-

Change in F	Σ Classes	Abstracts of class		
		20-100	100 - 200	200 - 1100
$\Delta_F > 1\%$	16 (43%)	75%	33%	8%
$\|\Delta_F\| \leq 1\%$	15 (41%)	6%	44%	75%
$\Delta_F < -1\%$	6 (16%)	19%	33%	17%

Table 6: F gain(Δ_F) of *MeSH* compared to *BOS*

Class	Σ	F
Carcinogenic activity	1068	92.8
Human study/epidemiology	190	77.7
Animal study	629	80.2
Cell experiments	319	78.5
Study on microorganisms	44	85.2
Mode of Action	653	85.5
Genotoxic	421	89.1
Nongenotoxic	324	76.3
Toxicokinetics	356	77.7
Absorption, . . . ,excretion	113	69.8
Metabolism	268	76.4
Toxicokinetic modeling	31	84.6

Table 7: Σ abstracts and F of level 1,2 classes.

tion, supplementing it with additional information in knowledge resources and/or by automatic methods. One knowledge resource potentially useful is the Medical Subject Headings (MeSH) taxonomy (Nelson et al. 2002) which classifies PubMed abstracts according to manually defined terms. We performed a small experiment to investigate the usefulness of MeSH for supplementing our current classification. MeSH terms were first retrieved for each abstract using EFetch (NCBI 2005) and then appended to the *BOS* feature vector. Best features were then selected using *fscore* and classified using *L-SVM*. The figures in table 6 show that the results improved significantly for 43% of the low frequency classes. Although this demonstrates the potential usefulness of additional resources, given the rapidly evolving nature of CRA data, the best approach long term is to develop technology for automatic updating of the taxonomy from literature. Given the basic resources we have constructed, the development of such technology is now realistic and can be done using unsupervised or semi-supervised machine learning techniques, e.g. (Cohen and Hersh 2005, Blaschko and Gretton 2009).

The automatic classification could be improved by the use of more sophisticated features extracted using NLP tools that have been tuned for biomedical texts, such as parsers, e.g. (Tsuruoka et al. 2005), and named entity recognizers, e.g. (Corbett et al. 2007), and exploiting resources such as the BioLexion (Sasaki et al. 2008).

Our long term goal is to develop a TM tool specifically designed for CRA. Some tools have recently been built to assist other critical activities of biomedicine (e.g. literature curation for genetics). A few of them have been evaluated for their practical usefulness in a real-world scenario (Karamanis et al. 2008, Demaine et al. 2006). Such tools and evaluations act as an important proof of concept for biomedical TM and help to develop technology for the needs of practical applications.

According to the interviews we conducted (Section 2), a tool capable of identifying, ranking and classifying articles based on the evidence they contain, displaying the results to experts, and assisting also in subsequent steps of CRA would be particularly welcome. Such a tool, if developed in close collaboration with users, could significantly increase the productivity of CRA and enable risk assessors to concentrate on what they are best at: the expert judgement.

Acknowledgements Our work was funded by the Royal Society (UK), the Medical Research Council (G0601766) (UK) and the Swedish Council for Working Life and Social Research (Sweden). LS was supported by a Dorothy Hodgkin Postgraduate Award (UK). We would like to thank Ian Lewin for his assistance at the early stages of this work and for providing the first version of the annotation tool. We are also grateful to Johan Hogberg for supporting the annotation and the taxonomy construction work.

References

Sophia Ananiadou, Douglas B. Kell, and Jun ichi Tsujii. Text mining and its potential applications in systems biology. *Trends in Biotechnology*, 24(12), 2006.

Y. Aphinyanaphongs, I. Tsamardinos, A. Statnikov, D. Hardin, and C.F. Aliferis. Text categorization models for high-quality article retrieval in internal medicine. *JAMIA*, 12(2), 2005.

Matthew Blaschko and Arthur Gretton. Learning taxonomies by dependence maximization. In *22rd NIPS*, 2009.

Yi-Wei Chen and Chih-Jen Lin. Combining SVMs with various feature selection strategies. In *Feature extraction, foundations and applications*. 2006.

Aaron M. Cohen and William R. Hersh. A survey of

current work in biomedical text mining. *Briefings in Bioinformatics*, 6(1), 2005.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.*, 20(1), 1960.

K. Bretonnel Cohen, Hong Yu, Philip E. Bourne, and Lynette Hirschman. Translating biology:text mining tools that work. In *PSB*, 2008.

Peter Corbett, Colin Batchelor, and Simone Teufel. Annotation of chemical named entities. In *Proceedings of the ACL*, 2007.

Jeffrey Demaine, Joel Martin, Lynn Wei, and Berry de Bruijn. Litminer: integration of library services within a bio-informatics application. *Biomedical Digital Libraries*, 3(1), 2006.

ECHA, 2008. *Guidance on Information Requirements and Chemical Safety Assessment.* European Chemicals Agency, 2008.

Bo Han, Zoran Obradovic, Zhang zhi Hu, Cathy H. Wu, and Slobodan Vucetic. Substring selection for biomedical document classification. *Bioinformatics*, 22, 2006.

Lawrence Hunter and K. Bretonnel Cohen. Biomedical language processing: What's beyond pubmed? *Mol Cell*, 21(5), 2006.

N. Karamanis, R. Seal, I. Lewin, P. McQuilton, A. Vlachos, C. Gasperin, R. Drysdale, and T. Briscoe. Natural language processing in aid of flybase curators. *BMC Bioinformatics*, 9(1), 2008.

Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. Multinomial naive bayes for text categorization revisited. In *Australian Conference on AI*, volume 3339, 2004.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. Corpus annotation for mining biomedical events from lterature. *BMC Bioinformatics*, 9, 2008.

Martin Krallinger, Florian Leitner, Carlos Rodriguez-Penagos, and Alfonso Valencia. Overview of the protein-protein interaction annotation extraction task of biocreative ii. *Genome Biology*, 2008.

J.Richard Landis and Gary G.Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 1977.

Ian Lewin, Ilona Silins, Anna Korhonen, Johan Hogberg, and Ulla Stenius. A new challenge for text mining: Cancer risk assessment. In *Proceedings of the ISMB BioLINK Special Interest Group on Text Data Mining.*, 2008.

NCBI. Efetch entrez utility, 2005. URL `http://www.ncbi.nlm.nih.gov/entrez/query/static/efetch_help.html`.

Sturart J. Nelson, Tammy Powell, and Besty L. Humphreys. The Unified Medical Language System (UMLS) Project. In *Encyclopedia of Library and Information Science*, pages 369–378. Marcel Dekker, 2002.

M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

Jason D. M. Rennie and David Karger. Tackling the poor assumptions of naive bayes text classifiers. In *In Proceedings of the 20th ICML*, 2003.

Y. Sasaki, S. Montemagni, P. Pezik, D. Rebholz-Schuhmann, J. McNaught, and S. Ananiadou. BioLexicon: A Lexical Resource for the Biology Domain. 2008.

Y. Tsuruoka, Y. Tateishi, J. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii. Developing a Robust Part-of-Speech Tagger for Biomedical Text. 3746, 2005.

EPA, 2005. *Guidelines for carcinogen risk assessment.* U.S. Environmental Protection Agency, 2005. URL `http://www.epa.gov/iris/cancer032505.pdf`.

Vladimir N. Vapnik. *The nature of statistical learning theory*. New York, NY, USA, 1995.

Hongning Wang, Minlie Huang, Shilin Ding, and Xiaoyan Zhu. Exploiting and integrating rich features for biological literature classification. *BMC Bioinformatics*, 9(Suppl 3), 2008.

Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd SIGIR*, New York, NY, USA, 1999.

Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. 1997.

Pierre Zweigenbaum, Dina Demner-Fushman, Hong Yu, and Kevin B. Cohen. Frontiers of biomedical text mining: current progress. *Brief Bioinform*, 8(5), 2007.

Transforming Controlled Natural Language Biomedical Queries into Answer Set Programs

Esra Erdem and **Reyyan Yeniterzi**
Faculty of Engineering and Natural Sciences
Sabancı University
Orhanlı, Tuzla 34956 Istanbul, Turkey

Abstract

We introduce a controlled natural language for biomedical queries, called BIOQUERYCNL, and present an algorithm to convert a biomedical query in this language into a program in answer set programming (ASP)—a formal framework to automate reasoning about knowledge. BIOQUERYCNL allows users to express complex queries (possibly containing nested relative clauses and cardinality constraints) over biomedical ontologies; and such a transformation of BIOQUERYCNL queries into ASP programs is useful for automating reasoning about biomedical ontologies by means of ASP solvers. We precisely describe the grammar of BIOQUERYCNL, implement our transformation algorithm, and illustrate its applicability to biomedical queries by some examples.

1 Introduction

The rapid increase in the popularity and usage of Web leads researchers to store data and make it publicly available in many ways. In particular, to facilitate access to its desired parts, it is stored in a structured form, like ontologies. These ontologies can be queried with an SQL-like formal query language. However, since these ontologies have been developed for and widely used by people that lacks the necessary knowledge in a formal query language, a simpler and more commonly known language is needed to represent queries. A natural language is the perfect answer, but ambiguities in its grammar and vocabulary make it difficult to automate reasoning about queries in natural language. Therefore, to represent queries, we consider a middle ground between these two options: a Controlled Natural Language (CNL).

A CNL is a subset of a natural language, with a restricted grammar and vocabulary, that overcomes the ambiguity of natural languages. Since we consider queries in a specific domain, namely biomedicine, and over specific sources of information, namely biomedical ontologies, a CNL designed and developed for reasoning about biomedical ontologies is sufficient to represent biomedical queries. Essentially, a CNL is a formal language but with a look of a natural language. Therefore, compared to a natural language, a CNL can be easily converted to some other formalisms. This allows us to use automated reasoners, specifically developed for such formalisms, to find answers to queries expressed in a CNL.

One such formalism is Answer Set Programming (ASP) (Baral, 2003). ASP is a new knowledge representation and reasoning paradigm which supports representation of defaults, constraints, preferences, aggregates, etc., and provides technologies that allow us to automate reasoning with incomplete information, and to integrate other technologies, like description logics reasoners and Semantic Web technologies. For instance, in (Bodenreider et al., 2008), the authors illustrate the applicability and effectiveness of using ASP to represent a rule layer that integrates relevant parts of some biomedical ontologies in RDF(S)/OWL, and to compute answers to some complex biomedical queries over these ontologies.

Although CNLs are appropriate for expressing biomedical queries, and methods and technologies

Proceedings of the Workshop on BioNLP, pages 117–124,
Boulder, Colorado, June 2009.

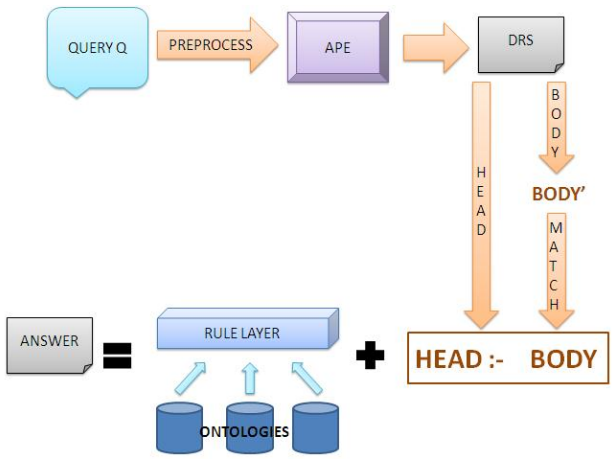


Figure 1: Architecture of the Overall System

of ASP are appropriate for automated reasoning about biomedical ontologies, there is no algorithm to convert a CNL biomedical query into a program. In (Bodenreider et al., 2008), biomedical queries are represented as programs in ASP; however, these programs are constructed manually. However, manually constructing ASP programs to represent biomedical queries is not only time consuming but also requires expertise in ASP. This prevents automating the whole process of computing an answer to a query, once it is given in a CNL.

In this paper, we design and develop a CNL (called BIOQUERYCNL) for expressing biomedical queries over some ontologies, and introduce an algorithm to convert a biomedical query expressed in this CNL into a program in ASP. The idea is to automatically compute an answer to the query using methods of (Bodenreider et al., 2008), once the user types the query. This idea is illustrated in Figure 1.

Similar approaches of using a CNL for querying ontologies have been investigated in various studies. For instance, (Bernstein et al., 2005) considers queries in the controlled natural language, Attempto Controlled English (ACE) (Attempto, 2008), and transforms them into queries in PQL (Klein and Bernstein, 2004) to be evaluated by a query engine. (Bernstein et al., 2006) presents a system that guides the user to write a query in ACE, and translates the query into SPARQL to be evaluated by the reasoner of JENA (Jena, 2008). On the other hand, (Kaufmann et al., 2006) transforms a given natural language query to a SPARQL query (using the Stanford Parser and WORDNET) to be evaluated by a reasoner like that of JENA. Our work is different from these studies in two ways: we consider queries over biomedical ontologies (thus different forms of queries, and vocabulary), and we transform a query into an ASP program to automate reasoning over a rule layer presented in ASP.

Transformations of natural language sentences into ASP has been studied in (Baral et al., 2008) and (Baral et al., 2007). In (Baral et al., 2008), the authors introduce methods to transform some simple forms of sentences into ASP using Lambda Calculus. In (Baral et al., 2007), the authors use C&C tools (CC, 2009) to parse the some forms of natural language input, and perform a semantic analysis over the parser output using BOXER (Boxer, 2009), to do reasoning in ASP. Our work is different in that we consider a CNL to express queries, and introduce a different method for converting CNL to a program in ASP, via Discourse Representation Structures (DRS) (Kamp, 1981).

In the rest of the paper, first we briefly discuss ASP with some examples (Section 2). Then we define the grammatical structure of BIOQUERYCNL and give some examples (Section 3). Next, we introduce our algorithm for transforming a BIOQUERYCNL query into an ASP program and explain it by an example (Section 4). We conclude with a discussion of challenges related to the implementation of our algorithm (Section 5) and other related problems that we are working on (Section 6).

2 Answer Set Programming

Answer Set Programming (ASP) (Lifschitz, 1999; Marek and Truszczyński, 1999; Niemelä, 1999; Baral, 2003) is a new knowledge representation and reasoning paradigm which supports representation of defaults, constraints, preferences, aggregates, etc., and provides technologies that allow us to automate reasoning with incomplete information, and to integrate other technologies, like description logics reasoners and Semantic Web technologies.

In ASP, knowledge is represented as a "program" (a finite set of "rules") whose meaning is captured by its models (called "answer sets" (Gelfond and Lifschitz, 1988)). Answer sets for a program can be computed by "answer set solvers" such as DLV

(DLV, 2009). Consider for instance the program:

```
gene_gene(``ADRB1'',``CHRM5'').
gene_gene(``CHRM1'',``CHRM5'').

chain(X,Y) :- gene_gene(X,Y).
chain(X,Y) :- gene_gene(Y,X).
chain(X,Y) :- gene_gene(X,Z), chain(Z,Y).
```

The first rule expresses that the gene ADRB1 interacts with the gene CHRM5. The second rule expresses that the gene CHRM1 interacts with the gene CHRM5. The third, the fourth, and the fifth rules express a chain of such interactions. In a rule containing `:-`, the left-hand-side of `:-` is called the *head* of the rule, the right-hand-side is called the *body* of the rule. Such a rule `p :- q, r.` is read as "`p` if `q` and `r`". Here the head atom is `p`, and the body atoms are `q` and `r`. The answer set for this program describes that there is a chain of interactions between CHRM1 and CHRM5, ADRB1 and CHRM5, and ADRB1 and CHRM1.

As mentioned above, the language of ASP is expressive enough to represent defaults, constraints, preferences, aggregates, etc.. For instance, the rule

```
treats_2diseases(R) :-
 #count{D:treats(R,D)}>=2, drug(R).
```

describes drugs `R` that treat at least 2 diseases.

3 A Controlled Natural Language for Biomedical Queries

We introduce a controlled natural language, called BIOQUERYCNL, to express biomedical queries, whose grammar is shown in Table 1. This grammar should be considered in connection with the given biomedical ontologies. The italic words in the grammar, for instance, represent the information extracted from the related ontologies. We call these italic words ontology functions; the detailed description of these functions are given in Table 2.

With BIOQUERYCNL, the users can ask simple queries, queries with nested relative clauses (with any number of conjunctions and disjunctions), and queries with cardinalities. Some sample queries are given below.

(Q1) Which symptoms are alleviated by the drug Epinephrine?

(Q2) What are the side-effects of the drugs that treat the disease Asthma?

(Q3) What are the genes that are related to the disease Asthma and are targeted by the drug Epinephrine?

(Q4) What are the symptoms of the diseases that are related to the gene ADRB1 or that are treated by the drug Epinephrine?

(Q5) Which genes are targeted by at least 2 drugs and are related to at most 3 diseases?

BIOQUERYCNL is a subset of Attempto Controlled English (ACE) (Attempto, 2008), which can represent a wide range of queries (Fuchs et al., 2008), specialized for biomedical ontologies.

4 Converting Controlled Natural Language Queries to Programs

We have implemented an algorithm, QUERY, presented in Algorithm 1, that obtains an ASP rule $Head \leftarrow Body$ from a query Q expressed in BIOQUERYCNL, via transforming Q into a DRS. We will explain the main steps of the QUERY algorithm by an example, considering query (Q4).

Algorithm 1 QUERY(Q)

Input: A query Q
Output: An ASP rule $Head \leftarrow Body$

1: $D :=$ Find the DRS of Q
2: $Head :=$ HEAD(D)
3: $Body' :=$ BODY(D)
4: $Body :=$ POSTPROCESSING($Body'$)
5: $return\ Head \leftarrow Body$

4.1 Transforming a CNL Query into DRS

Attempto Controlled English (ACE) text can be converted into Discourse Representation Structures (DRS) (Kamp, 1981) — a variant of the first-order logic that is used for the dynamic interpretation of natural language and systematic translation of natural language into logical form — without any ambiguity, using tools like Attempto Parsing Engine (APE). APE converts ACE text to DRS by an approach similar to (Blackburn and Bos, 2005), as explained in (Fuchs et al., 2008). For instance, APE transforms query (Q4) into the following DRS:

Table 1: The Grammar of BIOQUERYCNL

QUERY →	YESNOQUERY | WHQUERY QUESTIONMARK
YESNOQUERY →	DODOESQUERY | ISAREQUERY
WHQUERY →	WHATQUERY | WHICHQUERY
DODOESQUERY →	[Do | Does] $Type()$ $Instance(T)$ PREDICATERELATION
ISAREQUERY →	[Is | Are] $Type()$ $Instance(T)$ $Verb(T)$
WHATQUERY →	What BE $Type()$ that PREDICATERELATION
WHATQUERY →	What BE OFRELATION that PREDICATERELATION
WHATQUERY →	What BE OFRELATIONINSTANCE that PREDICATERELATION
WHICHQUERY →	Which $Type()$ PREDICATERELATION
OFRELATION →	$Noun(T)$ of $Type()$
OFRELATIONINSTANCE →	$Noun(T)$ of $Type()$ $Instance(T)$
PREDICATERELATION →	ACTIVERELATION (CONNECTOR (that)? PREDICATERELATION)*
PREDICATERELATION →	PASSIVERELATION (CONNECTOR (that)? PREDICATERELATION)*
ACTIVERELATION →	$Verb(T, T')$ $Type()$ $Instance(T')$
ACTIVERELATION →	$Verb(T, T')$ GENERALISEDQUANTOR PositiveNumber $Type()$
PASSIVERELATION →	BE $Verb(T', T)$ by $Type()$ $Instance(T')$
PASSIVERELATION →	BE $Verb(T', T)$ by GENERALISEDQUANTOR PositiveNumber $Type()$
BE →	is | are
CONNECTOR →	and | or
GENERALISEDQUANTOR →	at least | at most | more than | less than | exactly
QUESTIONMARK →	?

Table 2: The Ontology Functions

$Type()$	returns the type information the ontologies keep, ex. gene, disease, drug
$Instance(T)$	returns instances of the type T, ex. Asthma for type disease
$Verb(T)$	returns the verbs related to the type T, ex. approve for type drug
$Verb(T, T')$	returns the verbs where type T is the subject and type T' is the object, ex. drug treat disease
$Noun(T)$	returns the nouns that are related to the type T, ex. symptom for type disease

```
[A,B,C,D]
query(A,what)-1
predicate(B,be,A,C)-1
relation(C,of,D)-1
object(C,symptoms,countable,na,eq,1)-1
  [E,F,G]
  modifier_pp(F,to,E)-1
  property(G,related,pos)-1
  predicate(F,be,D,G)-1
  object(E,gene_ADRB1,countable,na,eq,1)-1
  v
  [H,I]
  predicate(I,treated,H,D)-1
  object(H,drug_Epinephrine,
         countable,na,eq,1)-1
object(D,diseases,countable,na,geq,2)-1
```

Note that the DRS consists of two kinds of expressions. The lines with a list of uppercase letters, like `[E,F,G]`, describe the domain of the DRS; each uppercase letter is a referent. The rest of the DRS describe the conditions about the domain.

The DRS above contains some predefined predicates, such as `object`, `property`, `predicate`, `query`, etc.. All the nouns, adjectives, verbs, modifiers, etc. are represented with one of them. For instance,

- `object` describes objects and the relevant forms of nouns denoting them (like "diseases")
- `predicate` describes relations that are pro-

duced by different forms of verbs (like "treated"),

- `relation` describes relations that are produced by of-constructions (like "symptoms of disease"),

- `query` describes the form of the query and the objects that the query is referring to.

Ontologies represent relations between concepts. A rule layer over ontologies introduce further concepts integrating them. ASP takes into account relevant concepts and relations to answer a given query about these ontologies. In the biomedical queries we consider, the concepts and instances are represented with `object` and the relations between these concepts are represented with `predicate` and `relation`. The `query` is also important in terms of the type of information the user asks for.

4.2 Constructing the Head and the Body Atoms

Once the corresponding DRS is obtained from a given BIOQUERYCNL query, the head and the body atoms are constructed by analyzing the conditions in the DRS, as described in Algorithms 2 and 3.

The HEAD algorithm is about the `query` predicate, which refers to objects or relations that are asked for in the given query. By following the referents, starting from the one mentioned in `query`, the algorithm finds out the type of the information that is asked for in the given query. Consider, for instance, query (Q4). The referent mentioned in `query(A,what)` is `A`. It is mentioned in `predicate(B,be,A,C)-1`, and here it denotes an object with referent `C`. Now let's find where `C` is mentioned: in `object(C,symptoms,countable,na,eq,1)-1` to denote symptoms. Therefore, the query asks for symptoms. Based on this information, Algorithm 2 returns the head of the ASP rules as follows:

```
what_be_symptoms(SYM1)
```

The BODY algorithm analyzes the `predicate` and the `relation` predicates. These two predicates describe relations between objects described by the `object` predicates. The algorithm starts from the `predicate` and the `relation` predicates, and then, by following the referents, it returns the body atoms of the ASP rule. For instance, Algorithm 3 returns the following body atoms for query (Q4):

```
symptoms_of_diseases(symptom_SYM1,
                     disease_DIS1)
diseases_be_related_to_gene(disease_DIS1,
                            gene_``ADRB1'')
drug_treated_diseases(drug_``Epinephrine'',
                      disease_DIS1)
```

These body atoms are given to POSTPROCESSING step, to produce bodies of the ASP rules.

4.3 Constructing the ASP Rules

POSTPROCESSING is the last step of the QUERY algorithm. At this step, first the number of rules is determined, and then the body atoms are placed in the bodies of these rules. In ASP, a conjunctive query can be represented by a rule. However, disjunctive queries are represented by several rules with same head but different bodies. For instance, query (Q4) is a disjunctive query (a disjunction of two queries), so there will be two rules representing this query:

```
what_be_symptoms(SYM1) :-
 symptoms_of_diseases(symptom_SYM1,
                      disease_DIS1),
 diseases_be_related_to_gene(disease_DIS1,
                             gene_``ADRB1'').

what_be_symptoms(SYM1) :-
 drug_treated_diseases(drug_``Epinephrine'',
                       disease_DIS1),
 symptoms_of_diseases(symptom_SYM1,
                      disease_DIS1).
```

Next, the predicate names in the bodies of these rules are matched with the names of the already defined predicates in ontologies or in the rule layer over these ontologies. After matching the predicate names, the parameters of the predicates may have to be reordered.

The matching of the predicates very much depends on the user interface (UI). If UI enforces users to use a specific grammar and lexicon while forming the query, then the matching can be done with an easy table look-up method. If the UI allows more flexibility of constructing a query, then the matching algorithm should use some basic Natural Language Processing methods and similarity metrics to find the most probable matching.

After matching the predicates, the ordering of the parameters can be done easily. The BODY algorithm

Algorithm 2 HEAD(D)

Input: A DRS

Output: Head of an ASP rule

1: *query*(*Ref*, *QuestionWord*) // e.g., *query*(A, *which*) for "Which drug ..."
2: **if** *Ref* is an object **then**
3: *Object* := REFERSTO(*Ref*) // e.g., A refers to a "drug" $DRG1$
4: *Head* := CONCAT(*QuestionWord*, *Object*, *Ref*) // e.g., *which_drug*($DRG1$)
5: **else if** *Ref* is a subject of a predicate **then** // *query*(A, *what*) for "What are the genes ..."
6: *Object* := REFERSTO(*Ref*) // e.g., A refers to "genes" $GENE1$
7: *Head* := CONCAT(*QuestionWord*, *Predicate*, *Object*, *Ref*) // e.g., *what_be_genes*($GENE1$)
8: **end if**
9: **return** *Head*

returns the body predicates with the parameters. In these parameters, the type and the instance names are kept together. Thus, ordering of those parameters are done just by using the type information. After the ordering is done, the type information part is removed from the parameters.

For instance, after matching the predicates, we get the following ASP rule for query (Q4).

```
what_be_symptoms(SYM1) :-
 disease_symptom(DIS1,SYM1),
 disease_gene(DIS1,``ADRB1'').

what_be_symptoms(SYM1) :-
 treats_disease(``Epinephrine'',DIS1),
 disease_symptom(DIS1,SYM1).
```

With an ASP rule layer over ontologies, and this ASP program, an ASP solver, like DLVHEX (DLVHEX, 2009), returns an answer to query (Q4).

For instance, consider the ASP rule layer, and the gene, disease, drug ontologies of (Bodenreider et al., 2008). The ontologies of (Bodenreider et al., 2008) are obtained from the ontologies PHARMGKB (PharmGKB, 2008), UNIPROT (UniProt, 2008), GENE ONTOLOGY (GO) (GeneOntology, 2008), GENENETWORK database (GeneNetwork, 2008), DRUGBANK (DrugBank, 2008), and the Medical Symptoms and Signs of Disease web page (MedicalSymptomsSignsDisease, 2008). With this rule layer and the ontologies, and the ASP program above, the following is a part of the answer DLVHEX finds to the query above:

```
noisy breathing       faster breathing
shortness of breath   coughing
chest tightness       wheezing
```

4.4 Another Example

The algorithm discussed above returns the following ASP program for query (Q5):

```
which_genes(GN1) :-
 2<=#count{DRG1:drug_gene(DRG1,GN1)},
 #count{DIS1:disease_gene(DIS1,GN1)}<=3.
```

Since query (Q5) contains cardinality constraints, the ASP program uses the aggregate `#count`.

More examples of biomedical queries, and the ASP programs generated by our program can be seen at `http://people.sabanciuniv.edu/esraerdem/bioquery-asp/bionlp09/`.

5 Implementational Issues

We have implemented the algorithms explained above in PERL. We have used Attempto Parsing Engine APE to convert a given BIOQUERYCNL query into a DRS. Since BIOQUERYCNL is about biomedical ontologies, we provided APE some information about biomedical concepts, such as gene, drug, and words that represent relations between these concepts such as treat, target etc..

However, providing such information is not sufficient to convert all BIOQUERYCNL biomedical queries into programs, mainly due to specific instances of these concepts (consider, for instance, various drug names that appear in ontologies). One way to deal with this problem is to extract from the ontologies all instances of each concept and provide them to APE as an additional lexicon. This may not be the perfect solution since this process has to be repeated when an instance is added to the ontology. An alternative way can be enforcing the user to enter

Algorithm 3 BODY(D)

Input: A DRS
Output: Body of an ASP rule

Body := empty string
for each predicate P **do**
    // P can be of the form *predicate*(*Ref*, *Verb*, *SubRef*), like *predicate*(H, *targeted*, A)
  Subject := REFERSTO(*SubRef*) // e.g., A refers to "genes" $GENE1$
  if P has a verb phrase modifier **then**
    ⟨*Modifier*, *Object*⟩ := REFERSTO(*Ref*) // e.g., H refers to ⟨ "by", "drug" $DRG1$ ⟩
  end if
  if P has an object **then** // P can be of the form *predicate*(*Ref*, *Verb*, *SubRef*, *ObjRef*)
    Object := REFERSTO(*ObjRef*)
  end if
  Body := CONCAT(*Body*, *Subject*, *Verb*, *Modifier*, *Object*)
    // e.g., *genes_targeted_by_drugs*($GENE1$, $DRG1$)
end for
for each relation R **do**
    // R can be of the form *relation*(*Ref1*, *of*, *Ref2*), like *relation*(C, *of*, D)
  Object1 := REFERSTO(*Ref1*) // e.g., C refers to "symptoms" $SYM1$
  Object2 := REFERSTO(*Ref2*) // e.g., D refers to "diseases" $DIS1$
  Body := CONCAT(*Body*, *Object1*, *"'of"'*, *Object2*)
    // e.g., *symptoms_of_diseases*($SYM1$, $DIS1$)
end for
return $Body$

the concept name just before the instance (like "the drug Epinephrine") in the query. This is how we deal with instance names, in the current version of our implementations. However, such BIOQUERYCNL queries are not in the language of APE; so, with some preprocessing, we rewrite these queries in the correct syntax for APE.

6 Conclusion

We have designed and developed a Controlled Natural Language (CNL), called BIOQUERYCNL, to represent biomedical queries over some ontologies, and provided a precise description of its grammatical structure.

We have introduced an algorithm to convert queries in BIOQUERYCNL to a program in Answer Set Programming (ASP). The idea is to compute answers to these queries automatically, by means of automated reasoners in ASP, over biomedical ontologies in RDF(S)/OWL and a rule layer in ASP integrating these ontologies. Our algorithm can handle various forms of simple/complex disjunctive/conjunctive queries that may contain (nested) relative clauses and cardinality constraints.

We have implemented this algorithm in PERL, and tried it with the ASP rule layer, and the ontologies of (Bodenreider et al., 2008).

One essential part of the overall system is an intelligent user interface that allows a user to enter biomedical queries in BIOQUERYCNL. Design and implementation of such a user-interface is a part of our ongoing work.

Acknowledgments

Thanks to Tobias Kuhn for his help with ACE. This work is supported by the Scientific and Technological Research Council of Turkey (TUBITAK) grant 108E229.

References

Attempto. 2008. `http://attempto.ifi.uzh.ch/site/`.

Chitta Baral, Juraj Dzifcak, and Luis Tari. 2007. Towards overcoming the knowledge acquisition bottleneck in answer set prolog applications: Embracing natural language inputs. In *Proc. of ICLP*, pages 1–21.

Chitta Baral, Juraj Dzifcak, and Tran Cao Son. 2008. Using answer set programming and lambda calculus to characterize natural language sentences with normatives and exceptions. In *Proc. of AAAI*, pages 818–823.

Chitta Baral. 2003. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press.

Abraham Bernstein, Esther Kaufmann, Anne Göhring, and Christoph Kiefer. 2005. Querying ontologies: A controlled english interface for end-users. In *Proc. of ISWC*, pages 112–126.

Abraham Bernstein, Esther Kaufmann, Christian Kaiser, and Christoph Kiefer. 2006. Ginseng: A guided input natural language search engine for querying ontologies. In *Jena User Conference*.

Patrick Blackburn and Johan Bos. 2005. *Representation and Inference for Natural Language. A First Course in Computational Semantics*. CSLI.

Olivier Bodenreider, Zeynep Hande Çoban, Mahir Can Doğanay, Esra Erdem, and Hilal Koşucu. 2008. A preliminary report on answering complex queries related to drug discovery using answer set programming. In *Proc. of ALPSWS*.

Boxer. 2009. `http://svn.ask.it.usyd.edu.au/trac/candc/wiki/boxer`.

CC. 2009. `http://svn.ask.it.usyd.edu.au/trac/candc/wiki`.

DLV. 2009. `http://www.dbai.tuwien.ac.at/proj/dlv`.

DLVHEX. 2009. `http://con.fusion.at/dlvhex/`.

DrugBank. 2008. `http://redpoll.pharmacy.ualberta.ca/drugbank/`.

Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. 2008. Discourse representation structures for ace 6.0. Technical Report IFI-2008.02, Department of Informatics, University of Zurich.

Michael Gelfond and Vladimir Lifschitz. 1988. The stable model semantics for logic programming. In Robert Kowalski and Kenneth Bowen, editors, *Logic Programming: Proceedings of the Fifth International Conference and Symposium*.

GeneNetwork. 2008. `http://humgen.med.uu.nl/~lude/genenetwork`.

GeneOntology. 2008. `http://www.geneontology.org`.

Jena. 2008. `http://jena.sourceforge.net/`.

Hans Kamp. 1981. A theory of truth and semantic representation. In J. A. G. Groenendijk, T. M. V. Janssen, and M. B. J. Stokhof, editors, *Formal Methods in the Study of Language*, volume 1, pages 277–322.

Esther Kaufmann, Abraham Bernstein, and Renato Zumstein. 2006. Querix: A natural language interface to query ontologies based on clarification dialogs. In *Proc. of ISWC*.

Mark Klein and Abraham Bernstein. 2004. Toward high-precision service retrieval. *IEEE Internet Computing*, 8(1):30–36.

Vladimir Lifschitz. 1999. Action languages, answer sets and planning. In *The Logic Programming Paradigm: a 25-Year Perspective*. Springer.

Victor Marek and Mirosław Truszczyński. 1999. Stable models and an alternative logic programming paradigm. In *The Logic Programming Paradigm: a 25-Year Perspective*. Springer.

MedicalSymptomsSignsDisease. 2008. `http://www.medicinenet.com/symptoms_and_signs/article.htm`.

Ilkka Niemelä. 1999. Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence*, 25.

PharmGKB. 2008. `http://www.pharmgkb.org`.

UniProt. 2008. `http://www.ebi.uniprot.org/index.shtml`.

Incorporating syntactic dependency information towards improved coding of lengthy medical concepts in clinical reports

Vijayaraghavan Bashyam, PhD*
Monster Worldwide Inc.
Mountain View, CA 94043
vbashyam@ucla.edu

Ricky K Taira, PhD
Medical Imaging Informatics Group
University of California, Los Angeles
Los Angeles, CA 90024
rtaira@mii.ucla.edu

Abstract

Medical concepts in clinical reports can be found with a high degree of variability of expression. Normalizing medical concepts to standardized vocabularies is a common way of accounting for this variability. One of the challenges in medical concept normalization is the difficulty in comparing two concepts which are orthographically different in representation but are identical in meaning. In this work we describe a method to compare medical phrases by utilizing the information found in syntactic dependencies. We collected a large corpus of radiology reports from our university medical center. A shallow semantic parser was used to identify anatomical phrases. We performed a series of transformations to convert the anatomical phrase into a normalized syntactic dependency representation. The new representation provides an easy intuitive way of comparing the phrases for the purpose of concept normalization.

1 Introduction

A vast amount of electronic information is generated in hospitals as a part of routine clinical care due to the adoption of the electronic medical record by health care centers in the United States (Berner *et al.*, 2005; Jha *et al.*, 2006). A significant portion of this information is in the form of unstructured free-text (Hall, 2000; Tange *et al.*, 1998). A free text representation makes it difficult for applications to accurately extract medical information for generic purposes (Ananiadou *et al.*, 2004). The problem of variability of expression in natural language expression has been well studied (Bates, 1986, 1989, 1998; Blair and Maron, 1985; Funk and Reid, 1983; Furnas *et al.*, 1984; Gomez *et al.*, 1990). In the medical domain in particular, users frequently express the same concept in different ways and different concepts in similar ways (Ananiadou and Nenadic, 2006). To illustrate, the terms *heart attack* and *cardiac attack* both refer to the same concept – *myocardial infarction*. Conversely the term *left lobe* could refer to the *left lobe of lung* or the *left lobe of liver* depending on the context (occurrence in a chest radiology report versus a gastro-intestinal radiology report). Such variability suggests a need to normalize concepts encountered in medical reports to a standard vocabulary in order to ensure interoperability.

Several standardized vocabularies exist in the medical domain such as the *Unified Medical Language System* (Humphreys and Lindberg, 1993), *Systematized Nomenclature of Medicine - Clinical Terms* (College of American Pathologists, July 2003), *Medical Subject Headings* (National Library of Medicine), and the *International Classification of Diseases* (World Health Organization). There have been several attempts in the past (Aronson, 2001; Bashyam and Taira, 2005; Bashyam *et al.*, 2007; Cooper and Miller, 1998; Friedman *et al.*, 2004; Nadkarni *et al.*, 2001; Oliver and Altman, 1994; Ruch *et al.*, 2003; Zou *et al.*, 2003) to map medical concepts to their standardized concept found in these terminologies. These approaches are based on mostly on lexical matching (Bashyam *et al.*, 2007), string matching (Nadkarni *et al.*, 2001), statistical indexing (Cooper and

* formerly with the Medical Imaging Informatics Group, Dept. of Radiological Sciences, University of California - Los Angeles, Los Angeles, CA 90024

Proceedings of the Workshop on BioNLP, pages 125–132,
Boulder, Colorado, June 2009.

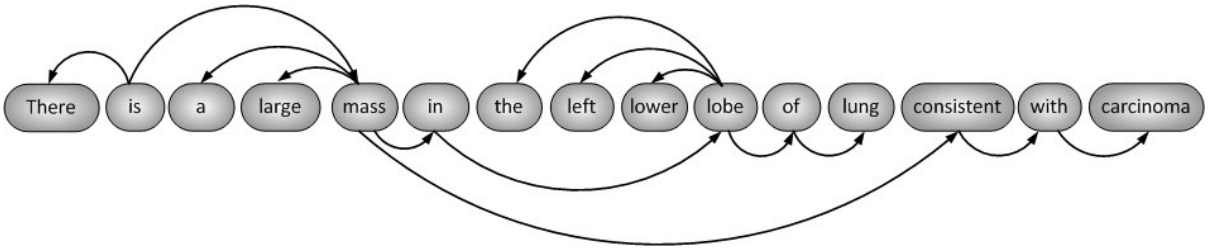


Figure 1. Example of a syntactic dependency parse tree with emphasis towards semantics. Each arc shows a dependency relation between a head and a modifier.

Miller, 1998), natural language processing (Aronson, 2001; Friedman *et al.*, 2004) information retrieval techniques (Bashyam and Taira, 2005; Oliver and Altman, 1994; Ruch *et al.*, 2003; Zou *et al.*, 2003) or a combination of these approaches (Cooper and Miller, 1998). These systems have managed to map a large percentage of medical terms to their respective standard terminologies in their reported experiments. While these systems have managed to perform satisfactorily for the task of normalizing simple expressions, they all acknowledge the larger problem of normalizing lengthy expressions. To illustrate, Nadkarni et al. (2001) mention the mapping of the phrase *spleen rupture and normal stomach* to the concept *stomach rupture* as a possible spurious mapping.

We hypothesize that using deep syntactic information can help in avoiding such spurious mapping. We describe a system which uses information found in syntactic dependencies to help in the coding of lengthy phrases. Preliminary results using this approach are reported as a proof-of-concept.

2 Background

Syntactic dependency parsing has received much focus from the natural language processing community (Eisner, 1996; Kudo and Matsumoto, 2000; Nivre and Scholz, 2004; Yamada and Matsumoto, 2003). A syntactic dependency relation is an asymmetric relation between two words. One word is called the head, and the other word is called the modifier or dependent. A word in the sentence can play the role of the head in several dependency relations (i.e., it can have several modifiers) but each word can play the role of the modifier only once. A special word, named the root, does not play the role of the modifier in any relation. The set of dependency relations that can be defined on a sentence form a tree, called the dependency tree. An example of dependencies in a typical sentence found in a radiology report is shown in Figure 1.

Systems based on syntactic dependencies have been used successfully in several information retrieval experiments with results outperforming traditional retrieval systems (Croft *et al.*, 1991; Gao *et al.*, 2004; Gonzalez *et al.*, 2005; Smeaton, 1986). In particular, this method has been used for word sense disambiguation (Lin, 1997) and thesaurus construction (Lin, 1998). Dependency trees have also been used for medical concept representation in the domains of radiology (Steimann, 1998) and pathology (Romacker *et al.*, 1999).

3 Methods

3.1 Anatomy Phrase Extraction

For identifying anatomy phrases, we use a specialized phrase parser trained to identify anatomy phrases within clinical reports. The input to the parser is a sentence tagged with a part-of-speech tag and a semantic tag. The lexical analyzer module of our NLP system takes a single sentence as the input and produces an output of word tokens tagged with their syntactic and semantic classes. The semantic tag is obtained by mapping tokens in a sentence to a taxonomy handcrafted for the domain of radiology reports custom built from radiology textbooks, radiology review manuals, radiology word compilations and published radiology glossaries apart from actual radiology reports (Taira *et al.*, 2001). Features of our implementation

include: 1) a large number (>450) of semantic classes as compared to lexical sources currently available allowing improved discrimination for tasks such as syntactic parsing, semantic interpretation and frame building; 2) the system recognizes special symbols including dates, medical abbreviations, medical coding symbols, numeric measurements, image slice references, and proper names; and 3) the system performs some word sense disambiguation using surrounding syntactic and semantic word features.

Our phrase parsing module currently targets anatomy phrases (e.g., *right upper lobe of lung*), existential relationships (e.g., *there is no evidence of*), and spatial relationships (e.g., *is located 1cm above*). We utilize a supervised learning approach to estimate the feature weights to a maximum entropy model which classifies words as the *start, inside, end, single,* or *outside* of a phrase boundary. A Viterbi dynamic programming algorithm is used to maximize the tag sequence probability. The anatomy phrase chunker has been tested on 4,500 sentences with recall and precision scores of 97.1% and 97.4% respectively.

3.2 Normalized Dependency Representation

We perform a series of transformations to convert an anatomical phrase from a *free-text* representation to a *normalized dependency vector space* representation. The following steps are taken in the representation conversion:

Syntactic Parsing

The anatomy phrase identified by the phrase parser preserves lexical information which is used to obtain a dependency parse tree using a full syntactic parser. This parser is based on a novel *field theory* approach to dependency parsing. The parser is strongly modeled for the radiology domain with performance accuracies of 84.9% and 89.9% for link precision and recall respectively for parsing whole sentences (Taira *et al.*, 2007). In comparison, the state-of-the-art parsers have performance accuracies in the low nineties for link precision and recall in the domain of newspaper text, with performance unknown in the domain of clinical text.

Link Reduction

Our system classifies dependency links into two types – bilexical links and trilexical links. A bilexical link is a strong dependency relation between two words (e.g. determiner←noun) whereas a trilexical link usually has a mediator word in between the two words (e.g. finding→in→location). When possible, a trilexical link is converted to a bilexical link by the elimination of the mediator word and the link type is tagged by the mediator word. The link type can play important roles in certain cases. In cases where the mediator word is also important, the trilexical link is considered as a pair of bilexical links.

Token Level Normalization

Once the parse tree is obtained, the tokens are normalized to their base form. The normalization is an approximate kind of lemmatization. However we also perform word level synonym normalization. For lemmatization, we use the Lexical Variant Generator tools developed by the National Library of Medicine for biomedical text (McCray *et al.*, 1994). For synonyms, we use a handcrafted lexicon built for the domain of radiology. This step helps in avoiding missing a mapping due to lexical differences due to pluralization, abbreviations and acronyms, case differences etc. This representation is referred to as the normalized dependency vector space representation

3.3 Mapping to a Terminology

The normalized dependency parse tree is represented as in a vector space as a *bag-of-links* as analogous to the so-called *bag-of-words* representation in conventional information retrieval. Two phrases can now be compared by using similarity measures such as cosine, dice, jaccard etc. within the dimension-space of dependency-links. One phrase can be the anatomy phrase in a clinical report and the other phrase can be an entry in a standardized terminology.

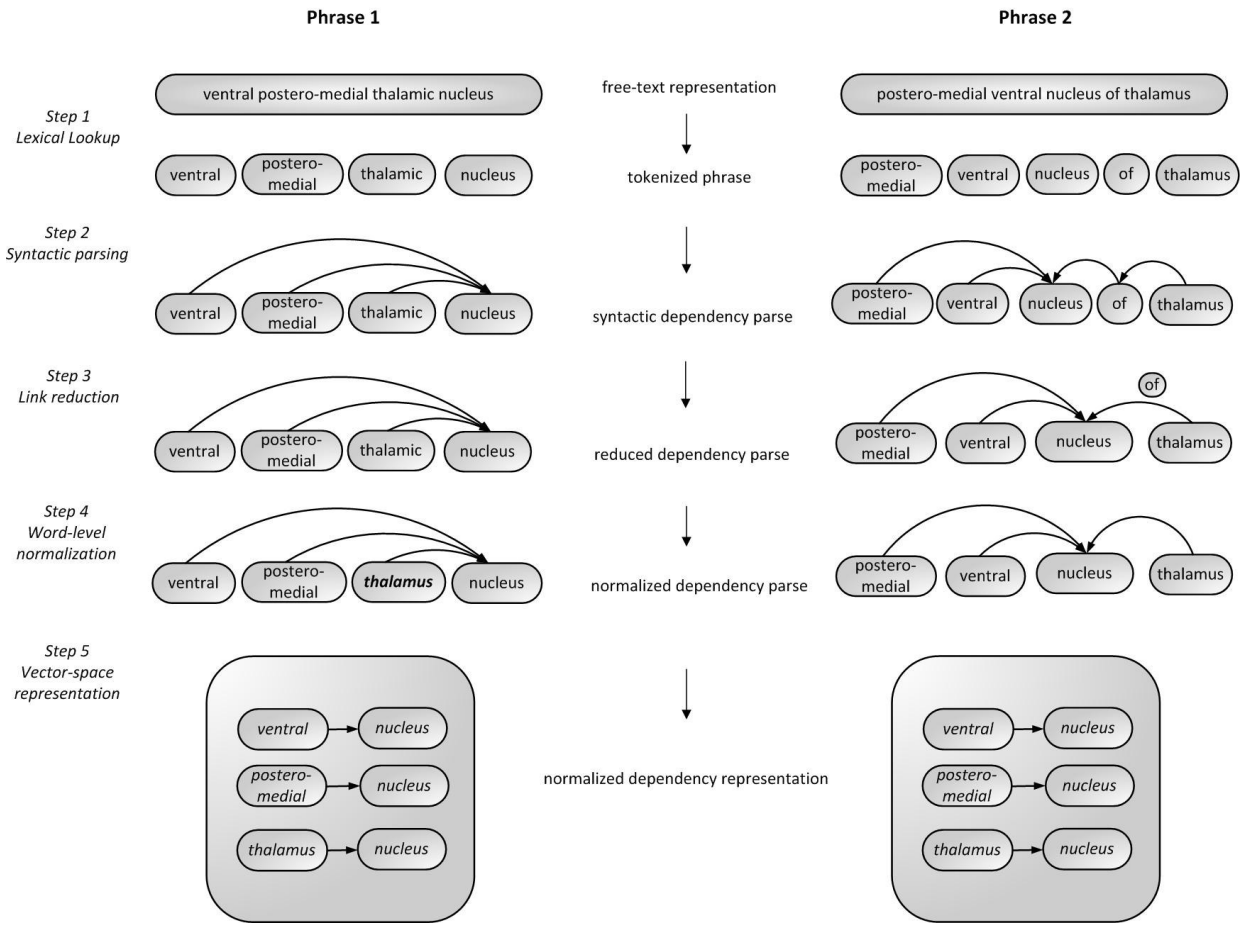


Figure 2. Example illustrating the transformation of a medical phrase from a free-text representation to a normalized syntactic dependency vector space representation.

An exercise in normalization is described in Figure 2 to illustrate how this method works. Consider the following phrase in a neuro-radiology report: *ventral postero-medial thalamic nucleus*. The corresponding concept in the target terminology is the phrase *postero-medial ventral nucleus of thalamus*. These phrases if compared by string matching will not result in direct matches. Permuting words and trying to compare rearrangements is complicated. In our approach, we first preprocess our terminology list and store it in a database. The preprocessing step is described in the right column (Phrase 2) of Figure 2. Starting with the phrase *postero-medial ventral nucleus* of thalamus, we first tokenize the individual words (lexical analysis) in the first step. In the second step, we parse the phrase to arrive at the dependency tree. In the third step, the trilexical link *nucleus←of←thalamus* is converted to a bilexical link by eliminating the word *of* and tagging it as the link type. In the following step, each word is normalized to its base form. In the fifth step, the phrase is represented as a *bag-of-links* and stored in a database. Similarly all the other phrases in our terminology are stored.

When the query phrase *ventral postero-medial thalamic nucleus* is compared against the terminology it undergoes the same processes previously described (Figure 2, Phrase 1). The importance of *word-normalization* can be seen here. In step 4, the word *thalamic* is normalized to *thalamus*. The final output is the *bag-of-links* representation. For con-

venience of comparison Figure 2 shows together, the query phrase and target phrase undergoing the various steps starting from a *bag-of-words* representation to a *bag-of-links* representation. It is clear that both phrases look identical in the final representation. While a string comparison would have missed equating the two in their original word-level representation, a comparison in the dependency vector space is likely to score them as a perfect match.

4 Experiment and Results

We obtained a set of 2500 neuro-radiology reports from our university medical center. Using the shallow semantic parser, we extracted a set of 2551 unique anatomical phrases. Of the 2551 phrases, 819 phrases were single worded terms. We discarded the single word terms. Single worded phrases do not fall into the difficult-to-map category which this method is specifically aiming to address. Moreover, a minimum of two words are required to define a syntactic dependency and thus the method is irrelevant for single worded terms. Thus we used only the 1732 multi-worded terms in our experiment. The average length of the multi-worded terms was 2.48 words.

We chose the UMLS, a coordinated repository of vocabularies as a target for concept coding. To reduce complexity, we removed non-English concepts and concepts outside the domain of neuro-radiology by filtering out unrelated concepts. Our final terminology had a size of about 100,000 entries. We preprocessed the entire terminology using the above mentioned steps and stored the dependency representation in a database. Every anatomy phrase was queried against this database and cosine similarity was used to measure relevance. No weighting system was employed although it is possible to weight links by their types. A physician domain expert manually evaluated the results of the 1732 queries for performance. Of the 1732 phrases, 1091 phrases (62.9% accuracy, 95% CI ±0.946%) were successfully matched. Since the target set is extremely large in size (as in any IR system), a recall analysis was not performed. A baseline comparison with MMTx (in phrase mode) resulted in 1051 phrases (60.68% accuracy, 95% CI ±0.49%) being mapped by MMTx. Table 1 summarizes the results.

MMTx Matched Phrases	Syn. Dependency Matched Phrases	
1051	1091	n=1732
60.68%	62.99%	
(±0.49%)	(±0.49%)	

Table 1. Overview of Results

5 Discussion

Analysis of the errors showed that the following error types resulted in the inability to match phrases perfectly:

Parsing without context:

A syntactic parser can parse a sentence and identify dependency relations in a sentence. However, when a phrase is given as an input, it is not always easy to parse a phrase and generate a dependency representation. There is context (remaining portions of the sentence) missing which is needed to unambiguously parse the phrase. In the case of anatomical phrases, our system was able to parse it because the source sentences from which they were extracted were available. However, in the case of the UMLS phrases, there is no such available information. Therefore manual parsing of several UMLS phrases had to be performed. One potential solution to this problem could be to identify MEDLINE sentences that contain these UMLS concepts and obtain a dependency parse tree using the context of the sentence.

Modular system architecture:

Since the system is modular, any errors in one of the modules (tokenization, word level normalization etc.) would result in the final dependency representation being imperfect. The specific errors we noticed were:

Parsing Errors:

Our parser has a higher accuracy for parsing phrases than whole sentences. However in this experiment, there were 37 instances where it failed in

assigning the correct links. This resulted in partial matches.

Word Normalization Errors:

There is a natural ambiguity introduced when words are normalized to their base forms. Words with completely different senses can have the same root form (e.g. *left←leaves* and *left←left* (spatial direction)). Similarly, a word can have different normalized forms depending on the sense (e.g. *leaf←leaves* and *left←leaves*). A robust method for word-level normalization is desired that can also perform word-sense disambiguation. Currently the NLM's word level normalization tool is being used which is not perfect and therefore errors introduced due to this module result in the entire phrase being transformed incorrectly or ambiguously. The ideal word level normalization will result in the words *cancer, cancerous, carcinoma* all conflating to the same word which is beyond purely morphological analysis.

Link Reduction Errors:

Not all relations manifest as simple bilexical and trilexical links. Some relations are tetralexical and although they can be reduced effectively to bilexical links, the methodology needs to be investigated. To illustrate, consider the phrases '*mass consistent with cancer*' and '*cancerous mass*' parsed as

mass←consistent←with←cancer
cancerous→mass.

The former is parsed as four words with three links. To convert it into a bilexical link, the words '*consistent'* and '*with'* need to be: (1) clustered as a single token and (2) eliminated by transferring it to the link as a label. This is a more complicated process and we still haven't explored such abstractions. A robust rule based link reduction system is desired to handle such cases.

Another limitation of this method is that the heuristic rules for link reduction may not be applicable outside the radiology domain. Finally, syntactic dependency parsers are built using computationally complex algorithms. Thus while using them can result in advanced language understanding, they may not be suitable for real-time applications. There is always a tradeoff between accuracy and speed and it remains to be seen if robust low complexity parsers can be developed.

The inability to perform a recall analysis also make is difficult to judge the theoretical best performance. That is, it is quite likely that there are many phrases in our dataset that do not have a corresponding UMLS concept. Performing a recall analysis would help in determining this.

While we noticed several areas of improvement in our system, we were encouraged by the comparison of the overall results of our system to that of MMTx. We did not do an error analysis of MMTx since several previous publications have documented the various kinds of errors in MMTx (Bashyam *et al.*, 2007; Divita *et al.*, 2004; Meng *et al.*, 2005). Our idea is to provide a baseline comparison showing that our approach performs comparably if not better than MMTx which is the most commonly used[1] tool for concept coding. To our knowledge this the first time syntactic dependencies have been used for this task, Previous attempts have relied purely on shallow parsers.

6 Future Work

Increasing the robustness of the individual modules is a primary requirement for further experiments to prevent the *weakest link effect* cascading to the final output. Specifically we plan to work towards a robust word level normalization system. Additionally, robust evaluation methods including comparisons with other techniques will be investigated.

7 Conclusion

Syntactic dependency based methods for medical concept coding show promise. While some of the described implementations are specific to domain (radiology) and phrase type (anatomy), it is expected that the principle is general enough to be applied in other domains as well.

[1] For an overview of recent applications of MMTx, see (Bashyam *et al.*, 2007)

Acknowledgements

The authors would like to thank Lew Andrada, Gregory Leazer, Jonathan Furner and Christine Borgman for several useful suggestions.

This work was supported in part by the following grants:

1. National Institute of Biomedical Imaging and Bioengineering P01-EB00216
2. National Institute of Health R01-EB002247

References

Ananiadou, S., Friedman, C., and Tsujii, J. (2004). Introduction: named entity recognition in biomedicine. Journal of Biomedical Informatics, 37(6), 393-395.

Ananiadou, Sophia, and Nenadic, Goran. (2006). Automatic Terminology Management in Biomedicine. In Sophia Ananiadou and John McNaught (Eds.), Text Mining for Biology And Biomedicine. Norwood MA: Artech House.

Aronson, A. R. (2001). Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program, AMIA Annu Symp Proc (pp. 17-21). Portland, OR.

Bashyam, V., and Taira, R. K. (2005). Indexing Anatomical Phrases in Neuro-Radiology Reports to the UMLS 2005AA, American Medical Informatics Association Annual Symposium (pp. 26-30). Washington DC.

Bashyam, Vijayaraghavan, Divita, Guy, Bennett, David B, Browne, Allen C, and Taira, Ricky K. (2007, August 2007). A Normalized Lexical Lookup Approach to identifying UMLS concepts in free text. Paper presented at the 12th World Congress on Health (Medical) Informatics (MEDINFO). Brisbane, Australia.

Bates, M. J. (1986). Subject access in online catalogs: A design model. Journal of the American Society for Information Science, 37(6), 357-376.

Bates, M. J. (1989). Rethinking Subject Cataloging in the Online Environment. Library Resources and Technical Services, 33(4), p400-412.

Bates, M. J. (1998). Indexing and access for digital libraries and the internet: Human, database, and domain factors. Journal of the American Society for Information Science, 49(13), 1185-1205.

Berner, E. S., Detmer, D. E., and Simborg, D. (2005). Will the Wave Finally Break? A Brief View of the Adoption of Electronic Medical Records in the United States. J Am Med Inform Assoc, 12(1).

Blair, David C., and Maron, M. E. (1985). An Evaluation of retrieval effectiveness for a full-text document retrieval system. Communications of the ACM, 28(3), 289-299.

College of American Pathologists. (July 2003). SNOMED Clinical Terms ® Technical Reference Guide.

Cooper, G. F., and Miller, R. A. (1998). An Experiment Comparing Lexical and Statistical Methods for Extracting MeSH Terms from Clinical Free Text, Proceedings of the American Medical Informatics Association Annual Symposium.

Croft, W. B., Turtle, H. R., and Lewis, D. D. (1991). The use of phrases and structured queries in information retrieval. Paper presented at the Proc of the 14th Annu Intnl ACM SIGIR Conf on R&D in Information Retrieval.

Divita, Guy, Tse, Tony, and Roth, Laura. (2004). Failure analysis of MetaMap transfer (MMTx). In M Fieschi et. al. (Ed.), Proceedings of the 11 World Congress on Health Informatics (MEDINFO 2004).

Eisner, J. (1996). Three new probabilistic models for dependency parsing: An exploration. Paper presented at the Proc COLING.

Friedman, C., Shagina, L., Lussier, Y., and Hripcsak, G. (2004). Automated Encoding of Clinical Documents Based on Natural Language Processing. Journal of the American Medical Informatics Association, 11(5), 392.

Funk, M. E., and Reid, C. A. (1983). Indexing consistency in MEDLINE. Bulletin of the Medical Library Association, 71(2), 176-183.

Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1984). Statistical semantics: analysis of the potential performance of keyword information systems. In Human factors in computer systems (pp. 187-242): Ablex Publishing Corp. Norwood, NJ, USA.

Gao, J., Nie, J. Y., Wu, G., and Cao, G. (2004). Dependence language model for information retrieval. Paper presented at the Proc ACM SIGIR.

Gomez, L. M., Lochbaum, C. C., and Landauer, T. K. (1990). All the right words: Finding what you want as a function of richness of indexing vocabulary. Journal of the American Society for Information Science, 41(8), 547-559.

Gonzalez, M., Strube De Lima, V. L., and Valdeni De Lima, J. (2005). Binary Lexical Relations for Text Representation in Information Retrieval. In Natural Language Processing and Information Systems (Vol. 3513, pp. 21-31). Berlin / Heidelberg: Springer

Hall, F. (2000). Language of the radiology report. American Journal of Roentology, 175, 1239-1241.

Humphreys, B. L., and Lindberg, D. A. B. (1993). The UMLS project: making the conceptual connection between users and the information they need. Bulletin of the Medical Library Association, 81(2), 170.

Jha, Ashish K., Ferris, Timothy G., Donelan, Karen, DesRoches, Catherine, Shields, Alexandra, Rosenbaum, Sara, and Blumenthal, David. (2006). How Common Are Electronic Health Records In The United States? A Summary Of The Evidence. Health Affairs, doi: 10.1377/hlthaff.25.w496.

Kudo, T., and Matsumoto, Y. (2000). Japanese dependency structure analysis based on support vector machines. Paper presented at the Proc International Conference On Computational Linguistics.

Lin, D. (1997). Using syntactic dependency as local context to resolve word sense ambiguity, Proceedings of the 35th Annual Meeting of the Association for ComputationalLinguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics. Madrid, Spain.

Lin, D. (1998). Automatic retrieval and clustering of similar words. Paper presented at the Proc 36th Annu Meeting of the ACL and 17th Intnl Conf on Comp Ling.

McCray, AT, Srinivasan, S, and Browne, AC. (1994). Lexical methods for managing variation in biomedical terminologies. Paper presented at the Proceedings of the Annual Symposium on Computer Applications in Medical Care.

Meng, F., D'Avolio, L. W., Chen, A. A., Taira, R. K., and Kangarloo, H. (2005). Generating Models of Surgical Procedures using UMLS Concepts and Multiple Sequence Alignment, AMIA Annu Symp Proc (Vol. 520, pp. 4). Washington, DC.

Nadkarni, P., Chen, R., and Brandt, C. (2001). UMLS Concept Indexing for Production Databases: A Feasibility Study. Journal of the American Medical Informatics Association, 8(1), 80.

National Library of Medicine. MEdical Subject Headings. www.nlm.nih.gov/mesh/ meshhome.html Last Accessed 09 March 2007 Bethesda, MD.

Nivre, J., and Scholz, M. (2004). Deterministic dependency parsing of English text. Paper presented at the Proc COLING.

Oliver, D. E., and Altman, R. B. (1994). Extraction of SNOMED concepts from medical record texts, Proc of the 18th Annual SCAMC (Vol. 17983). Washington: McGraw Hill.

Romacker, M., Schulz, S., and Hahn, U. (1999). Streamlining semantic interpretation for medical narratives. Paper presented at the AMIA Annu Symp Proc.

Ruch, P., Baud, R., and Geissbühler, A. (2003). Learning-free Text Categorization, Proc of 9th Conf on Artificial Intelligence in Medicine (pp. 199–204).

Smeaton, A. F. (1986). Incorporating syntactic information into a document retrieval strategy: an investigation. Paper presented at the Proc of the 1986 ACM Conf on R&D in Information Retrieval, Pisa, Italy.

Steimann, F. (1998). Dependency parsing for medical language and concept representation. Artificial Intelligence in Medicine, 12(1), 77-86.

Taira, R. K., Soderland, S. G., and Jakobovits, R. M. (2001). Automatic Structuring of Radiology Free-Text Reports. Radiology, 21, 237-245.

Taira, Ricky K, Bashyam, Vijayaraghavan, and Kangarloo, Hooshang. (2007). A Field Theoretical Approach for Medical Natural Language Processing. IEEE Transactions on IT in Biomedicine, 11(4), 364-375.

Tange, H.J., Schouten, H.C., Kester, A.D.M., and Hasman, A. (1998). The Granularity of Medical Narratives and Its Effect on the Speed and Completeness of Information Retrieval. J Am Med Inform Assoc, 5(6), 571-582.

World Health Organization. http://www.who.int/classifications/icd/en/ Last Accessed 09 March 2007.

Yamada, H., and Matsumoto, Y. (2003). Statistical dependency analysis with support vector machines. Paper presented at the Proceedings of the 8th International Workshop on Parsing Technologies (IWPT).

Zou, Q., Chu, W. W., Morioka, C., Leazer, G. H., and Kangarloo, H. (2003). IndexFinder: A Method of Extracting Key Concepts from Clinical Texts for Indexing, AMIA Annu Symp Proc (Vol. 2003, pp. 763-767).

Identifying Interaction Sentences from Biological Literature Using Automatically Extracted Patterns

Haibin Liu
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
haibin@cs.dal.ca

Christian Blouin
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
cblouin@cs.dal.ca

Vlado Kešelj
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
vlado@cs.dal.ca

Abstract

An important task in information retrieval is to identify sentences that contain important relationships between key concepts. In this work, we propose a novel approach to automatically extract sentence patterns that contain interactions involving concepts of molecular biology. A pattern is defined in this work as a sequence of specialized Part-of-Speech (POS) tags that capture the structure of key sentences in the scientific literature. Each candidate sentence for the classification task is encoded as a POS array and then aligned to a collection of pre-extracted patterns. The quality of the alignment is expressed as a pairwise alignment score. The most innovative component of this work is the use of a Genetic Algorithm (GA) to maximize the classification performance of the alignment scoring scheme. The system achieves an F-score of 0.834 in identifying sentences which describe interactions between biological entities. This performance is mostly affected by the quality of the preprocessing steps such as term identification and POS tagging.

1 Introduction

Recent research in information extraction (IE) in biological science has focused on extracting information about interactions between biological entities from research communications. The type of interaction of interest includes protein-protein, protein-DNA, gene regulations and other interactions between macromolecules. This work broadens the definition of the term "interaction" to include other types of concepts that are semantically related to cellular components and processes. This contrasts with the past efforts focusing strictly on molecular interactions (Blaschke et al., 1999; Ono et al., 2001). We anticipate that identifying the relationships between concepts of molecular biology will facilitate the building of knowledge models, improve the sensitivity of IE tasks and ultimately facilitate the formulation of new hypothesis by experimentalists.

The extraction of interactions is based on the heuristic premise that interacting concepts co-occur within a given section of text. The challenge is that co-occurrence certainly does not guarantee that a passage contains an interaction(Jang et al., 2006; Skusa et al., 2005). Co-occurrence is highly dependent on the definition of the section of text within which the target terms are expected to be found. A thorough comparison on the prediction of protein-protein interaction between abstract-level co-occurrence and sentence-level co-occurrence was undertaken (Raychaudhuri, 2006). It is demonstrated that abstract co-occurrence is more sensitive but less specific for interactions. At the cost of wide coverage, sentence co-occurrence increases the accuracy of interaction prediction. Since the ultimate goal of IE is to extract knowledge and accuracy is the most important aspect in evaluating the performance of such systems, it makes sense to focus the effort in seeking interaction sentences rather than passages or abstracts. Not every co-occurrence in sentences implies a relationship that expresses a fact. In the 2005 Genomics Track dataset, 50% of all sentence co-occurrences of entities correspond to definite relationships while the rest of the co-occurrences only convey some possible relationships or contain no relationship of interest (Li et al., 2005). Therefore, more sophisticated text mining strategies are required to classify sentences that describe interactions between co-occurring concepts.

In the BioCreative II challenge [1], teams were asked to determine whether a given passage of text contained information about the interaction between two proteins. This classification task worked at the abstract level and the interacting protein pairs were not required to be extracted. The task for the Learning Language in Logic

[1] http://biocreative.sourceforge.net/

Proceedings of the Workshop on BioNLP, pages 133–141,
Boulder, Colorado, June 2009.

(LLL'05) challenge [2] was to build systems that extract interactions between genes or proteins from biological literature. From individual sentences annotated with agent-target relations, patterns or models had to be learned to extract these interactions. The task focused on extracting only the interacting partners. The context of an interaction may also be critical to the validity of the extracted knowledge since not all statements found in the literature are always true.

In this work, we propose an approach to automatically extract patterns containing relevant interaction between biological concepts. This extraction is based on the assumption that biological interactions are articulated by a limited number of POS patterns embedded in sentences where entities/concepts are co-occurring. The extracted patterns are then applied to identify interaction sentences which describe interactions between biological entities. Our work aims to identify precise sentences rather than passages. Because of the nature of the patterns, we hope that some of the contextual information present in interaction sentences also play a role in the classification task.

The rest of the paper is organized as follows: In Section 2, we review recent research advances in extracting biological interactions. Section 3 describes an experimental system designed for our work. Sections 4, 5 and 6 elaborate the approaches and algorithms. Performance is evaluated in Section 7. Finally, Section 8 summarizes the paper and introduces future work.

2 Related work

Early on, Blaschke (Blaschke et al., 1999) employed patterns to predict the presence of a protein-protein interaction. A series of patterns was developed manually to cover the most obvious descriptions of protein functions. This process was based on a set of keywords, including interaction verbs, that are commonly used to describe this type of interaction. A sentence extraction system BioIE (Divoli and Attwood, 2005) also uses patterns to extract entire sentences related to protein families, protein structures, functions and diseases. The patterns were manually defined and consisted of single words, word pairs, and small phrases.

Although systems relying on hand-coded patterns have achieved some success in extracting biological interactions, the strict requirement of dedicated expert work is problematic. Moreover, each type of interaction may require a definition of many different patterns including different arrangements and different variants of the same keyword. Manually encoding all patterns encountered in a corpus is time-consuming and potentially impractical in real applications. Thus, automatically learning such patterns is an attractive solution.

An approach which combines dynamic programming and sequence alignment algorithms as normally used for the comparison between nucleotide sequences was introduced by Huang *et al.* (Huang et al., 2004). This approach is designed to generate patterns useful for extracting protein-protein interactions. The main problem with this approach is that the scoring scheme that is required to implement the alignment algorithm is difficult to define and contains a potentially large number of free parameters. We propose a method based on Genetic Algorithm (GA) heuristics to maximize the alignment procedure for the purpose of classification. GAs were also used as a learning strategy to train finite state automata for finding biological relation patterns in texts(Plake et al., 2005). It was reported (Bunescu et al., 2005; Hakenberg et al., 2005) that automatically learned patterns identify biological interactions even more accurately than hand-coded patterns.

3 Overview of system design

In this work, we have designed an experimental system to facilitate the automatic extraction of biological interaction patterns and the identification of interaction sentences. It consists of three major modules: biological text preprocessing, interaction pattern extraction, and interaction sentence identification.

Biological text preprocessing reformats the original biological texts into candidate sentences. A pattern learning method is then proposed to automatically extract the representative patterns of biological interactions. The obtained patterns are further used to identify instances that evidently describe biological interactions. Poor performance during preprocessing will have detrimental effects on later stages. In the following sections, we will describe each component.

4 Biological text preprocessing

4.1 Sentence preparation

A heuristic method is implemented to detect sentence boundaries (Mikheev, 2002) based on the assumption that sentences are usually demarcated by some indicative delimiting punctuation marks in order to segment the biological texts into sentence units. Captions and headings that are not grammatically valid sentences are therefore detected and further eliminated for our work.

[2] http://genome.jouy.inra.fr/texte/LLLchallenge/

4.2 Part-of-Speech tagging

POS tagging is then performed to associate each word in a sentence with its most likely POS tag. Because subsequent processing steps typically depend on the tagger's output, high performance at this level is crucial for success in later stages. A statistical tagger Lingua::EN::Tagger [3] is used to perform this task.

4.3 Biological term annotation

A learning-based biological term annotation system, ABTA (Jiampojamarn et al., 2005), is embedded in our system. The type of terms includes molecules, such as genes, proteins and cell lines, and also biological processes. Examples of biological processes as entities are: "T cell activation" and "IL-2 gene transcription". We consider that a broader definition of biological term will include more facts from literature, thus leading to more general use of interaction patterns for IE tasks.

ABTA considers the longest expression and ignores embedded entities. Further, instead of distinguishing terms from their relevant biology concepts, a unified tag "*BIO*" is assigned to all the identified terms. We aim to discover patterns of the general interactions between biological concepts, not only the interactions between molecules, e.g., protein-protein interaction.

Tags like *NN*(noun) and *VB*(verb) are typically used to define entities and the action type of interactions, and thus they are indispensable. However, tags such as *JJ*(adjective) and *RB*(adverb) could occur at different positions in a sentence. We decided to remove these tags to prevent the combinatorial effect that these would induce within the set of extracted patterns.

4.4 Text chunking

Next, a rule-based text chunker (Ramshaw and Marcus, 1995) is applied on the tagged sentences to further identify phrasal units, such as base noun phrases *NP* and verbal units *VB*. This allows us to focus on the holistic structure of each sentence. Text chunking is not applied on the identified biological terms. In order to achieve more generalized interaction patterns, a unified tag "*VB*" is used to represent every verbal unit instead of employing different tags for various tenses of verbs.

As a result of preprocessing, every sentence is represented by its generalized form as a sequence of corresponding tags consisting of POS tags and predefined tags. Table 1 summarizes the main tags in the system.

A biological interaction tends to involve at least three objects: a pair of co-occurring biological entities connected by a verb which specifies the action type of the interaction. Thus, a constraint is applied that only sentences satisfying form "BioEntity A – Verb – BioEntity B" will be preserved as candidate sentences to be further processed in the system. It is possible that the presence of two entities in different sentence structures implies a relationship. However, this work assumes the underlying co-occurrence of two concepts and a verb in the interest of improving the classification accuracy.

Tag name	Tag description	Tag type
BIO	Biological entity	Predefined
NP	Base noun phrase	Predefined
VB	Verbal unit	Predefined
IN	Preposition	POS
CC	Coordinating conjunction	POS
TO	to	POS
PPC	Punctuation comma	POS
PRP	Possessive 2nd determiner	POS
DET	Determiner	POS
POS	Possessive	POS

Table 1: Main tags used in the system

The obtained candidate sentences are split into training and testing sets. The training set is used to extract the representative patterns of biological interactions. The testing set is prepared for identifying sentences that evidently describe biological interactions.

5 Interaction pattern extraction

5.1 PATRICIA trees

The method we propose to extract interaction patterns from candidate sentences is based on the use of PATRICIA trees (Morrison, 1968). A PATRICIA tree uses path compression by grouping common sequences into nodes. This structure provides an efficient way of storing values while maintaining the lookup time for a key of O(N). It has been applied to many large information retrieval problems (Chien, 1997; Chen et al., 1998).

In our work, a PATRICIA tree is used for the first time to facilitate the automatic extraction of interaction patterns. All training sentences are inserted and stored in a generic PATRICIA tree from which the common patterns of POS tags can be efficiently stored and the tree structure used to compute relevant usage statistics.

5.2 Potential pattern extraction

Patterns of straightforward biological interactions are frequently encountered in a range of actual sentences. Conversely, vague relationships or complex interactions patterns are seldom repeated. Therefore, the

[3] http://search.cpan.org/~acoburn

premise of this work is that there is a set of frequently occurring interaction patterns that matches a majority of stated facts about molecular biology. In this work, a *biological interaction pattern* is defined as follows:

Definition 5.1. A biological interaction pattern bip is a sequence of tags defined in Table 1 that captures an aggregate view of the description of certain types of biological interactions based on the consistently repeated occurrences of this sequence of tags in different interaction sentences. $BIP = \{bip_1, bip_2, \cdots, bip_k\}$ represents the set of biological interaction patterns.

We first extract potential interaction patterns by populating a PATRICIA tree using training sentences. Every node in the tree contains one or more system tags, which is the preceding tag sequence of its descendant nodes in each sentence. Every sentence is composed of a path of system tags from the root to a leaf. Hence, we propose that the sequence of system tags that can be formed from traversing the nodes of the tree is a potential pattern of biological interactions. At the same time, the occurrence frequency of each pattern is also retrieved from the traversal of tree nodes.

A predefined frequency threshold f_{min} is used as a constraint to filter out patterns that occur less than f_{min} times. It has been demonstrated that if an interaction is well recognized, it will be consistently repeated (Blaschke et al., 1999; Ono et al., 2001). The generalization and the usability of patterns can be controlled by tuning f_{min}. Further, some filtering rules are adapted to control the form of a pattern and enhance the quality of the discovered patterns, such as if a pattern ends with a tag *IN, VB, CC* or *TO*, the pattern will be rejected. Flexibility in setting this threshold can be applied to meet special demands. Algorithm 1 shows our pattern learning method which has a time complexity of $O(n)$ in the size of candidate sentences, n.

Algorithm 1 Patricia-Tree-based Extraction of Biological Interaction Patterns

Input: Candidate Sentences $CS \in$ Biological text; a predefined threshold f_{min}; a set of filtering rules FR
Output: BIP : Set of biological interaction patterns

```
BIP ← ∅;  PT ← ∅                          //PT : Patricia Trie
for all sentences s ∈ CS do
    PT ← Insert(s)                        //Populating Patricia Tree
for all nodes n_i ∈ PT do
    bip_i ← Pattern(n_i)                  //Concatenating tags in nodes
    from root to n_i, which is a potential pattern
    if Count(bip_i) ≥ f_min and bip_i does not meet FR
    then
        //Count(bip_i) returns No. of occurrences of bip_i;
        BIP ← bip_i
```

5.3 Interaction verb mining

Although the obtained patterns are derived from the candidate sentences possessing the form "BioEntity A – Verb – BioEntity B", some of them may not contain facts about biological interactions. This is possible if the action verbs do not describe an interaction. Quite a few verbs, such as "report", "believe", and "discover", only serve a narrative discourse purpose. Therefore, mining the correct interaction verbs becomes an important step in the automatic discovery of patterns. We decided to perform the method applied in (Huang et al., 2004) to mine a list of interaction verbs. This will be used to further improve the relevance of achieved patterns by filtering out patterns formed by the sentences in which the action verbs are not on the list.

6 Interaction sentence identification

Once the biological interaction patterns are obtained, we perform interaction sentence identification on testing sentences. For our work, they are partitioned into two sets: interaction sentences which explicitly discuss interactions between entities, and non-interaction sentences which do not describe interactions, or merely imply some vague relationships between entities. The task of interaction sentence identification is treated as a classification problem to differentiate between interaction sentences and non-interaction sentences.

6.1 Pattern matching scoring

We first perform pattern matching by iteratively applying the interaction patterns to each testing sentence. This is done using sequence alignment which calculates the degree of the similarity of a sentence to an interaction pattern. Since patterns capture various ways of expressing interactions among sentences, a high similarity between an interaction sentence and a pattern is expected. Therefore, we conjecture that the alignment scores can be used to discriminate some type of interaction sentences from other types of sentences.

The scoring scheme involved in the pattern matching consists of penalties for introducing gaps, match rewards and mismatch penalties for different system tag pairs. Table 2 presents an example scoring scheme for main tags. Penalties and rewards are denoted respectively by negative and positive values.

As a variation of global alignment, an end-space free alignment algorithm is implemented to facilitate the alignment between patterns and testing sentences. The shortest pattern is always preferred for a sentence in case that same alignment score is achieved by multiple

Tag	Gap	Match	Mismatch
BIO	-10	+8	-3
NP	-8	+6	-3
VB	-7	+7	-3
IN	-6	+5	-1
CC	-6	+5	-1
TO	-1	+5	-1
PPC	-1	+3	-1
PRP	-1	+3	-1
DET	-1	+3	-1
POS	-1	+3	-1

Table 2: An alignment scoring scheme for system tags

patterns. As a result, each sentence is assigned to its most appropriate pattern along with a maximum alignment score. Therefore, an interaction sentence will be highlighted with a high alignment score by its most similar interaction pattern, while a non-interaction sentence will be characterized by a low alignment score indicating rejections by all patterns. Essentially, this procedure can be seen as a variation of the well-known k Nearest Neighbors classification method, with $k = 1$.

6.2 Performance evaluation

We then evaluate whether the alignment scores can be used to classify the testing sentences. We have proposed two independent evaluation measures: statistical analysis (SA) and classification accuracy (AC).

SA measures whether the scoring difference between the mean of interaction sentences and the mean of non-interaction sentences is statistically significant. If the difference is significant, there will be a tendency that interaction sentences outscore non-interaction sentences in alignment. Hence, it would be reliable to use alignment scores to classify testing sentences. Although non-interaction sentences could come from the same documents as interaction sentences and discuss concepts that are associated with the target interactions, we assume that interaction sentences and non-interaction sentences are two independent samples.

The statistical two-sample z test (Freund and Perles, 2006) is performed with the null hypothesis that there is no scoring difference between the means of interaction and non-interaction sentences. A comparatively large z will lead to the rejection of the null hypothesis. Naturally, the increase of z value will increase the difference between the means and therefore conceptually keep pushing the overall scoring distributions of two samples further away from each other. Consequently, interaction sentences can be separated from non-interaction sentences according to alignment scores. In reality, the distinction between interaction and non-interaction sentences is not absolute. Thus, the scoring distributions of two samples can only be distanced by a certain maximum value of z depending on the scoring scheme applied in pattern matching.

Conversely, AC measures the proportion of correctly classified testing sentences, including both interaction and non-interaction sentences, to the total testing sentences. An appropriate threshold T is determined for obtained alignment scores to differentiate between interaction and non-interaction sentences, and to facilitate the calculation of classification accuracy.

It is not possible to evaluate the performance without correctly pre-labeled testing sentences. We decided to manually classify the testing sentences in advance by assigning each sentence an appropriate label of interaction or non-interaction. This work was done by two independent experts, both with Ph.D. degrees in molecular biology or a related discipline.

6.3 Scoring scheme optimization

The scoring scheme applied in pattern matching has a crucial impact on the performance of interaction sentence identification. An interesting problem is whether there exists an optimal scoring scheme covering the costs of gap, match and mismatch for different system tags in the pattern matching alignment, which is destined to achieve the best performance on classifying testing sentences. To the best of our knowledge, no efforts have been made to investigate this problem. Instead, an empirical or arbitrary scoring scheme was adopted in previous research for the pairwise alignments (Huang et al., 2004; Hakenberg et al., 2005). We have proved that the problem is NP-hard by reducing a well-known NP-hard problem 3-SAT to this problem. The proof is not presented in this work.

A genetic algorithm (GA) is used as a heuristic method to optimize parameters of the scoring scheme for sentence classification. The costs of penalties and rewards for different system tags are encoded by integer values within two predefined ranges: [-50, 0) and (0, 50], and assembled as a potential solution of scoring scheme, which consists of 30 parameters covering the costs for tags in the alignment as listed in Table 2. The two evaluation measures SA and AC are used as the fitness function for GA respectively with the goal of maximizing z value or classification accuracy.

GA is set up to evolve for 100 generations, each of which consists of a population of 100 potential solutions of scoring scheme. GA starts with a randomly

generated population of 100 potential solutions and proceeds until 100 generations are reached. The number of generations and the population size are decided with consideration of the runtime cost of evaluating the fitness function, which requires running the scoring algorithm with each sentence. A large number of generations or a large population size would incur an expensive runtime cost of evaluation.

In addition, we further divide the labeled set of candidate sentences into two subsets: The first dataset is used to optimize parameters of the scoring scheme, while the second dataset, testing set, is used to test the achieved scheme on the task of sentence classification.

7 Results and evaluation

7.1 Dataset

Our experiments have been conducted on Genia corpus (v3.02) [4], the largest, publicly available corpus in molecular biology domain. It consists of 2,000 biological research paper abstracts and is intended to cover biological reactions concerning transcription factors in human blood cells. The information of sentence segmentation, word tokenization, POS tagging and biological term annotation is also encoded in the corpus.

7.2 Biological text preprocessing results

Evaluated using the inherently equipped annotation information, our system achieves nearly 99% accuracy on segmenting sentences. Further, it obtains an overall POS tagging accuracy of 91.0% on 364,208 individual words. We noticed that the tagging information encoded in Genia corpus is not always consistent throughout the whole corpus, thus introducing detrimental effects on the tagging performance. Also, considering that the tagger is parameterized according to the general English domain, porting this tagger to the biology domain is accompanied by some loss in performance.

The system reaches an F-score of 0.705 on annotating all biological terms including both multi-word and single word terms. After performing text chunking, the system produces a set of candidate sentences. We further perform text chunking on Genia corpus based on its encoded annotations and use the resulting set of sentences for the subsequent experiments to provide a gold standard to which results produced based on our system annotations can be compared. Table 3 presents some statistics of the preprocessed dataset. For each type of annotations, we randomized the candidate sentence set and chose 12,525 candidate sentences as the training set to extract biological interaction patterns. The rest of candidate sentences are prepared as the testing set.

Attributes	Genia	Our system
Total preprocessed sentences	18,545	18,355
Candidate sentences	16,272	17,525
Training set sentences	12,525	12,525
Testing set sentences	6,020	5,000

Table 3: Statistics of experimental dataset

7.3 Interaction pattern extraction results

$f_{min} = 5$ is used to filter out the potential patterns that appear less than 5 times in the training set. Evaluated by domain experts, lists of 300 interaction verbs and 700 non-interaction verbs are obtained from 12,525 training sentences with Genia annotations. Inflectional variants of the verbs are also added into the lists.

Refined by the filtering rules and the interaction verbs, a final set of representative patterns of biological interactions are obtained from Algorithm 1. We performed our proposed pattern learning method on training sentences of both the GENIA and our own annotations. There are respectively 241 and 329 potential patterns. Of these, 209 and 302 were extracted. Interestingly, only 97 extracted patterns are common to both annotation schemes.

Table 4 lists the 10 most frequent interaction patterns based on Genia annotations. For instance, a training sentence conforming to the second pattern is "The expression of the QR gene is regulated by the transcription factor AP-1." (MEDLINE: 96146856).

Pattern count	Pattern
264	*BIO VB BIO IN BIO*
261	*NP IN BIO VB IN BIO*
182	*NP IN BIO VB BIO*
162	*BIO IN BIO VB IN BIO*
160	*BIO VB IN BIO IN BIO*
143	*NP IN BIO VB IN NP IN BIO*
142	*NP VB IN BIO VB BIO*
138	*PRP VB IN BIO VB BIO*
126	*BIO VB NP IN BIO IN BIO*
121	*NP IN BIO VB NP IN BIO*

Table 4: Extracted Biological Interaction Patterns

7.4 Interaction sentence identification results

Since the total testing sentence set is large, we decided to randomly extract 400 sentences from it as the sample set for our task. The 400 sentences were manu-

[4] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/

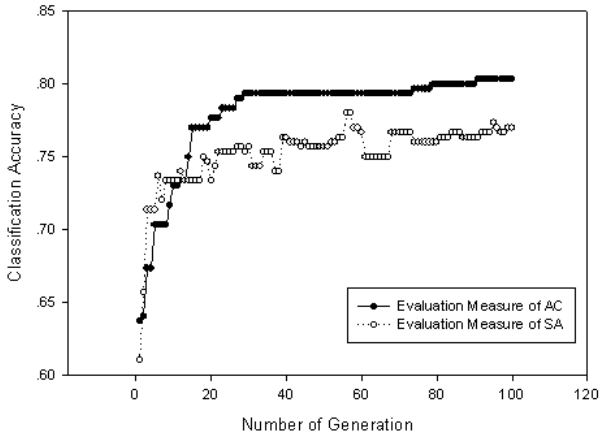


Figure 1: AC comparison between two measures

ally pre-labeled into two classes: interaction and non-interaction. Further, a subset of 300 testing sentences was used by GA to optimize parameters of the scoring scheme, while the remaining 100 sentences were prepared to test the achieved scheme on sentence classification. The distribution of class labels of the sample sentences is shown in Table 5.

Class label	300 sentences		100 sentences	
	No.	%	No.	%
Interaction	158	52.67	53	53
Non-interaction	142	47.33	47	47

Table 5: Class distribution of sample sentences

7.4.1 Comparison between two measures

We applied the evaluation measures, SA and AC, respectively to the subset of 300 testing sentences as the fitness function for GA, and recorded the scoring scheme of every generation resulted from GA. Figure 1 presents the distribution of achieved classification accuracy in terms of each scoring scheme optimized by GA. This comparison is done with respect to the generation and evaluated on 300 testing sentences using the annotations from the Genia corpus.

The achieved classification accuracy for AC generally outperforms the classification accuracy derived by SA. It reaches its highest classification accuracy 80.33% from the 91th generation. Therefore, AC is considered more efficient with the system and becomes our final choice of fitness function for GA.

7.4.2 Results of sentence identification

GA results in an optimized performance on the 300 sentences. It also results in an optimized scoring scheme along with its associated scoring threshold T, which are then applied together to the other 100 testing sentences. Table 6 and 7 present the system performance on the two sets respectively to both annotations.

Experimental Results	Genia		Our system	
	Interaction	Non	Interaction	Non
Precision	0.757	0.887	0.704	0.702
Recall	0.928	0.665	0.761	0.640
F-score	0.834	0.750	0.731	0.670
Overall $AC(\%)$	80.33		70.33	

Table 6: Performance on 300 testing sentences

Experimental Results	Genia		Our system	
	Interaction	Non	Interaction	Non
Precision	0.739	0.762	0.676	0.697
Recall	0.792	0.723	0.755	0.638
F-score	0.765	0.742	0.713	0.666
Overall $AC(\%)$	75.96		70.00	

Table 7: Performance on 100 testing sentences

Table 6 shows that when using the Genia annotations the system achieves an 0.834 F-score in identifying interaction sentences and an overall AC of 80.33%, which is much higher than the proportion of either interaction or non-interaction sentences in the 300 sentence subset. This indicates that the system performs well on both classes. In 100 generations GA is not able to evolve a scoring scheme that leads to an AC above 80.33%. Moreover, our system annotations achieve a lower performance than Genia annotations. We attribute the difference to the accuracy loss of our system annotations in the preprocessing steps as inaccurate annotations will lead to inappropriate patterns, thus harming the performance of sentence identification. For Genia annotations, the performance on the 100 testing sentences suggests an overfitting problem.

There are a number of preprocessing steps that affect the final classification performance. However, even assuming an ideal preprocessing of the unstructured text, our method relies on the assumption that all interaction sentences are articulated by a set of POS patterns that are distinct to all other types of sentences. The manual annotation of the training/testing set was a difficult task, so it is reasonable to assume that this will also be difficult for the classifier. The use of passive voice and the common use of comma splicing within patterns makes sentence-level classification an especially difficult task. Another source of interactions that our system cannot identify are implied and assume a deeper semantic understanding of the concepts them-

selves. Other sentences are long enough that the interaction itself is merely a secondary purpose to another idea. All of these factors pose interesting challenges for future development of this work.

Moreover, we also experimented with 10 empirical scoring schemes derived from previous experiments on the 300 sentences respectively, including the scheme in the Table 2. Several fixed thresholds were attempted for obtained alignment scores to differentiate between interaction and non-interaction sentences. Without using GA to optimize parameters of the scoring scheme, the best performance of 10 empirical schemes is an overall AC of 65.67%, which is outperformed at the 3rd generation of the GA optimization with Genia annotations.

7.5 System performance comparison

Within the framework of our system, we further conducted experiments on the same dataset for sentence identification using interaction patterns generated by another pattern generating algorithm (PGA) (Huang et al., 2004) in order to compare with the performance of patterns obtained by our pattern learning method.

In our implementation, PGA iterates over all pairs of candidate sentences in the training set and calculates the best alignment for each pair in terms of the cost scheme of gap penalties proposed (Huang et al., 2004). Each consensus sequence from the optimal alignment of each pair forms a pattern. The filter rules proposed are also applied. PGA has a time complexity of $O(n^2)$ in the size of candidate sentences, n. Hence, our proposed pattern learning method is much more efficient when dealing with large collections of biological texts. PGA produces a large number of patterns, even with $f_{min} = 5$ and other filtering criteria. There are 37,319 common patterns between two types of annotations.

Attributes	Genia	Our system
Potential patterns ($f_{min} = 5$)	476,600	387,302
Extracted patterns ($f_{min} = 5$)	176,082	88,800

Table 8: Pattern extraction results of PGA

In order to make a direct comparison, we decided to experiment with the same number of interaction patterns. For Genia annotations, we chose the most frequent 209 patterns generated by PGA to compare with the 209 patterns by our method. For our system annotations, two sets of 302 patterns are employed. Further, it is found that there are 96 common patterns between the two sets of 209 patterns for Genia annotations, and 153 common patterns between the two sets of 302 patterns for our system annotations. Table 9 and 10 present the results of sentence identification of PGA. The results show that patterns generated by PGA do not perform as well as patterns obtained by our method.

Experimental	Genia		Our system	
Results	Interaction	Non	Interaction	Non
Precision	0.721	0.869	0.663	0.699
Recall	0.918	0.606	0.785	0.556
F-score	0.808	0.714	0.719	0.619
Overall $AC(\%)$	77.00		67.67	

Table 9: Performance of PGA on 300 testing sentences

Experimental	Genia		Our system	
Results	Interaction	Non	Interaction	Non
Precision	0.664	0.796	0.698	0.635
Recall	0.849	0.574	0.566	0.766
F-score	0.745	0.667	0.625	0.694
Overall $AC(\%)$	71.98		66.00	

Table 10: Performance of PGA on 100 testing sentences

8 Conclusion and future work

In this paper, a novel approach is presented to automatically extract the representative patterns of biological interactions, which are used to detect sentences that describe biological interactions. We conducted the experiments on our designed system based on the Genia corpus. By means of a genetic algorithm, the system achieves an 0.834 F-score using Genia annotations and an 0.731 F-score using our system annotations in identifying interaction sentences by evaluating 300 sentences. By applying the optimized scoring scheme to another set of 100 sentences, the system achieves comparable results for both types of annotations. Furthermore, by comparing with another pattern generating algorithm, we infer that our proposed method is more efficient in producing patterns to identify interaction sentences.

In our future work, we would like to employ the obtained interaction patterns to guide the extraction of specific interactions. The matching between patterns and sentences will be performed and the matched parts of each sentence will be extracted as candidate interactions. Further reasoning processes can be performed by means of available biological ontologies, such as UMLS Semantic Network (Mccray and Bodenreider, 2002) and Gene Ontology (Consortium, 2001), to infer new relations from the initial interactions. Such processes can be employed to derive additional biological knowledge from existing knowledge, or test for biological consistency of the newly entered data.

References

Christian Blaschke, Miguel A. Andrade, Christos Ouzounis, and Alfonso Valencia. 1999. Automatic extraction of biological information from scientific text: Protein-protein interactions. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 60–67. AAAI Press.

Razvan Bunescu, Ruifang Ge, Rohit J Kate, Edward M Marcotte, Raymond J Mooney, Arun K Ramani, and Yuk W Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155.

Keh-Jiann Chen, Wen Tsuei, and Lee-Feng Chien. 1998. Pat-trees with the deletion function as the learning device for linguistic patterns. In *Proceedings of the 17th international conference on Computational linguistics*, pages 244–250, Morristown, NJ, USA. Association for Computational Linguistics.

Lee-Feng Chien. 1997. Pat-tree-based keyword extraction for chinese information retrieval. *SIGIR Forum*, 31(SI):50–58.

Gene Ontology Consortium. 2001. Creating the gene ontology resource: design and implementation. *Genome Research*, 11(8):1425–1433.

Anna Divoli and Teresa K. Attwood. 2005. Bioie: extracting informative sentences from the biomedical literature. *Bioinformatics*, 21(9):2138–2139.

John E. Freund and Benjamin M. Perles. 2006. *Modern Elementary Statistics*. Prentice Hall.

Jorg Hakenberg, Conrad Plake, Ulf Leser, Harald Kirsch, and Dietrich Rebholz-Schuhmann. 2005. Lll'05 challenge: Genic interaction extraction with alignments and finite state automata. In *Proceedings of Learning Language in Logic Workshop (LLL'05) at ICML*, page 38C45, Bonn, Germany.

Minlie Huang, Xiaoyan Zhu, Yu Hao, Donald G. Payan, Kunbin Qu, and Ming Li. 2004. Discovering patterns to extract protein-protein interactions from full texts. *Bioinformatics*, 20:3604–3612.

Hyunchul Jang, Jaesoo Lim, Joon-Ho Lim, Soo-Jun Park, Kyu-Chul Lee, and Seon-Hee Park. 2006. Finding the evidence for protein-protein interactions from pubmed abstracts. *Bioinformatics*, 22(14):e220–e226.

Sittichai Jiampojamarn, Nick Cercone, and Vlado Kešelj. 2005. Biological Named Entity Recognition using N-grams and Classification Methods. In *Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING'05*, Tokyo, Japan.

Jiao Li, Xian Zhang, Yu Hao, Minlie Huang, and Xiaoyan Zhu. 2005. Learning domain-specific knowledge from context–thuir at trec2005 genomics track. In *Proceedings of 14th Text Retrireval Conference (TREC2005)*, Gaithersburg, USA.

Alexa T. Mccray and Olivier Bodenreider. 2002. A conceptual framework for the biomedical domain. In *Semantics of Relationships, Kluwer*, pages 181–198. Kluwer Academic Publishers.

Andrei Mikheev. 2002. Periods, capitalized words, etc. *Comput. Linguist.*, 28(3):289–318.

Donald R. Morrison. 1968. Patricia — Practical Algorithm To Retrieve Information Coded in Alphanumeric. *Journal of the ACM*, 15(4):514–534.

Toshihide Ono, Haretsugu Hishigaki, Akira Tanigami, and Toshihisa Takagi. 2001. Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161.

Conrad Plake, Jorg Hakenberg, and Ulf Leser. 2005. Learning patterns for information extraction from free text. In *Proceedings of AKKD 2005*, Karlsruhe, Germany.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94, Somerset, New Jersey.

Soumya Raychaudhuri. 2006. *Computational Text Analysis: For Functional Genomics and Bioinformatics*. Oxford University Press.

Andre Skusa, Alexander Ruegg, and Jacob Kohler. 2005. Extraction of biological interaction networks from scientific literature. *Brief Bioinform*, 6(3):263–276.

Using Hedges to Enhance a Disease Outbreak Report Text Mining System

Mike Conway, Nigel Collier
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo 101-8430, Japan
{mike|collier}@nii.ac.jp

Son Doan
Vanderbilt University Medical Center
2525 West End Ave., Suite 800
Nashville, TN 37235, USA
son.doan@vanderbilt.edu

1 Introduction

Identifying serious infectious disease outbreaks in their early stages is an important task, both for national governments and international organizations like the World Health Organization. Text mining and information extraction systems can provide an important, low cost and timely early warning system in these circumstances by identifying the first signs of an outbreak automatically from online textual news. One interesting characteristic of disease outbreak reports — which to the best of our knowledge has not been studied before — is their use of speculative language (*hedging*) to describe uncertain situations. This paper describes two uses of hedging to enhance the BioCaster disease outbreak report text mining system.

Following a brief description of the BioCaster system and corpus (section 2), we discuss in section 3 previous uses of hedging in NLP and the methods used to identify hedges in the current work. In section 4 we describe some initial classification experiments using hedge features. Section 5 describes a "speculative" method of tagging disease outbreak reports with a metric designed to aid users of the BioCaster system in identifying articles of interest.

2 BioCaster System & Corpus

The BioCaster system scans online news reports for stories concerning infectious disease outbreaks (e.g. H5N1, Ebola) and makes its results available to registered users as email alerts (Collier et al., 2008). In addition to this email service, data that has been filtered through a topic classifier but which is still uninterpreted is used to populate a Google Map application called the *Global Health Monitor*.[1]

The BioCaster corpus consists of 1000 news articles downloaded from the WWW and then manually categorized and annotated with Named Entities by two PhD students. Articles were collected from various news sources (e.g. *BBC*, *New York Times* and ProMED-Mail[2]). Each document is classified as either *relevant* (350) or *reject* (650).[3]

The corpus is designed to include difficult borderline cases where more advanced understanding of the context is required. For example, an article may be about, say, polio, but not centrally concerned with specific outbreaks of that disease. Instead, the article could report a vaccination campaign or research breakthrough.

3 Hedges

According to Hyland (1998), in an extensive study of speculative language in science writing, hedges "are the means by which writers can present a proposition as an opinion rather than a fact." More recently, Kilicoglu and Bergler (2008) have presented a method for automatically identifying hedges in the biomedical domain. In the current work, we used a science orientated hedge lexicon derived from Mercer et al. (2004). The lexicon consisted of 72 verbs (including *appear, appears, appeared, appearing, indicate, indicates, indicated, indicating*, and so on) and 32 non-verbs (including, *about, quite, poten-*

[1] www.biocaster.org

[2] ProMED-Mail is a human curated service for monitoring disease outbreak reports (www.promedmail.org.)

[3] For copyright reasons, the BioCaster corpus is not publicly available.

Proceedings of the Workshop on BioNLP, pages 142–143,
Boulder, Colorado, June 2009.

Rank	Hedge	Rank	Hedge
1	reported	9	suggests
2	suspected	10	estimated
3	probable	11	appeared
4	suspect	12	appearing
5	usually	13	mostly
6	see	14	assumes
7	reports	15	predicted
8	sought	16	suggested

Table 1: Statistically Significant Hedges

Features	Naive Bayes		SVM	
	Acc	F	Acc	F
9000 χ^2	94.8	0.93	92.2	0.89
Unigram	88.4	0.85	90.9	0.87
Unigram+hedge	88.0	0.85	91.7	0.89

Table 2: Classification Results

tially, likely and so on). Preliminary work showed that the frequency of hedge words differs in the two categories of the `BioCaster` corpus (*relevant* and *reject*) at a highly significant level using the χ^2 test ($P < 0.01$). Table 1 shows the 16 most discriminating hedge words in the `BioCaster` corpus (identified using the χ^2 feature selection method.)

4 Classification Experiment

The current `BioCaster` system uses n-gram based text classification to identify disease outbreak reports, and reject other online news. We used hedging features to augment this classifier, and evaluated the results using a subset of the `BioCaster` corpus. One binary hedging feature was used. The feature was "true" if and only if one of the 105 hedge lexemes identified by Mercer et al. (2004) occurred in the input document within 5 words of a disease named entity. Results are shown in Table 2, where it can be seen that the addition of a single binary hedge feature to the unigram feature set increases accuracy by 0.8%. The performance does not however reach the level achieved by the χ^2 9000 n-gram feature set described in Conway et al. (2008).

5 Towards a "Speculative" Metric

Users of the `BioCaster` system would benefit from an indicator of how "speculative" each news article is, as breaking news regarding disease outbreaks is characterized by uncertainty, which is encoded using hedging. We use the Mercer list of 105 hedging words as described above, in conjunction with statistics derived from a 10,000 document section of the Reuters corpus to provide a "speculative" metric.[4] We calculated total frequencies for all 105 hedge words in each of the 10,000 Reuters documents — that is, the *total* number of hedge words per document — then ranked these frequencies (after normalizing the frequencies to take account of document length). The bottom third of documents had hedge percentages in the range 0% - 0.2544% (LOW). The middle third had hedge percentages in the range 0.2545% - 1.0574 (MEDIUM). The range for the top third was 1.0575% - 100% (HIGH). Documents inputted to the `BioCaster` system automatically have their proportion of hedge words calculated and are assigned a value according to their position on the scale (LOW, MEDIUM or HIGH). Table 3 shows that a majority of the documents in the *accept* segment of the `BioCaster` corpus can be tagged as highly speculative using this method.

	Accept (%)	Reject (%)
High	64.2	48.3
Medium	29.5	36.7
Low	6.3	15.0

Table 3: Proportion of Articles in Each Category

References

N. Collier, S. Doan, A. Kawazoe, R. Matsuda-Goodwin, M. Conway, Y. Tateno, Q-H. Ngo, D. Dien, A. Kawtrakul, K. Takeuchi, M. Shigematsu, and K. Taniguichi. 2008. BioCaster: Detecting Public Health Rumors with a Web-based Text Mining System. *Bioinformatics*, 24(24):2940–2941.

M. Conway, S. Doan, A. Kawazoe, and N. Collier. 2008. Classifying Disease Outbreak Reports Using N-grams and Semantic Features. *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008), Turku, Finland*, pages 29–36.

K. Hyland. 1998. *Hedging in Scientific Research Articles*. John Benjamins, Amsterdam.

H. Kilicoglu and S. Bergler. 2008. Recognizing Speculative Language in Biomedical Research Articles: a Linguistically Motivated Perspective. *BMC Bioinformatics*, 9(Suppl 11):S10.

R. Mercer, C. DiMarco, and F. Kroon. 2004. The Frequency of Hedging Cues in Citation Contexts in Scientific Writing. In *Proceedings of the Canadian Conference on AI*, pages 75–88.

[4] Reuters Corpus, Volume 1, English language, 1996-08-20 to 1997-08-19 (Release date 2000-11-03, Format version 1, correction level 0).

Exploring Two Biomedical Text Genres for Disease Recognition

Aurélie Névéol, Won Kim, W. John Wilbur, Zhiyong Lu*

National Center for Biotechnology Information
U.S. National Library of Medicine
Bethesda, MD 20894, USA
{neveola,wonkim,wilbur,luzh}@ncbi.nlm.nih.gov

Abstract

In the framework of contextual information retrieval in the biomedical domain, this paper reports on the automatic detection of disease concepts in two genres of biomedical text: sentences from the literature and PubMed user queries. A statistical model and a Natural Language Processing algorithm for disease recognition were applied on both corpora. While both methods show good performance (F=77% *vs.* F=76%) on the sentence corpus, results on the query corpus indicate that the statistical model is more robust (F=74% *vs.* F=70%).

1 Introduction

Contextual Information Retrieval (IR) is making use of additional information or assumptions about the users' needs beyond the obvious intent of the query. IR systems need to go beyond the task of providing generally relevant information by assisting users in finding information that is relevant to them and their specific needs at the time of the search. A practical example of a Google contextual IR feature is when the search engine returns a map showing restaurant locations to a user entering a query such as "Paris restaurants."

The *contextual* aspects of a user's search were defined for example by Saracevic (1997) who discussed integrating the cognitive, affective, and situational levels of human computer interaction in IR systems. Other research efforts studied users' search behavior based on their level of domain knowledge (Zhang et al., 2005) or aimed at modeling users' interests and search habits (Rose and Levinson, 2004; Teevan et al., 2005).

Information about the search context may be sought explicitly from the user through profiling or relevance feedback (Shen et al., 2005). Recent work also exploited query log analysis and basic computer environment information (Wen et al. 2004), which involve no explicit interaction with the user. In adaptive information retrieval, context information is inferred based on query analysis and collection characteristics (Bai and Nie 2008).

In the biomedical domain, a need for contextual information retrieval was identified in particular for clinical queries submitted to PubMed (Pratt and Wasserman, 2000). Building on the idea that a specific type of document is required for searches with a "clinical" context, the PubMed Clinical Queries portal was developed (Haynes and Wilczynski, 2004). A perhaps more prominent contextual feature of PubMed is the "citation sensor", which identifies queries classified by Rose and Levinson as reflecting a "Navigational" or "Obtain resource" goal. For example, the citation sensor will identify and retrieve a specific citation if the user enters the article title as the query. The analysis of Entrez logs shows that MEDLINE is the most popular database among the 30 or so databases maintained by the National Center for Biotechnology Information (NCBI) as it receives most of Entrez traffic. This suggests that there is a need to complement the information retrieved from MEDLINE by giving contextual access to other NCBI resources re-

Proceedings of the Workshop on BioNLP, pages 144–152,
Boulder, Colorado, June 2009.

levant to users' queries, such as Entrez Gene, Clinical Q&A or BookShelf. In addition, the NLM estimated that about 1/3 of PubMed users are not biomedical professionals. In this light, providing an access point to consumer information such as the Genetics Home Reference might also be useful. To achieve this, the *sensor* project was recently launched with the goal of recognizing a variety of biomedical concepts (e.g. gene, protein and drug names) in PubMed queries. These high-level concepts will help characterize users' search context in order to provide them with information related to their need beyond PubMed. For instance, if a user query contains the drug name "Lipitor", it will be recognized by the drug sensor and additional information on this drug from Clinical Q&A will be shown in the side bar in addition to default PubMed results. Since disease names are common in PubMed queries, the goal of this work is to investigate and benchmark computational techniques for automatic disease name recognition as an aid to implementing PubMed search contexts.

2 Related Work

Despite a significant body of literature in biomedical named entity recognition, most work has been focused on gene, protein, drug and chemical names through challenges such as BioCreAtIvE[1] or the TREC Genomics/Chemical tracks (Park and Kim, 2006). Other work addressed the identification of "medical problems" in clinical text (Aronson et al. 2007; Meystre and Haug, 2005). This task was the topic of a Medical NLP challenge[2], which released a corpus of anonymized radiography reports annotated with ICD9 codes. Although there is some interest in the biomedical community in the identification of disease names and more specifically the identification of relationships between diseases and genes or proteins (Rindflesh and Fizman, 2003), there are very few resources available to train or evaluate automatic disease recognition systems. To the best of our knowledge, the only publicly available corpus for disease identification in the literature was developed by Jimeno et al. (2008). The authors annotated 551 MEDLINE sentences with UMLS concepts and used this dataset to benchmark three different automatic methods for disease name recognition. A MEDLINE corpus annotated with "malignancy" mentions and part-of-speech tags is also available (Jin et al. 2006). This corpus is targeted to a very restricted type of diseases. The annotations are also domain specific, so that "cancer of the lung" is not considered a malignancy mention but a mention of malignancy and a mention of malignancy location.

As in previous studies, we aim to investigate the complexity of automatic disease recognition using state-of-the-art computational techniques. This work is novel in at least three aspects: first, in addition to using the MEDLINE sentence corpus (Jimeno et al 2008), we developed a new corpus comprising disease annotations on 500 randomly selected PubMed queries. This allowed us to investigate the influence of *local context*[3] through the comparison of system performance between two different genres of biomedical text. Second, by using a knowledge based tool previously benchmarked on the same MEDLINE corpus (Jimeno et al. 2008), we show that significant performance differences can be observed when parameters are adjusted. Finally, a state-of-the-art statistical approach was adapted for disease name recognition and evaluated on both corpora.

3 Two Biomedical Corpora with *disease* annotations

The first issue in the development of such a corpus is to define the very concept of disease. Among the numerous terminological resources available, such as Medical Subject Headings (MeSH®, 4,354 disease concepts) or the International Classification of Diseases (ICD9, ~18,000 disease concepts), the UMLS Metathesaurus® is the most comprehensive: the 2008AB release includes 252,284 concepts in the *disorder* Semantic Group defined by McCray et al. (2001). The UMLS Metathesaurus is part of the Semantic Network, which also includes a set of broad subject categories, or Semantic Types, that provide a consistent categorization of all concepts represented in the Metathesaurus. The Semantic Groups aim at providing an even broader categorization for UMLS concepts. For example, the disorder Semantic Group comprises 12 Semantic Types including *Disease or Syndrome*, *Cell or Molecular Dysfunction* and *Congenital Abnormalities*.

[1] http://biocreative.sourceforge.net/

[2] http://www.computationalmedicine.org/challenge/index.php

[3] Here, by *context*, we mean the information surrounding a disease mention available in the corpora. This is different from the "search context" previously discussed.

Furthermore, like the gene mention (Morgan et al. 2008) and gene normalization (Smith et al. 2008) tasks in BioCreative II, the task of disease name recognition can also be performed at two different levels:

1. *disease mention*: the detection of a snippet of text that refers to a disease concept (e.g. "alzheimer" in the sample query shown in Table 2)
2. *disease concept*: the recognition of a controlled vocabulary disease concept (e.g. "C0002395-alzheimer's disease" in our Table 2 example) in text.

In this work, we evaluate and report system performance at the concept level.

3.1 Biomedical literature corpus

Sentence	Kniest dysplasia is a moderately severe chondrodysplasia phenotype that results from mutations in the gene for type ii collagen col2a1.
Annotations	C0265279-Kniest dysplasia C0343284-Chondrodysplasia, unspecified

Table 1: Excerpt of literature corpus (PMID: 7874117)

The corpus made available by Jimeno et al. consists of 551 MEDLINE sentences annotated with UMLS concepts or concept clusters: concepts that were found to be linked to the same term. For example, the concepts "Pancreatic carcinoma" (C0235974) and "Malignant neoplasm of pancreas" (C0346647) share the same synonym "Pancreas Cancer", thus they were clustered. The sentences were selected from a set of articles curated for Online Mendelian Inheritance in Man (OMIM) and contain an average of 27(+/- 11) tokens, where tokens are defined as sequences of characters separated by white space. A set of UMLS concepts (or clusters) is associated with each sentence in the corpus. However, no boundary information linking a phrase in a sentence to an annotation was available. Table 1 shows a sample sentence and its annotations.

3.2 Biomedical query corpus

A total of 500 PubMed queries were randomly selected and divided into two batches of 300 and 200 queries, respectively. Queries were on average 3.45(+/- 2.64) tokens long in the 300 query batch and 3.58(+/- 4.63) for the 200 query batch, which is consistent with the average length of PubMed queries (3 tokens) reported by Herskovic et al. (2007).

The queries in the first set were annotated using Knowtator (Ogren, 2006) by three annotators with different backgrounds (one biologist, one information scientist, one computational linguist). Two annotators annotated the queries using UMLS concepts from the *disorder* group, while the other annotator simply annotated diseases without reference to UMLS concepts. Table 2 shows a sample query and its annotations. A consensus set was obtained after a meeting between the annotators where diverging annotations were discussed and annotators agreed on a final, unique, version of all annotations. The consensus set contains 89 disease concepts (76 unique).

Query	alzheimer csf amyloid
Annotations	Ann. 1: "alzheimer"; 0-8; Ann. 2, 3: "alzheimer"; 0-8; C0002395-alzheimer's disease

Table 2: Excerpt of annotated 300-query corpus. Boundary information is given as the character interval of the annotated string in the query (here, 0-8).

The queries in the second set were annotated with UMLS concepts from the *disorder* group by one of the annotators who also worked on the previous set. In this set, 53 disease concepts were annotated (51 unique).

4 Automatic *disease* recognition

With the perspective of a contextual IR application where the disease concepts found in queries will be used to refer users to disease-specific information in databases other than MEDLINE, we are concerned with high precision performance. For this reason, we decided to experiment with methods that showed the highest precision when compared to others. In addition, given the size of the corpora available and the type of the annota-

tions, machine learning methods such as CRFs or SVM did not seem applicable.

Table 3 shows a description of the training and test sets for each corpus.

Data	Lit. Corpus	Query Corpus
Training	276 sentences (487 disease concepts, 185 unique)	300 queries (89 disease concepts, 76 unique)
Testing	275 sentences (437 disease concepts, 185 unique)	200 queries (53 disease concepts, 51 unique)
All	551 sentences (924 disease concepts, 280 unique)	500 queries (142 disease concepts, 120 unique)

Table 3: Description of the training and test sets

4.1 Natural Language Processing

Disease recognition was performed using the Natural Language Processing algorithm implemented in MetaMap (Aronson, 2001)[4]. The tool was restricted to retrieve concepts from the *disorder* group, using the UMLS 2008AB release and "longest match" feature.

In practice, MetaMap parses the input text into noun phrases, generates variants of these phrases using knowledge sources such as the SPECIALIST lexicon, and maps the phrases to UMLS concepts.

4.2 Priority Model

The priority model was first introduced in (Tanabe and Wilbur, 2006) and is adapted here to detect *disease mentions* in free text. Because our evaluation is performed at the *concept* level, the mentions extracted by the model are then mapped to UMLS using MetaMap.

The priority model approach is based on two sets of phrases: one names of diseases, *D*, and one names of non-diseases, *N*. One trains the model to assign two numbers, *p* and *q*, to each token *t* that appears in a phrase in either *D* or *N*. Roughly, *p* is the probability that a phrase from *D* or *N* that has the token *t* in it is actually from *D* and *q* is the relative weight that should be assigned to *t* for this purpose and represents a quality estimate. Given a phrase

$$ph = t_1 t_2 \ldots t_k \quad (1)$$

and for each t_i the corresponding numbers p_i and q_i we estimate the probability that $ph \in D$ by

$$prob = p_1 \prod_{j=2}^{k} 1-q_j \; + \sum_{i=2}^{k} q_i p_i \prod_{j=i+1}^{k} 1-q_j \quad (2)$$

The training procedure for the model actually chooses the values of all the p and q quantities to optimize the $prob$ values over all of *D* and *N*. For this work we have extended the approach to include a quantity

$$qual = \left[q_1 p_1 \prod_{j=2}^{k} 1-q_j \; + \sum_{i=2}^{k} q_i^2 p_i \prod_{j=i+1}^{k} 1-q_j \right] \Big/ prob \quad (3)$$

which represents a weighted average of all the quality numbers q_i. We apply this formula to obtain $qual$ as long as $prob \geq 0.5$. If $prob < 0.5$ we replace all numbers p_i by $1-p_i$ in (2) and (3) to obtain $qual$.

For this application we obtained the sets *D* and *N* from the SEMCAT data (Tanabe, Thom et al. 2006) supplemented with the latest UMLS data. We removed any term from *D* and *N* that contained less than five characters in order to decrease the occurrence of ambiguous terms. Also the 1,000 most frequent terms from *D* were examined manually and the ambiguous ones were removed. The end result is a set of 332,984 phrases in *D* and 4,253,758 phrases in *N*. We trained the priority model on *D* and *N* and applied the resulting training to compute for each phrase in *D* and *N* a vector of values $prob, qual$. In this way *D* and *N* are converted to V_D and V_N. We then constructed a Mahalanobis classifier (Duda, Hart and Stork, 2001) for two dimensional vectors as the difference in the Mahalanobis distance of any such vector to Gaussian approximations to V_D and V_N. We refer to the number produced by this classifier as the Mahalanobis score. By randomly dividing both *D* and *N* into three equal size pieces and training on two from each and testing on the third, in a three-fold cross validation we found the Mahalanobis classifier to perform at 98.4% average precision and 93.9% precision-recall breakeven point. In a final step we applied a simple regression method to estimate the probability that a given Maha-

[4] Additional information is also available at http://metamap.nlm.nih.gov/

lanobis score was produced by a phrase belonging to *D* and not *N*. Given a phrase *phr* we will denote this final probability produced as PMA(*phr*).

The second important ingredient of our statistical process is how we produce phrases from a piece of text. Given a string of text *TX* we apply tokenization to *TX* to produce an ordered set of tokens $t_1, t_2, \ldots, t_n$. Among the tokens produced will be punctuation marks and stop words and we denote the set of all such tokens by Z . We call a token segment $t_j, \ldots, t_k$ maximal if it contains no element of Z and if either $j=1$ or $t_{j-1} \in Z$ and likewise if $k=n$ or $t_{k+1} \in Z$. Given text *TX* we will denote the set of all maximal token segments produced in this way by $S_{\max}(TX)$. Now given a maximal token segment *mts*= $t_j, \ldots, t_k$ we define two different methods of finding phrases in *mts*. The first assumes we are given an arbitrary set of phrases *PH*. We recursively define a set of phrases $I\ mts, PH$ beginning with this set empty and with the parameter $u = j$. Each iteration consists of asking for the largest $v \le k$ for which $t_u, \ldots, t_v \in PH$. If there is such a v we add $t_u, \ldots, t_v$ to $I\ mts, PH$ and set $u = v+1$. Otherwise we set $u = u+1$. We repeat this process as long as $u \le k$. The second approach assumes we are given an arbitrary set of two token phrases *P2*. Again we recursively define a set of phrases $J\ mts, P2$ beginning with this set empty and with the parameter $u = j$. Each iteration consists of asking for the largest $v \le k$ for which given any $i,\ u \le i < v,\ \ t_i, t_{i+1} \in P2$. If there is such a v we add $t_u, \ldots, t_v$ to $J\ mts, P2$ and set $u = v+1$. Otherwise we set $u = u+1$. We repeat this process as long as $u \le k$.

In order to apply our phrase extraction procedures we need good sets of phrases. In addition to *D* and *N* already defined above, we use another set of phrases defined as follows. Let *R* denote the set of all token strings with two or more tokens which do not contain tokens from Z and for which there are at least three MEDLINE records (title and abstract text only) in which the token string is repeated at least twice.

We then define $R' = R - D \cup N$. We make use of R' in addition to *D* and *N*. For the set $P2$ we take the set of all two token phrases in MEDLINE documents for which the two tokens co-occur as this phrase much more than expected, i.e., with a $\chi^2 \ge 10,000$ (based on the two-by-two contingency table).

#Initialization: Given a text *TX*, set $S \leftarrow S_{\max}\ TX$ and $X \leftarrow \varnothing$.
#Processing: While($S \neq \varnothing$){
I. select $mts \in S$
II. If($I\ mts, D \neq \varnothing$) $K \leftarrow I\ mts, D$
else if($I\ mts, R' \neq \varnothing$) $K \leftarrow I\ mts, R'$
else if($I\ mts, N \neq \varnothing$) $K \leftarrow \varnothing$
else
if($J\ mts, P2 \neq \varnothing$) $K \leftarrow J\ mts, P2$
else $K \leftarrow \varnothing$
III. $X \leftarrow X \cup K$
IV. $S \leftarrow S - mts$
}
#Return: All pairs $phr, PMA\ phr\ ,\ phr \in X$

Figure 1: Phrase finding algorithm

With these preliminaries, our phrase finding algorithm in pseudo-code is shown in Figure 1.

The output of this algorithm may then be filtered by setting a threshold on the *PMA* values to accept.

5 Results

5.1 Assessing the difficulty of the task

To assess the difficulty of disease recognition, we computed the inter-annotator agreement (IAA) on the 300-query corpus. Agreement was computed at the *disease mention* level for all three annotators and at the *disease concept* level for the two annotators who produced UMLS annotations.

Inter-annotator agreement measures for NLP applications have been recently discussed by Artstein and Poesio (2008) who advocate for the use of chance corrected measures. However, in our case, agreement was partly computed on a very large set of categories (UMLS concepts) so we decided to use Knowtator's built-in feature, which computes IAA as the percentage of agreement and

allows partial string matches. For example, in the query "dog model transient ischemic attacks", annotator 1 selected "ischemic attacks" as a *disorder* while annotator 2 and 3 selected "transient ischemic attacks" as UMLS concept *C0007787: Attacks, Transient Ischemic*. In this case, at the subclass level ("disorder") we have a match for this annotation. But at the exact span or exact category level, there is no match. Table 4 shows details of IAA at the disease mention level when partial matches are taken into account. For exact span matches, the IAA is lower, at 64.87% on average.

Disorder IAA	**Ann. 1**	**Ann. 2**	**Ann. 3**
Ann. 1	100%	71.77%	75.86%
Ann. 2		100%	71.68%
Ann. 3			100%

Table 4: Agreement on disease mention annotations (partial match allowed) – **average is 73.10%**

At the concept level, the agreement (when partial matches were allowed) varied significantly depending on the semantic types. It ranged between 33% for *Findings* and 83% for *Mental or Behavioral Dysfunction*. However, agreement on the most frequent category, *Disease or Syndrome*, was 72%, which is close to the annotators' overall agreement at the mention level. One major cause of disagreement was ambiguity caused by concepts that were clustered by Jimeno et al. For example, in query "osteoporosis and "fracture pattern", annotator 2 marked "osteoporosis" with both "C0029456-osteoporosis"(a *Disease or Syndrome* concept) and "C1962963-osteoporosis adverse event"(a *Finding* concept) while annotator 3 only used "C0029456-osteoporosis".

5.2 Results on Literature corpus

As shown in Table 3, the corpus was randomly split into a training set (276 sentences) and a test set (275 sentences). The training set was used to determine the optimal probability threshold for the Priority Model and parameter selection for MetaMap, respectively.

Priority Model parameter adjustments: the first result observed from applying the Priority Model was that *D* yielded about 90% of the output of the algorithm. Also results coming from R' and $P2$ were not well mapped to UMLS concepts by MetaMap. As a result, in this work we ignored disease candidates retrieved based on R' and $P2$. The best F-measure was obtained for a threshold of 0.3, which was consequently used on the test set.

Since the Priority Model algorithm does not perform any mapping to a controlled vocabulary source, the mapping was performed by applying MetaMap to the snippets of text returned with a probability value above the threshold.

Threshold	**P**	**R**	**F**
0	64	73	67
.1	67	73	70
.2	67	73	70
.3	**68**	**73**	**71**
.4	68	73	70
.5	68	72	69
.6	68	72	69
.7	68	72	69
.8	68	68	68
.9	65	60	62

Table 5: Precision (P), Recall (R) and F-measure of the Priority Model on the training set for different values of the probability threshold.

The results presented in Table 5 were obtained before any MetaMap adjustments were made.

MetaMap parameter adjustments: an error analysis was performed to adjust MetaMap settings. Errors fell into the following categories:

- A more specific disease should have been recognized (e.g. "deficiency" *vs.* "C2 deficiency")
- The definition of a cluster was lacking (e.g. "G6PD deficiency" comprised C0237987- Glucose-6-phosphate dehydrogenase deficiency anemia and C0017758- Glucosphosphate Dehydrogenase Deficiency but not C0017920- Deficiency of glucose-6-phosphatase)
- MetaMap mapping was erroneous (e.g. "hereditary breast" was mapped to C0729233-Dissecting aneurysm of the thoracic aorta instead of C0346153-Hereditary Breast Cancer)

The results of inter-annotator agreement and further study of MetaMap mappings indicated that concepts with the semantic type *Findings* seemed

to be frequently retrieved erroneously. For this reason, we also experimented not taking *Findings* into account as an additional adjustment for MetaMap. Table 6 shows the results of applying the MetaMap adjustments yielded from the error analysis on the training corpus.

Threshold	Findings	P	R	F
.3	Yes	80	78	79
.3	**No**	**85**	**78**	**81**

Table 6: performance of the Priority Model on the training set for threshold .3 depending on whether mappings to *Findings* are used in the "adjustments"

MetaMap disorder detection was also performed directly on the training corpus. An error analysis similar to what was presented above was carried out to determine the best parameters. Table 7 below shows the results obtained when all concepts from the 12 Semantic Types (STs) in the *disorder* group are taken into account with no adjustments ("raw"). Then, results including the adjustments from the error analysis are shown when all 12 STs are taken into account, when *Findings* are excluded (11STs) and when only the most frequent 6STs in the training set are taken into account.

Processing	P	R	F
Raw (12 STs)	50	77	61
Adjusted (12 STs)	52	75	61
Adjusted (11 STs)	57	73	64
Adjusted (6 STs)	**77**	**72**	**74**

Table 7: Performance of MetaMap on the training set

Finally, Table 8 shows the performance of both methods on the test set, using the optimal settings determined on the training set:

Method	P	R	F
Priority Model	**80**	**74**	**77**
MetaMap	75	78	76

Table 8: Precision (P), Recall (R) and F-measure of the Priority Model and MetaMap on the test set

5.3 Results on Query Corpus

The 300-query corpus was used as a training set and the 200-query corpus was used as a test set. For consistency with work on the literature corpus, we assessed the disease recognition on a gold standard set including "clusters" of UMLS concepts were appropriate. As previously with the Literature corpus, we used the training set to determine the best settings for each method. The performance of the Priority Model at different values of the probability threshold, based on the use of *D* and *N* as the sets of sample phrases is similar to that obtained with the literature corpus; 0.3 stands out as one of the three values for which the best F-measure is obtained (tied with .5 and .8).

Because of the brevity of queries *vs.* sentences, the MetaMap error analysis was very succinct and resulted in:

- Removal of C0011860-Diabetes mellitus type 2 as mapping for "diabetes"
- Removal of all occurrences of C0600688-Toxicity and C0424653-Weight symptom (finding)
- Adjustment on the number of STs taken into account

The difference in performance obtained on the training set for the different MetaMap adjustments considered is shown in Table 9 when MetaMap was applied to Priority Model output and in Table 10 when it was applied directly on the queries.

Threshold	Findings	P	R	F
.3	Yes	60	72	65
.3	**No**	**73**	**70**	**71**

Table 9: performance of the Priority Model on the training set for threshold .3 depending on whether mappings to *Findings* are used in the "adjustments"

Processing	P	R	F
Raw (12 STs)	41	82	55
Adjusted (12 STs)	44	82	57
Adjusted (11 STs)	58	81	68
Adjusted (6 STs)	**64**	**75**	**69**

Table 10: performance of MetaMap on the training set

Finally, Table 11 shows the performance of both methods on the test set, using the optimal settings determined on the training set:

Method	P	R	F
Priority Model	**76**	**72**	**74**
MetaMap	66	74	70

Table 11: Precision (P), Recall (R) and F-measure of the Priority Model and MetaMap on the test set

6 Discussion

Comparing the Two Methods. The performance of both methods on the query corpus is comparable to inter-annotator agreement (F=70-74 *vs.* IAA=72 on *Disease and Syndromes*). On both corpora, the Priority Model achieves higher precision and F-measure, while MetaMap achieves better recall.

Comparing the results obtained with MetaMap with those reported by Jimeno et al., precision is lower, but recall is much higher. This is likely to be due to the different MetaMap settings, and the use of different UMLS versions - Jimeno et al. did not provide any of this information, but based on the publication date of their paper, it is likely that they used one of the 2006 UMLS releases. Meystre and Haug (2006) also found that significant performance differences could be obtained with MetaMap by adjusting the content of the knowledge sources used.

On both text genres, 0.3 was found to be the optimal probability threshold for the Priority Model. Based on the performance at different values of the threshold, it seems that the model is quite efficient at ruling out highly unlikely diseases. However, for values above .3 the performance does not vary greatly.

Comparing Text Genres. For both methods, disease recognition seems more efficient on sentences. This is to be expected: sentences provide more context (e.g. more tokens surrounding the disease mention are available) and allow for more efficient disambiguation, for example on acronyms. Although acronyms are frequent both in queries and sentences, more undefined acronyms are found in queries. However, the difference in performance between the two methods seems higher on the query corpus. This indicates that the Priority Model could be more robust to sparse context.

It should be noted that there were diseases in all sentences in the literature corpus *vs.* about 1/3 to 1/2 of the queries. In addition, the query corpus included many author names, which could create confusion with disease names (in particular for the Priority Model). This difficulty was not found in the sentence corpus. However, sentences sometimes contain negated mention of diseases, which never occurred in the query corpus where little to no syntax is used.

We also noticed that while *Findings* seemed to be generally problematic concepts in both corpora, other concepts such as *Injury and Poisoning* were much more prevalent in the query corpus. For this reason, for the general task of disease recognition, a drastic restriction to as little as 6 STs is probably not advisable.

Limitations of the study. One limitation of our study is the relatively small number of disease concepts in the query corpus. Although the query and sentence corpus contain about 500 queries/sentences each, there are significantly less disease concepts found in queries compared to sentences. As a result, there is also less repetition in the disease concept found. This is partly due to the brevity of queries compared to sentences but mainly to the fact that while all the sentences in the literature corpus had at least one disease concept, this was not the case for the query corpus. We are currently addressing this issue with the ongoing development of a large scale query corpus annotated for diseases and other relevant biomedical entities.

7 Conclusions

We found that of the two steps of disease recognition, disease mention gets the higher inter-annotator agreement (*vs.* concept mapping). We have applied a statistical and an NLP method for the automatic recognition of disease concepts in two genres of biomedical text. While both methods show good performance (F=77% vs. F=76%) on the sentence corpus, results indicate that the statistical model is more robust on the query corpus where very little disease context information is available (F=74% *vs.* F=70%). As a result, the priority model will be used for disease detection in PubMed queries in order to characterize users' search contexts for contextual IR.

Acknowledgments

This research was supported by the Intramural Research Program of the NIH, National Library of Medicine. The authors would like to thank S. Shooshan and T. Tao for their contribution to the annotation of the query corpus; colleagues in the NCBI engineering branch for their valuable feedback at every step of the project.

References

Alan R. Aronson, Olivier Bodenreider, Dina Demner-Fushman, Kin Wah Fung, Vivan E. Lee, James G. Mork et al. 2007. From Indexing the Biomedical Literature to Coding Clinical Text: Experience with MTI and Machine Learning Approaches. ACL Workshop BioNLP.

Alan Aronson. 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proceedings of AMIA Symp*:17-21.

Ron Artstein and Massimo Poesio. 2008. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics* 34(4): 555-596

Jing Bai, and Jian-Yun Nie. 2008. Adapting information retrieval to query contexts. *Information Processing & Management.* 44(6):1902-22

Robert O. Duda, Peter. E. Hart and David G. Stork. 2001. *Pattern Classification.* New York: John Wiley & Sons, Inc.

R. Brian Haynes and Nancy L. Wilczynski. 2004. Optimal search strategies for retrieving scientifically strong studies of diagnosis from Medline: analytical survey. *BMJ*. 328(7447):1040.

Jorge R. Herskovic, Len Y. Tanaka, William Hersh and Elmer V. Bernstam. 2007. A day in the life of PubMed: analysis of a typical day's query log. *Journal of the American Medical Informatics Association.* 14(2):212-20.

Antonio Jimeno, Ernesto Jimenez-Ruiz, Vivian Lee, Sylvain Gaudan, Rafael Berlanga and Dietrich Rebholz-Schuhmann. 2008. Assessment of disease named entity recognition on a corpus of annotated sentences. *BMC Bioinformatics*. 11;9 Suppl 3:S3.

Yang Jin, Ryan T McDonald, Kevin Lerman, Mark A Mandel, Steven Carroll, Mark Y Liberman et al. 2006. Automated recognition of malignancy mentions in biomedical literature. *BMC Bioinformatics.* 7:492.

Alexa T. McCray, Anita Burgun and Olivier Bodenreider. 2001. Aggregating UMLS semantic types for reducing conceptual complexity. *Proceedings of Medinfo* 10(Pt 1):216-20.

Stéphane Meystre and Peter J. Haug. 2006. Natural language processing to extract medical problems from electronic clinical documents: performance evaluation. *J Biomed Inform.* 39(6):589-99.

Alexander A. Morgan, Zhiyong Lu, Xinglong Wang, Aaron M. Cohen, Juliane Fluck, Patrick Ruch et al. 2008. Overview of BioCreative II gene normalization. *Genome Biol.* 9 Suppl 2:S3.

Phillip V. Ogren. 2006. Knowtator: A plug-in for creating training and evaluation data sets for Biomedical Natural Language systems. *9th Intl. Protégé Conference*

Jong C. Park and Jung-Jae Kim. 2006. Named Entity Recognition. In S. Ananiadou and J. McNaught (Eds), *Text Mining for Biology and Biomedicine* (pp. 121-42). Boston|London:Artech House Inc.

Wanda Pratt and Henry Wasserman. 2000. QueryCat: automatic categorization of MEDLINE queries. *Proceedings of AMIA Symp*:655-9.

Tom C. Rindflesh and Marcelo Fiszman. 2003. The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. *J Biomed Inform.* 36(6):462-77

Daniel E. Rose and Danny Levinson. 2004. Understanding user goals in web search. In *Proceedings of the 13th international Conference on World Wide Web*:13-9

Tefko Saracevic. 1997. The Stratified Model of Information Retrieval Interaction: Extension and Application. *Proceedings of the 60th meeting of the. American Society for Information Science*:313-27

Xuehua Shen, Bin Tan and ChengXiang Zhai. 2005 Context-sensitive information retrieval using implicit feedback, In *Proceedings of the 28th annual international conference ACM SIGIR conference on Research and development in information retrieval*: 43-50.

Larry Smith, Laurraine K. Tanabe, Rie J. Ando, Cheng-Ju Kuo, I-Fang Chung , Chun-Nan Hsu et al. 2008. Overview of BioCreative II gene mention recognition. Genome Biol. 9 Suppl 2:S2.

Laurraine K. Tanabe, Lynn. H. Thom, Wayne Matten, Donald C. Comeau and W. John Wilbur. 2006. SemCat: semantically categorized entities for genomics. *Proceedings of AMIA Symp*: 754-8.

Laurraine K. Tanabe and W. John Wilbur. 2006. A Priority Model for Named Entities. *Proceedings of HLT-NAACL BioNLP Workshop*:33-40

Jaime Teevan, Susan T. Dumais and Eric Horvitz. 2005. Personalizing search via automated analysis of interests and activities. In *Proceeding of ACM-SIGIR'05*:449–56.

Ji-Rong Wen, Ni Lao, Wei-Ying Ma. 2004. Probabilistic model for contextual retrieval. *Proceedings of ACM-SIGIR'04*:57–63

Xiangmin Zhang, Hermina G.B. Anghelescu and Xiaojun Yuan. 2005. Domain knowledge, search behavior, and search effectiveness of engineering and science students: An exploratory study, *Information Research* 10(2): 217.

Towards Retrieving Relevant Information for Answering Clinical Comparison Questions

Annette Leonhard
School of Informatics
University of Edinburgh
EH8 9AB, Edinburgh, Scotland
annette.leonhard@ed.ac.uk

Abstract

This paper introduces the task of automatically answering clinical comparison questions using MEDLINE® abstracts. In the beginning, clinical comparison questions and the main challenges in recognising and extracting their components are described. Then, different strategies for retrieving MEDLINE® abstracts are shown. Finally, the results of an initial experiment judging the relevance of MEDLINE® abstracts retrieved by searching for the components of twelve comparison questions will be shown and discussed.

1 Introduction

Clinicians wishing to practice evidence-based medicine need to keep up with a vast amount of ever changing research to be able to use the current best evidence in individual patient care (Sackett et al., 1996). This can be difficult for time-pressed clinicians, although methods such as systematic reviews, evidence summaries and clinical guidelines can help to translate research into practice.

In a survey commissioned by Doctors.net.uk, 97% of doctors and nurses said that they would find a Question Answering (QA) Service useful, where they can ask questions in their own words (Bryant and Ringrose 2005). Studies have also shown that clinicians often want answers to particular questions, rather than getting information on broad topics (Chambliss & Conley, 1996; Ely et al., 1999, 2005).

A type of question that clinicians commonly want answered are comparison questions. In a corpus of clinical questions collected from the National Library of Health (NLH) Question Answering Service (http://www.clinicalanswers.nhs.uk), approximately 16% of the 4580 questions concern comparisons of different drugs, different treatment methods or different interventions as in (1).

(1) Have any studies directly compared the effects of Pioglitazone and Rosiglitazone on the liver?

Despite the frequency of comparison questions in the clinical domain, there are no clinical QA methods specially designed to answer them. This paper introduces the task of answering clinical comparison questions, focusing initially on questions involving comparisons between drugs. Section 2 presents an overview of comparative structures and Section 3, relevant previous work on clinical question answering and the computational extraction of comparisons. Section 4 discusses strategies for retrieving MEDLINE® abstracts involving comparisons. Section 5 presents the results of an initial experiment judging the relevance of MEDLINE® abstracts, which are then discussed in Section 6.

2 Background

2.1 Indicators of Comparative Constructions

Proceedings of the Workshop on BioNLP, pages 153–161,
Boulder, Colorado, June 2009.

In order to identify questions about comparisons that should trigger special purpose search and extraction mechanisms, as well as identifying explicit comparisons made in text, one needs to recognize constructions commonly used to express comparisons in English (i.e. similarities and/or differences between two or more entities). In this paper, the term "entity" refers to drugs, treatment methods or interventions, and the initial focus of the work is on comparative questions in which two or more drugs or interventions are compared with respect to a particular criterion such as efficacy in treating a certain disease. This reflects their common occurrence in the NLH corpus.

Comparisons can appear in either a comparative form or a superlative form. The comparative form is used to compare two or more entities with respect to a certain attribute. The superlative form compares or contrasts one entity with a set of other entities and expresses the end of a spectrum. The following examples illustrate the difference:

Comparative form:
Is Ibuprofen **better** than Paracetamol for treating pain?
Superlative form:
Is Ibuprofen the **best** treatment for pain?

Friedman (1989) developed one of the first computational treatments of comparative structures. Comparisons are challenging because they correspond to a diverse range of syntactic forms such as coordinate or subordinate conjunctions, adverbial constructions or wh-relative-like clauses. Comparisons are cross-categorical and encompass adjectives, quantifiers, and adverbs. Adjectives and adverbs indicating comparisons occur in the following patterns:

Comparative adjectives and adverbs:

Regular adjectives and adverbs:
ADJ/ADV -er (e.g. safer) [[as/than][1] X] [for Y]
Irregular adjectives and adverbs:
e.g. worse/better [[as/than] X] [for Y]
Analytical adjectives and adverbs:
e.g. less/more *ADJ/ADV* [than X] [for Y]

[1] *As/ than* are optional. For example see "A or B: What is safer?"

Superlative adjectives and adverbs:

Regular adjectives and adverbs:
ADJ/ADV -est (eg. safest) X [for Y]
Irregular adjectives and adverbs:
e.g. worst/best X [for Y]
Analytical adjectives and adverbs:
e.g. least/most *ADJ/ADV* X [for Y]

Comparisons can also be expressed in other parts of speech. In the NLH corpus the following examples occur:

Verbs: compared to/with, differ from
Nouns: comparison, difference
Conjunctions: versus/vs, or and instead of

With respect to their semantics (and hence, with respect to other phrases or constructions they may appear with) comparatives can be *scalar* or *non-scalar* and express either *equality* or *inequality* between the compared entities. (Superlatives are absolute and the notion of scalability and equality does not apply to them).

Scalar adjectives and adverbs refer to attributes that can be measured in degrees, implying a scale along which entities can be arrayed. *Non-scalar* adjectives and adverbs refer to attributes that cannot be measured in degrees. *Equality* refers to constructs where two or more compared entities are equal in respect to a shared quality, whereas i*nequality* emphasises the difference between entities in respect to a certain quality.

Table 1 gives an example showing the four possibilities for drugs and interventions.

Scalability	Equality	Example
+	+	As efficient as x
-	+	Same intervention as x
+	-	Better treatment than x
-	-	Drug x differs from drug y

Table 1. Features of comparatives.

The difference between *scalar* and *non-scalar* comparisons plays an important role as far as automatic processing of comparative constructions with SemRep (Rindflesch and Fiszman, 2003; Rindflesch et al., 2005) is concerned. This will be discussed in Section 3.1.

Regular expressions based on the given patterns for adjectives and adverbs and on the other parts of speech shown above, as well as their respective part-of-speech tags, were used to extract a subset of comparison questions from a corpus collected from the National Library of Health Question Answering Service website at http://www.clinicalanswers.nhs.uk, as described in Section 2.3.

2.2 The NLH QA Service

The NLH Question Answering service (QAS) was a on-line service that clinicians in the UK could use to ask questions, that were then answered by a team of clinical librarians from Trip Database Ltd.[2], founded by Jon Brassey and Dr Chris Price. The questions and their answers were then retained at the website and indexed by major clinical topics (e.g. Cancer, Cardiovascular disease, Diabetes, etc.) so that clinicians could consult the QA archive to check whether information relevant to their own clinical question was already available.

While the NHS QAS service was discontinued in 2008, its archive of questions and answers was integrated into ATTRACT[3], the Welsh National Public Health Service run by Jon Brassey. The aim of both services has been to provide answers in a clinically relevant time frame using the best available evidence.

From the NLH QAS archive, a total of 4580 unique Q-A pairs of different degrees of complexity were collected for 34 medical fields representing questions asked and answered over a 36 month period. These were put into an XML format that separated the questions from the answers, while co-indexing them to indicate their association.

2.3 The Comparison Question Corpus

A sub-corpus specifically of comparison questions was created by POS-tagging the questions of the initial corpus with the Penn Treebank tagset, using the TnT tagger (Brants 1999). Regular expression were then used to search the tagged corpus for tagged lexical elements that indicated the constructions noted in Section 2.2.

[2]http://www.tripdatabase.com/index.html
[3]http://www.attract.wales.nhs.uk/

Some questions were initially retrieved more than once because these questions contained more than one tag which was a comparison indicator. These duplicates were removed. There may be other comparative questions that might have been missed because of POS tagging errors. A small number of false positives were removed during manual post-processing. False positives were due to the fact that not all words tagged as superlatives are proper comparisons, but idiomatic expressions, such as "**best practise",** or proportional quantifiers (Huddleston and Pullum, 2002) such as "**Most** NSAIDs". (Scheible (2008) distinguishes eight different classes in which the superlative construction is used in English but only five of the eight classes involve true comparisons.) The result is a subset of 742 comparison questions out of the the total corpus of 4580 Q-A pairs.

Table 2. shows the number of occurrences for each item.

POS tag/Lexical item	Occurrences
JJR	195
RBR	124
JJS	207
RBS	68
versus, instead of	18
compared to/with, differ from	45
comparison, difference	85
Total	**742**

Table 2. Number of comparison indicators

3 Related Work

As the focus of this paper is biomedical text, the discussion here is limited to the work done in this context. Section 3.1 will present work on finding assertions involving comparisons in MEDLINE® abstracts and Section 3.2 will show work on answering clinical questions about comparisons.

3.1 Interpretation of Comparative Structures

(Fiszman et al., 2007) describes work on automatically interpreting comparative constructions in MEDLINE® abstracts. They use an extension of an

existing semantic processor, SemRep (Rindflesch and Fiszman, 2003; Rindflesch et al., 2005), from the Unified Medical Language System resources to construct semantic predications for the extracted comparative expressions.

Fiszman et al. concentrate on extracting "structures in which two drugs are compared with respect to a shared attribute", such as a drug's efficacy in treating a certain condition, illustrated in the following in example:

(3) **Losartan** *was more effective than* **atenolol** in reducing cardiovascular morbidity and mortality in patients with hyptertension, diabetes, and LVH.
[Example (20) in (Fiszman et al. 2007)]

The drugs' relative merits in achieving their purpose is expressed by positions on a scale. Words like *than*, *as*, *with*, and *to* are cues for identifying compared terms, the comparison scale and the relative position of the compared entities on the scale.

Fiszman et al. focused on extracting the drug names, the scale and the position on the scale as illustrated in the SemRep representation from example (1):

(4) Losartan COMPARED_WITH Atenolol
Scale: Effectivness
Losartan HIGHER_THAN Atenolol
[Example (21) in (Fiszman et al. 2007)]

The overall F-score for the SemRep performance on the test set is 81% .

Fiszman et al. do not deal with questions, nor with identifying the basis of the comparison or the population in this paper, both of which are important for generating relevant answers for clinical questions. However, as Fiszman and Demner-Fushman have pointed out (personal communication), it is possible to identify the basis of the comparison and the population. Two drugs function as arguments to the TREATS predicate, which identifies the disease that is the basis for the comparison. SemRep can also identify the population using the predicate PROCESS_OF. For the question "Is treatment A better than treatment B for treating disease C in population D?", SemRep would produce the following representation for the basis of the comparison (C) and the population (D):

A TREATS **C**
B TREATS **C**
C PROCESS_OF **D**

There is an essential limitation to SemRep, however: Its comparative module only considers *scalar* comparative constructions, as presented in Section 2.1. *Non-scalar* comparisons, e.g. comparisons like "Is X the same intervention as Y?" or "How does drug X differ from drug Y?" cannot be extracted using SemRep. Also, the SemRep algorithm only recognises entities which occur on the left and the right side of the comparison cue and hence cannot recognize comparisons in which both compared entities are to the right side of the comparative cue as in "Which is better: X or Y?". This means that different methods are needed in order to process *non-scalar* comparisons and *scalar* comparisons that cannot be recognized because of their structure. In future work, rules will be defined for the different syntactic structures in which *non-scalar* comparisons and *scalar* comparison with both entities on the same side of a comparative cue can occur to serve as a basis for argument extraction during parsing.

There may also be problems with "Wh-" or "anything" questions (e.g. "What is better than X for treating Y?" or "Is there anything better than X for treating Y?"), because "Wh-words" or "anything" do not have a type that can be mapped. While Question Typing might solve such problems, the point is that questions involving comparisons raise somewhat different problems than assertions, which I will have to deal with in the work being carried out here.

3.2 Answering Clinical Questions

Demner-Fushman and Lin (2006) address superlative clinical questions of the type "What is the best treatment for X" by using a hybrid approach consisting of information retrieval and summarization.

Demner-Fushman and Lin's task breaks down into subtasks of identifying the drugs using UMLS concepts, clustering the abstracts for the drugs using UMLS semantic relationships and creating a short summary for each abstract by using the abstract title and outcome sentence. They focus primarily on synthesising correct answers from a set

of search results consisting of MEDLINE® citations.

The system (*Cluster* condition) performs well compared to the baseline, which consists of the main interventions from the first three MEDLINE® abstracts retrieved by the manual PubMed queries. In a manual evaluation, only 20% of the drugs for the baseline were evaluated as beneficial, compared to 39% for the *Cluster* condition. 60% of the *PubMed* answers were judged as "good" in comparison to 83% for the *Cluster* condition.

The system orders the clusters by size, equating the most popular drug with the best drug. While this assumption is not always correct, the authors have observed that drugs that are studied more are more likely to be beneficial. In addition, while this approach might work for questions of the form "What is the best drug for X?" it cannot be used to answer other superlative questions such as Examples (5) or (6), because looking for the most studied drugs will not provide an answer to the question which drug has the fewest side effects or is safest to use.

(5) Which drug for treating X has the fewest side effects?
(6) Which drug is safest to use for treating X?

Despite this shortcoming, however, Demner Fushman and Lin's work of implementing an end-to-end QA system for superlatives provides a model for all future work in this area.

4 Strategies for Retrieving MEDLINE® Abstracts

As with (Fiszman et al., 2007) and (Demner-Fushman and Lin 2006), the current work starts with information retrieval. In particular, exploratory manual searches were first carried out via the OVID® portal to see if MEDLINE® abstracts are a useful resource for answering comparison questions such as "Is drug A better than drug B for treating X?"

With the assistance of a medical librarian from the University of Edinburgh's Information Services, different strategies to achieve the best possible retrieval of relevant abstracts were tried out. Two separate cases were considered: comparisons involving very popular, well-studied drugs and ones involving other drugs. First, strategies for the former will be described and illustrated with the following example question:

(7) Is paracetamol better than ibuprofen for reducing fever?

Titles and abstracts were searched for each compared entity (paracetamol and ibuprofen) and the basis of the comparison (fever). Then, the results were combined to return only abstracts containing both entities and the basis of the comparison. We found that search precision could be increased by limiting the search to *comparative study*, using OVID's publication type limit. That is, all abstracts that mention all three terms (i.e. the entities and the basis of the comparison) in the title or abstract involve relevant comparisons. The most common sources that were excluded by constraining the search to comparative studies are reviews, evaluation studies and case reports. These may contain relevant information but the initial focus was on the study type that was most likely to increase precision. (As the experiment reported in Section 5 and discussed in Section 6 shows, the restriction to comparative studies is insufficient to guarantee relevance.)

Constraining the search to comparative studies has somewhat different effects, depending on whether the drugs mentioned in the search are well-studied or not.

For popular, well-studied drugs, looking for the drug names often leads to hundreds of returned abstracts, most of which are not relevant. By including the basis of the comparison and limiting the study type to comparative studies, the number of returned abstracts for a set of 30 questions drops on average to 15% of the size of the original set of returned abstracts. For Example (7) a search for the combination of both drug names retrieved 593 abstracts. Including the basis of the comparison decreased the number to 139 abstracts. After constraining the results to comparative studies, the number of retrieved abstracts dropped to 24, which is a reduction of 83%.

For less-studied drugs, the difference in numbers of abstracts retrieved by including the basis of the comparison and limiting the search to the *comparative study* publication type is smaller compared

1. Is there any evidence to suggest that torasemide is better than furosemide as a diuretic?
2. Is lansoprazole better than omeprazole in treating dyspepsia?
3. Are there any studies comparing topical diclofenac gel with ibuprofen gel?
4. Effectiveness of Decapeptyl in treatment of prostate cancer in comparison to Zoladex?
5. Which is more effective ibuprofen or diclofenac for arthritis pain for pain relief?
6. Is calcium citrate better absorbed and a more effective treatment for osteoporosis than calcium carbonate?
7. Have any studies directly compared the effects of Pioglitazone and Rosiglitazone on the liver?
8. Is Famvir (famciclovir) better than acyclovir for Herpes zoster?
9. Is it true that men on captopril have a better quality of life than men on enalapril?
10. What is the first choice for Type 2 diabetes patients: sulphonylurea or metformin?
11. Is there any evidence as to which is more effective at preventing malaria: Malarone or Doxycyline?
12. In conjunctivitis which is better chloramphenicol or fucithalmic eye drops?

Figure 1. Questions used in the experiment.

to the numbers retrieved by only looking for the drug names, because fewer abstracts exist for these drugs, but the relevance of the returned abstracts improves as considerably as for the more studied drugs. (Recall was not analyzed during the explorations because for answering clinical questions the relevance of the retrieved abstracts is more important than retrieving all possible abstracts.)

There have also been cases where including the basis of the comparison leads to the return of no relevant abstracts. In this case, different strategies from the one discussed above will be necessary.

Often drugs are known under generic names or the basis of the comparison is related to symptoms which are not explicitly mentioned in the question but which are still relevant. In order to recognise that different terms are actually related to the same drug or disease and belong to the same hierarchy, advantage was taken of OVID's ability to map the entities to their corresponding MeSH (Medical Subject Headings) terms and to "explode" the MeSH terms to include all of the narrower, more specific subheadings during the search.

So far the focus has been on manual retrieval of abstracts. The described search strategy of combining search terms and restricting the results to the specific publication type could have been done using a search engine which implements Boolean operators and is capable of indexing XML documents However, the description of the search strategy and the presentation of the intermediate searches, which would have been performed internally by a search engine, was regarded important to illustrate the impact of adding the basis of the comparison and the use of a publication type limit on the number of retrieved abstracts.

5 Judging the Relevance of MEDLINE® Abstracts

A initial experiment was carried out to evaluate the relevance of the abstracts retrieved from MEDLINE® via Ovid® using the strategies described in the previous section.

The experimental subjects were eight 4th year medical students, who evaluated the abstracts retrieved for twelve clinical comparison questions in which two drugs were compared to each other with respect to a particular attribute. The questions differ in syntactic structure, but they all contain comparisons of two drugs. Figure 1 shows the list of questions.

The material presented to the medical students in the experiment was created as follows: The drug names and the basis of the comparison from the natural language questions were manually mapped to their corresponding MeSH terms and used to retrieve abstracts via OVID® using the final strategy described in Section 4.

For any question, the maximum number of abstracts given to the student judges was 15, comprising up-to-15 of the most recent abstracts. In total, each judge evaluated 103 abstracts. Each abstract was assigned by each judge into one of three categories, based on the criteria given after the category label:

1. Relevant: Both drugs from the question or their generic names are mentioned in the abstracts, the drugs are directly compared to each other and the disease or the attribute with respect to which they are being compared is also mentioned and the

same as stated in the question or synonymous to it (e.g. heartburn and dyspepsia would both count as right because they are closely related).

2. Not Relevant: The drugs or their generic names are not mentioned in the abstract, the drugs are not compared and/or the disease or the attribute with respect to which they are being compared is wrong (as in different from what is stated in the question, e.g. effect on blood pressure instead of use as a painkiller).

3. Somewhat Relevant: The drugs or their generic names are mentioned but there are no single sentences indicating a comparison between them or the disease is not mentioned. If the wrong disease is mentioned, the abstract should be labeled "not relevant".

The judges were also asked to explain the reason for their choice of labels.

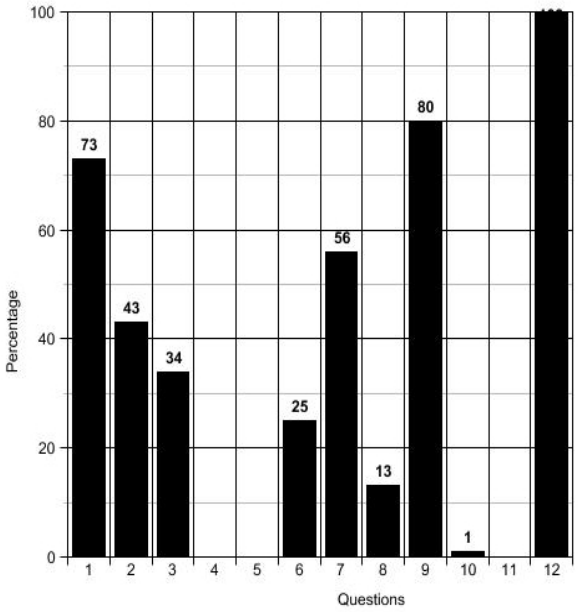
The inter-annotator agreement between the judges was computed using a variant kappa statistic for multiple annotators (Fleiss, 1971). The null hypothesis was rejected and it was ensured that the observed agreement is not accidental.

Overall inter-annotator agreement for all three categories measured by the kappa statistic was moderate at 0.58 for a total of 103 judgments. 47 judgments were in the "somewhat relevant" category. If annotator agreement is only assessed on the remaining 56 judgments from the two categories "relevant" and "not relevant", kappa is 0.97, which represents almost perfect agreement.

6 Results and Discussion

Graph 1 shows the percentage of abstracts that were judged relevant by the eight judges for each question. The numbers of retrieved abstracts for each question were: 15 abstracts for Question 1, 5, 8 and 10, 9 abstracts for question 7 and 11, 7 abstracts for Question 2, 5 abstracts for Question 9, 4 abstracts for Question 6 and 12, 3 abstracts for Question 3 and 2 abstracts for Question 4.

Question 1, 9 and 12 show a very high percentage of relevant abstracts (73%, 80% and 100% respectively), whereas no relevant abstracts were retrieved for questions 4, 5 and 11, and only one relevant abstract (out of 15) for question 10. An abstract was considered relevant when at least five of the eight judges considered it relevant.



Graph 1. Percentage of abstracts judged relevant by the majority of the judges for each of the twelve questions. The label on the top of each bar is the actual percentage.

Here the main sources for these disparate results are discussed, based on both the explanations given by the student judges and discussions with our medical librarian.

Approximately 30% (31 of 103) of the abstracts were labeled "not relevant" by the judges because they lacked any direct evidence of a comparison e.g. at least one sentence that explicitly compares the two drugs in question, even though the drugs are mentioned in the abstract and the study is a *comparative study* (as indicated in its MeSH indices). This is illustrated in Example (9), which shows the three sentences from one of the abstracts retrieved for Question 1 that explicitly mention the two drugs:

> (9) Piretanide and **furosemide** have a constant extrarenal elimination and thus accumulate in renal failure.[...] Elimination of **torasemide** is independent of its renal excretion. Thus in renal failure, **torasemide** is the only loop diuretic in which the plasma concentration is strictly dose dependent.

About 10% (10) of the abstracts were judged to be irrelevant because the drugs were compared as part of a treatment regime in combination with other drugs, as in Abstract 4 for Question 6 in which calcium citrate and calcium carbonate are compared co-administered with different preparations of sodium fluoride. In two cases (2% of the abstracts), doses of a given drug were compared against other dosages instead of the drugs themselves, e.g. 30 mg lansoprazole versus 20mg omeprazole.

A major factor for "not relevant" judgments was the time frame. This was relevant when retrieving abstracts about well-established drugs that have been in existence for a long time, such as ibuprofen or diclofenac. All but one of the 18 abstracts retrieved for the two questions about these two drugs were irrelevant, even though the two drugs were explicitly mentioned in the abstract. The problem is that they were grouped together as conventional non-steroidal anti-inflammatory drugs (NSAIDs) and compared to newer NSAIDs or different pain medication. Such abstracts could only be excluded by analyzing the abstracts themselves. Whether to proceed systematically back through the abstracts ordered by recency, or to retrieve abstracts from a random time interval, or from a window of n-years after the drug came on the market, will be a matter to be assessed empirically.

The final source of "non relevant" judgments was a problem with the judges and not with the abstracts. In Question 2 regarding dyspepsia, two out of seven abstracts were judged irrelevant because the drugs were not explicitly compared regarding dyspepsia but only regarding H. pylori, which is one of the possible causes for dyspepsia. Also abstracts retrieved for Question 7 about the effect on lipid profiles were wrongly categorised by roughly a third of the judges as not being relevant to the liver.

The experiment has shown that searching for the drugs, the basis of the comparison and studies of the publication type *comparative study* is a first step towards retrieving abstracts that can serve as answer candidates for clinical comparison questions, but it has been shown not to be sufficient to guarantee the relevance of the retrieved abstracts.

The two main problems discovered during the experiment that need to be addressed in further processing steps for the retrieved abstracts concern abstracts lacking sentences in which the drugs are directly compared to each other and the retrieval of irrelevant abstracts for well-established drugs, which are used as a reference for comparing newer drugs to, instead of containing direct comparisons of the drugs in question.

7 Conclusion and Future Work

This work introduced the task of answering clinical comparison questions and pointed out challenges in recognising and extracting their components. It also described strategies for retrieving MEDLINE® abstracts and showed that only looking for the compared entities without including the basis of the comparison is not enough to retrieve useful abstracts.

The initial experiment evaluating the relevance of retrieved abstracts for twelve clinical comparison questions revealed a number of problems that need to be taken into account for future work, especially the lack of sentences containing explicit comparisons and dealing with well-established drugs.

During the next stages, the process of identifying and extracting the elements of a comparison question as well as the process of retrieving MEDLINE® abstracts will be automated using tools from the UMLS Knowledge Sources. Features or rules will be defined to augment SemRep to deal with the problems concerning *non-scalar* comparisons and structurally different *scalar* comparison discussed in Section 3.1 to be able to automatically extract the relevant comparison components. Also, possible solutions will be researched to automatically overcome the problems of retrieving relevant abstracts identified and discussed in Section 6.

Acknowledgments

I would like to thank my advisors Bonnie Webber, Marshall Dozier and Claudia Pagliari for their helpful comments and support. I would also like to thank John Brassey for providing insight into the NLH and ATTRACT QAS.

References

Thorsten Brants. TnT – A Statistical Part-of-Speech Tagger. Available at http://www.coli.uni-saarland.de/~thorsten/publications/Brants-TR-TnT.pdf Accessed 10 August 2008.

Lacey Sue Bryant and Tim Ringrose (2005). Clinical Question Answering Services: What users want and what providers provide. Poster.

M Lee Chambliss and Jennifer Conley (1996). Answering Clinical Questions. **Journal of Family Practice** 43: 140–144.

Dina Demner-Fushman and Jimmy Lin (2006). Answer Extraction, Semantic Clustering, and Extractive Summarization for Clinical Question Answering. **Proc. COLING/ACL 2006**: 841–848.

John W Ely, Jerome A Osheroff and Mark H Ebell (1999). Analysis of questions asked by family doctors regarding patient care. **BMJ** 319: 358–361.

John W Ely, Jerome A Osheroff, M Lee Chambliss, et al. (2005). Answering physicians' clinical questions: Obstacles and potential solutions. **Journal of the American Medical Informatics Association** 12(2): 217–224.

Marcelo Fiszman, Dina Demner-Fushman, Francois M. Lang et. al. (2007). Interpreting comparative constructions in biomedical text. **Proc. BioNLP 2007**: 137–144.

Joseph L Fleiss (1971). Measuring nominal scale agreement among many raters. **Psychological Bulletin** 76 (5): 378–382.

Carol Friedman (1989). A general computational treatment of the comparative. **Proc. ACL 1989**: 161–168.

Rodney Huddleston and Geoffrey K Pullum (eds.) 2002. *The Cambridge Grammar of the English Language*. Cambridge: Cambridge University Press.

Thomas C Rindflesch and Marcelo Fiszman (2003). The interaction of domain knowledge and linguistic structure in natural language processing: Interpreting hypernymic propositions in biomedical text. **JBI** 36(6): 462–77.

Thomas C Rindflesch, M Fiszman and Bisharah Libbus (2005). Semantic interpretation for the biomedical research literature. *Medical informatics: Knowledge management and data mining in biomedicine.* Springer, New York, NY.

David L Sackett, William M C Rosenberg, J A Muir Gray, et al. (1996). Evidence based medicine: what is is and what it isn't: It's about integrating individual clinical expertise and the best external evidence. **BMJ** 312, pp. 71–72.

Silke Scheible (2008). Annotating Superlatives. **Proc. LREC 2008:** 28–30.

Bridging the Gap between Domain-Oriented and Linguistically-Oriented Semantics

Sumire Uematsu Jin-Dong Kim Jun'ich Tsujii
Department of Computer Science
Graduate School of Information Science and Technology
University of Tokyo
7-3-1 Hongo Bunkyo-ku Tokyo 113-0033 Japan
{uematsu,jdkim,tsujii}@is.s.u-tokyo.ac.jp

Abstract

This paper compares domain-oriented and linguistically-oriented semantics, based on the GENIA event corpus and FrameNet. While the domain-oriented semantic structures are direct targets of Text Mining (TM), their extraction from text is not straghtforward due to the diversity of linguistic expressions. The extraction of linguistically-oriented semactics is more straghtforward, and has been studied independentely of specific domains. In order to find a use of the domain-independent research achievements for TM, we aim at linking classes of the two types of semantics. The classes were connected by analyzing linguistically-oriented semantics of the expressions that mention one biological class. With the obtained relationship between the classes, we discuss a link between TM and linguistically-oriented semantics.

1 Introduction

This paper compares the linguistically-oriented and domain-oriented semantics of the GENIA event corpus, and suggests a factor for utilizing NLP techniques for Text Mining (TM) in the bio-medical domain.

The increasing number of scientific articles in the bio-medical domain has contributed in drawing considerable attention to NLP-based TM. An important step in NLP-based TM is obtaining the domain-oriented semantics of sentences, as shown at the bottom of figure 1. The BioInfer (Pyysalo et al., 2007) and the GENIA event corpus (Kim et al., 2008) provide annotations of such semantic structures on collections of bio-medical articles. Domain-oriented semantic structures are valuable assets because their representation suits information needs in the domain; however, the extraction of such structures is difficult due to the large gap between the text and these structures.

On the other hand, the extraction of linguistically-oriented semantics from text has long been studied in computational linguistics, and has recently been formalized as Semantic Role Labeling (Gildea and Jurafsky, 2002), and semantic structure extraction (Baker et al., 2007)(Surdeanu et al., 2008). Semantic structures in such tasks are exemplified in the middle of figure 1. The linguistically-oriented semantic structures are easier to extract, although the information is not practical to the domain.

We aim at relating linguistically-oriented frames of semantics with domain-oriented classes, thus making a step forward in utilizing the computational linguistic resources for the bio-medical TM. Of all the differences in the two type of semantics, we focused on the fact that the former frames are more sensitive to the perspective imposed by the sentence writer. In the right hand-side example of figure 1, the linguistically-oriented structure treats *PBMC*, a cell entity, as an agent; however the bio-medical structure reflects the scientific view that there are no agents, objects acting with intention, in bio-molecular phenomena.

As a preliminary investigation, we selected four representative classes of bio-molecular phenomena; Localization, Binding, Cell_adhesion, and Gene_expression, and investigated domain-oriented annotations for the classes in the GENIA

Proceedings of the Workshop on BioNLP, pages 162–170,
Boulder, Colorado, June 2009.

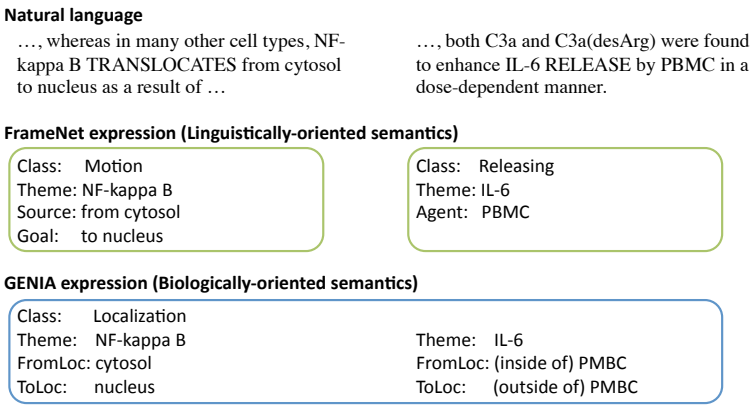


Figure 1: A comparison of the linguistically-oriented and biologically-oriented structure of semantics

event corpus. Expressions mentioning the four classes were examined and manually classified into linguistically-oriented frames, represented by those defined in FrameNet (Baker et al., 1998). FN frames associated to a bio-molecular event class constitute a list of possible perspectives in mentioning phenomena of the class.

The rest of this paper is structured in the following way: Section 2 reviews the existing work on semantic structures and expression varieties in the bio-medical domain, and provides a comparison to our work. In section 3, we describe the GENIA event corpus, and the FrameNet frames used as linguistically-oriented classes in our investigation. Sections 4 and 5 explain the methods and results of the corpus investigation; in particular the sections investigate how the linguistic frames were associated to the domain-oriented classes of semantics. Finally, we provide discussion and conclusion in section 6 and 7.

2 Related Work

Existing work on semantics approached domain-oriented semantic structures from linguistically-oriented semantics. In contrast, our approach uses domain-oriented semantics to find the linguistic semantics that represent them. We believe that the two different approaches could complement each other.

The PASbio(Wattarujeekrit et al., 2004) proposes Predicate Argument Structures (PASs), a type of linguistically-oriented semantic structures, for domain-specific lexical items, based on PASs defined in PropBank(Wattarujeekrit et al., 2004) and NomBank(Meyers et al., 2004). The PASs are defined per lexical item, and is therefore distinct from a biologically-oriented representation of events. (Cohen et al., 2008) investigated syntactic alternations of verbs and their nominalized forms which occurred in the PennBioIE corpus(Kulick et al., 2004), whilst keeping PASs of the PASBio in their minds.

The BioFrameNet(Dolbey et al., 2006) is an attempt to extend the FrameNet with specific frames to the bio-medical domain, and to apply the frames to corpus annotation. Our attempts were similar, in that both were: 1) utilizing the FN frames or their extensions to classify mentions of biological events, and 2) relating the frames and the FEs (roles of participants) with classes in domain ontologies; e.g. the Gene Ontology(Ashburner et al., 2000).

As far as the authors know, it is the first attempt to explicitly address the problem of linking linguistically-oriented and domain-oriented frames of semantics. However, it has been indirectly studied through works on TM or Relation Extraction using linguistically-oriented semantic structures as features, such as in the case with (Harabagiu et al., 2005).

3 Corpora

We used domain-oriented annotations of the GENIA event corpus and linguistically-oriented frames defined in FrameNet (FN), to link domain-oriented and linguistically-oriented frames of semantics. We briefly describe these resources next.

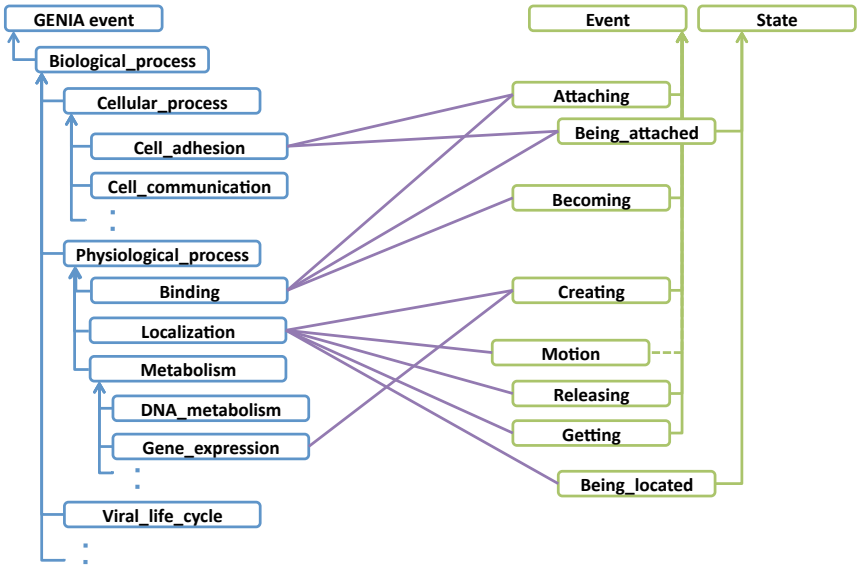


Figure 2: The resulting relationship between linguistically-oriented and biologically-oriented frames.

The GENIA event corpus consists of 1,000 Medline abstracts; that is, 9,372 sentences annotated with domain-oriented semantic structures. The annotation was completed for all mentions of biological events, and resulted in 6,114 identified events. Examples of annotated event structures are shown at the bottom of figure 1. Each structure has attributes *type* and *theme*s, which respectively show the biological class of the mentioned event and phrases expressing the event participants. The event classes are defined based on the terms in the Gene Ontology. For example, the Localization class in the GENIA event corpus is defined as an equivalent of the GO term Localization (GO0051179). The event classifications used in the corpus are depicted in the left hand-side of figure 2. Arrows in the figure depict the inheritance relations defined in the GENIA event ontology. For instance, the Localization class is defined as a type of Physiological_process. Each of the annotated structures has additional attributes that point phrases that the annotator of the structure used as a clue. Among the attributes, the *clueType* attribute shows a clue phrase to the event class. In our investigation, the attribute was treated as a predicate, or an equivalent of the lexical unit in the FN.

FN is a network of frames that are are linguistically-oriented classifications of semantics. A FN frame is defined as "a script-like conceptual structure that describes a particular type of situation, object, or event and the participants and propositions involved in it," and is associated with words, or lexical units, evoking the frame. For instance, the verbs *move*, *go* and *fly* are lexical units of the Motion frame, and they share the same semantic structure. Each FN frame has annotation examples forming an attestation of semantic overlap between the lexical units. Additionally, FN defines several types of frame-frame relations; e.g. inheritance, precedence, subframe, etc. The right hand-side of figure 2 shows some FN frames and inheritance relationships between them. The FN provides linguistically-oriented classifications of event mentions based on surface expressions, and also shows abstract relations between the frames.

4 Additional Annotation

Our aim is to link linguistically-oriented and domain-oriented frames of the bio-medical text's semantics. A major problem in this task was that there were no annotated corpora with both types of semantic structures. Therefore, we decided to concentrate on the mentions of a few classes of biological phenomena, and to annotate samples of the mentions with linguistically-oriented structures conforming to

Freq.	Keyword	Frame
693	binding	Attaching
247	bind	Attaching
125	interaction	Attaching, Being_attached
120	complex	–
99	bound	Attaching, Being_attached
91	interact	Attaching, Being_attached
61	form	Becoming
52	crosslink	Attaching
46	formation	Becoming

Table 1: The most frequent keywords of the Binding class, mentioned 2,006 times in total.

Freq.	Keyword	Frame
131	translocation	Motion
81	secretion	Releasing
75	release	Releasing
32	secrete	Releasing
25	mobilization	Motion
23	localization	Being_located
20	uptake	Getting
18	translocate	Motion
15	expression	Creating
9	present	Being_located

Table 2: The most frequent keywords of the Localization class, mentioned 582 times in total.

the FrameNet annotations.

The following provides the annotation procedures. First, we collected linguistic expressions that mention each of the selected GENIA event classes from the GENIA event corpus. We then sampled and annotated them with their linguistically-oriented semantics which conformed to the FrameNet.

4.1 Target Classes and Keywords

We concentrated mainly on the mentions of four GENIA classes; Localization, Binding, Cell_adhesion, and Gene_expression. Gene_expression, Binding, and Localization are three of the most frequent four classes in the GENIA event corpus.[1] Binding and Localization are the two most primitive molecular events. The Cell_adhesion class was included as a comparison for the Binding class.

Counting keywords for mentioning events was close to automatic. We extracted phrases pointed by a *clueType* attribute from each event structure. We then tokenized the phrases, performed a simple stemming on the tokens, and counted the resulting words. The stemming process simply replaced each inflected word to its stem by consulting a small list of inflected words with their stems. Manual work was only used in making the small list.

4.2 FN Annotation

A major challenge encountered in annotating a sampled expression with a semantic structure conforming to FN, was in the assignment of a FN frame to the mention. Our decision was based on the following four points: 1) keywords used in the mention, 2) description of FN frames, 3) syntactic positions of the event participants, and 4) frame-frame relations.

The first indicates that a FN frame became a candidate frame for the mention, if the keyword in the mention is a lexical unit of the FN frame. FN frames and their lexical units could be easily checked by consulting the FN dictionary. If there were no entries for the keyword in the dictionary, synonyms or words in the keyword's definition were used. For example, the verb *translocate* has no entries in the FN dictionary, and the frames for verbs such as *move* were used instead.

For the second point, we discarded FN frames that are either evoked by a completely different sense of the keyword, or too specific of a non-biological situations.

Before we assigned a FN frame to each mention, we manually examined the syntactic positions of all event participants present in the sampled GENIA mentions. Combinations of the syntactic position and event participants observed for a keyword were compared with sample annotations of the candidate FN frames.

We checked frame-frame relations between the candidate frames, because they can be regarded as evidence that shows that the conception of the frames is related. For our aim, it was sufficient to choose a set of frames that best describes the different perspectives for mentioning one type of molecular phenomena. Even when some keywords seemed to be dissimilar in the three points mentioned above,

[1] Except correlation and regulation classes which express relational information rather than events.

Freq.	Keyword	Frame
98	adhesion	Being_attached
19	adherence	Being_attached
16	interaction	Being_attached, Attaching
15	binding	Attaching
8	adherent	Being_attached

Table 3: The most frequent keywords of the Cell_adhesion class, mentioned 193 times in total.

Freq.	Keyword	Frame
1513	expression	Creating
357	express	Creating
239	production	Creating
71	overexpression	Creating
69	produce	Creating
62	synthesis	Creating

Table 4: The most frequent keywords of the Gene_expression class, mentioned 2,769 times in total.

a single frame could be assigned to them if it was quite clear that they shared a similar perspective. The frame-frame relations provided in the FN were treated as clues to the similarity.

Keywords frequently used in each event class are listed in tables 1, 2, 3, and 4, with the final assignment of FN frames to each keyword.

5 Analysis

After the linguistic annotation was performed, we compared the GENIA event structure and the frame structure of each sampled expression, and obtained relations of the GENIA class-FN frame and GENIA slot-FN participant. The resulting relationships between FN frames and the four GENIA classes demonstrate a gap between linguistically-oriented and domain-oriented classification of events, as shown in figure 2.

The relations can be explained by decomposing it into two cases: 1) 1-to-n mappings, and 2) n-to-1 mappings. The n-to-n mapping from GENIA to FN can then be regarded as a mix of the two cases. In the following sections, the two cases are described in detail. Further, we show conversion examples of a FN structure to a GENIA event structure, which were supported by the obtained GENIA participant-FN participant relations.

5.1 1-to-N Mapping: Different Perspectives on the Same Phenomena

A 1-to-n mapping from GENIA to FN can be explained as the case where the same molecular phenomena are expressed from different perspectives.

5.1.1 Binding Expressed in Multiple frames

The Binding class in GENIA is defined as "the selective, often stoichiometric interaction of a molecule with one or more specific sites on another molecule." We associated the class with three frames, and two frames of the three, Attaching and Becoming frames, represent different perspectives for mentioning the class. The Being_attached frame shares the same conception as Attaching, but expresses states instead of events. See table 1 for keywords of the class, and the frames assigned to the words.

Attaching: In the perspective represented by this frame, a binding phenomenon was recognized as a event in which protein molecules were simply attached to one another.

> [The 3'-CAGGTG E-box$_{\text{Item}}$] could BIND [USF proteins$_{\text{Goal}}$], $\cdots$
> (PubMed ID 10037751, Event IDs E11, E12, E13)

Becoming: In the perspective represented by this frame, a product of a binding event was treated, on the surface, as a different entity from the original parts.

> When activated, [glucocorticoid receptors$_{\text{Entity}}$] FORM [a dimer$_{\text{Final_category}}$] $\cdots$
> (PubMed ID 10191934, Event ID E5)

This type of expression was possible because a product of a binding often obtains a different functionality, and can be treated as a different type of entity. Note that this frame was not associated with the Cell_adhesion class described in section 5.2.

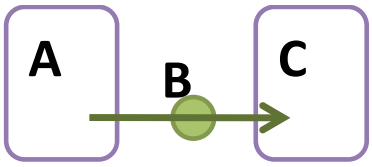


Figure 3: A schematic figure of translocation.

Being_attached: Annotators recognized a protein binding event from the sentence below, which basically mentions a state of the NF-kB.

> In T cells and T cell lines, [NF-kB$_{\text{Item}}$] is BOUND [to a cytoplasmic proteic inhibitor, the IkB$_{\text{Goal}}$].
> (PubMed ID 1958222, Event ID E2, E102)

Although this type of expression shares a similar point of view with the Attaching frame, we classified these expressions into the Being_attached frame in order to demonstrate cases in which a prerequisite Binding event was inferred from a state.

5.1.2 Translocation Expressed in Multiple Frames

The Localization class in the GENIA corpus is defined as a class for "any process by which a cell, a substance, or a cellular entity, such as a protein complex or organelle, is *transported to, and/or maintained in* a specific location." Sampled expressions of the class separated into mentions of a process, by which an entity was *transported to* a specific location, and those of the process in which an entity was *maintained in* a specific location. We concentrate on the former in this section, and describe the latter in section 5.1.3.

We associated the frames: Motion, Releasing and Getting with what we call translocation events, or Localization events in which an entity was *transported to* a specific location. Figure 3 provides a schematic representation of a translocation event. Each of the three frames had a different perspective in expressing the translocations. See table 2 for keywords of the frames.

Motion: This group consists of expressions centered on the translocated entities of the translocation - namely, B in the figure 3.

> [NK cell NFAT$_{\text{Theme}}$] $\cdots$ MIGRATES [to the nucleus$_{\text{Goal}}$] upon stimulation,$\cdots$
> (PubMed ID 7650486, Event ID E33)
> Activation of T lymphocytes $\cdots$ results in TRANSLOCATION [of the transcription factors NF-kappa B, AP-1, NFAT, and STAT$_{\text{Theme}}$] [from the cytoplasm$_{\text{Source}}$] [into the nucleus$_{\text{Goal}}$].
> (PubMed ID 9834092, Event ID E67)

These expressions are similar to those of the Motion frame in the FN.

> [Her foot$_{\text{Theme}}$] MOVED [from the brake$_{\text{Source}}$] [to the accelerator$_{\text{Goal}}$] and the car glided forward.

Releasing: This group consists of expressions centered on a starting point of the translocation - namely, A in the figure 3.

> In [unstimulated cells which$_{\text{Agent}}$] do not SECRETE [IL-2$_{\text{Theme}}$], only Sp1 binds to this region, $\cdots$
> (PubMed ID 7673240, Event ID E13)
> Activation of NF-kappaB is thought to be required for [cytokine$_{\text{Theme}}$] RELEASE [from LPS-responsive cells$_{\text{Agent}}$], $\cdots$
> (PubMed ID 1007564, Event ID E14)

The verbal keywords occurred as a transitive in most cases, and had subjects and objects that expressed starting points and entities in the translocations. This is a typical syntactic pattern of the Releasing frame, if we regarded an Agent in the FN as a starting point of the movement of a Theme.

> [The police$_{\text{Agent}}$] RELEASED [the suspect$_{\text{Theme}}$].

Getting: This group consists of expressions centered on a goal point of the translocation - namely, C in figure 3. We assumed that this group has an opposite point of view from the Releasing frame. The noun *uptake* was found to be a keyword in this group.

> The integral membrane $\cdots$ appears to play a physiological role in binding and UPTAKE [of Ox LDL$_{\text{Theme}}$] [by monocyte-macrophages$_{\text{Recipient}}$], $\cdots$
> (PubMed ID 9285527, Event ID E10)

To summarize, we observed three groups of expressions that mention translocation events, and each group represented different perspectives to mention the events. Each of the groups and the associated frame seemed similar, in that they shared similar keywords and possible syntactic positions to express the event participant.

5.1.3 Localization excluding Translocation Expressed in Multiple Frames

Localization events excluding translocations were expressed in the Being_located and Creating frames.

Being_located: This group consists of expressions that simply mention an entity in a specific location.

> $\cdots$ [recombinant NFAT1$_{\text{Theme}}$] LOCALIZES [in the cytoplasm of transiently transfected T cells$_{\text{Location}}$] $\cdots$
> (PubMed ID 8668213, Event ID E23)

Creating: A noun *expression* was observed to be used by instances mentioning the presence of proteins.

> horbol esters are required to induce [AIM/CD69$_{\text{Created_entity}}$] Cell-surface EXPRESSION as well as $\cdots$
> (PubMed ID 1545132, Event ID E12)

*Expression*s in these cases indicate an abbreviation for *gene expression*, which is a event of Gene_expression class. This type of overlap between the Localization and Gene_expression is explained in section 5.2.2

5.2 N-to-1 Mapping: Same Conception for Different Molecular Phenomenon

In contrast to the cases described in section 5.1, the same conception could be applied to different biological phenomena.

5.2.1 Shared Conception for Binding and Cell_adhesion

Molecular events classified into Binding and Cell_adhesion shared the conception that two entities were attached to each other. However, types of the entities involved are different. They are: the protein molecule in Binding, and cell in Cell_adhesion.

> CD36 is a cell surface glycoprotein $\cdots$, which INTERACTS with thrombospondin, $\cdots$, and erythrocytes parasitized with Plasmodium falciparum.

In the sentence above, an event involving *a cell surface glycoprotein* and *thrombospondin* was recognized as a Binding, whereas an event involving *a cell surface glycoprotein* and *erythrocytes* was classified as a Cell_adhesion event.

5.2.2 Shared Expressions of Localization and Gene_expression

Both Localization and Gene_expression classes are connected with the Creating frame. Some Localization events have a dependency on the Gene_expression event. Protein molecules are made in events classified into the Gene_expression class.

> [Th1 cells$_{\text{Creator}}$] PRODUCE [IL-2 and IFN-gamma$_{\text{Created_entity}}$], $\cdots$
> (PubMed ID 10226884, Event ID E11, E12)

The molecules are then translocated somewhere. Consequently, localized protein molecules might indicate a Gene_expression event, and a phrase "protein expression" was occasionally recognized as mentioning a Localization.

> horbol esters are required to induce [AIM/CD69$_{\text{Created_entity}}$] cell-surface EXPRESSION as well as $\cdots$
> (PubMed ID 1545132, Event ID E12)

5.3 Conversion of FN Structures to GENIA Events

During the investigation, we compared participant slots of GENIA and FN structures, in addition to the structures themselves. Figures 4 and 5 depict conversion examples from a FN structure and its participants to a GENIA structure, with the domain-oriented type of each participant entity. The conversions were supported by samples, and need quantitative evaluation.

6 Discussion

By annotating sentences of the GENIA event corpus with semantic structures conforming to FrameNet, we explicitly compared linguistically-oriented and

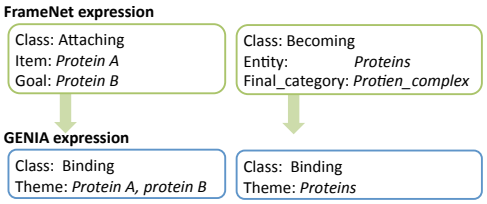


Figure 4: FN-to-GENIA conversions for Binding

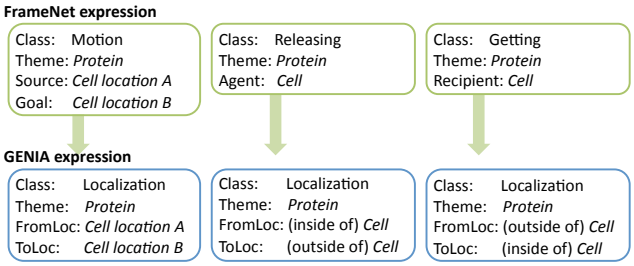


Figure 5: FN-to-GENIA conversions for Localization.

domain-oriented semantics of the bio-molecular articles. Our preliminary result illustrates the gap between the two type of semantics, and a relationship between them. We discuss development of a Text Mining (TM) system, in association with the extraction of linguistically-oriented semantics, which has been studied independently of TM.

First, our result would show that TM involves at least two qualitatively different tasks. One task is related to our results; that is, recognizing equivalent events which are expressed from different perspectives, and hence expressed by using different linguistic frames, and at the same time distinguishing event mentions which share the same linguistic frame but belong to different domain classes. Our investigation indicates that this task is mainly dependent on domain knowledge and how a phenomenon can be conceptualized. Another task of TM is the extraction of linguistically-oriented semantics, which basically maps various syntactic realizations to the shared structures. In order to develop a TM system, we need to solve the two difficult tasks.

Second, TM could benefit from linguistically-oriented frames by using them as an intermediating layer between text and domain-oriented information. The domain-oriented semantic structures, which is a target of TM, are inevitably dependent on the domain. On the other hand, the extraction of linguistically-oriented semantics from text is less dependent. Therefore, using the linguistically-oriented structure could be favorable to domain portability of a TM system.

Our aim was explicitly linking linguistically-oriented and domain-oriented semantics of the bio-molecular articles, and the preliminary result show the possibility of the extraction of linguistically-oriented semantics contributing to TM. Further investigation of the relationship would be a important step forward for TM in the bio-molecular domain.

Our investigation was preliminary. For example, conversions from FN structures to GENIA event structures, depicted in figures 4 and 5, were based on manual investigation. Further, they were attested by limited samples in the corpus. For our results to contribute to a TM system, evaluation of the conversions and automatic extraction of such conversions must be considered.

7 Conclusion

This paper presents a relationship of domain-oriented and linguistically-oriented frames of semantics, obtained by an investigation of the GENIA event corpus. In the investigation, we annotated sample sentences from the GENIA event corpus with linguistically-oriented semantic structures as those of FrameNet, and compared them with domain-oriented semantic annotations that the corpus originally possesses. The resulting relations between the domain-oriented and linguistically-oriented frames suggest that mentions of a biological phenomenon could be realized in a number of linguistically-oriented frames, and that the linguistically-oriented frames represent possible perspectives in mentioning the phenomenon. The resulting relations would illustrate a challenge in developing a Text Mining system, and would indicate importance of linguistically-oriented frames as an intermediating layer between text and domain-oriented information. Our future plan includes evaluation of our conversions from a linguistically-oriented to a domain-oriented structure, and automatic extraction of such conversions.

References

M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–29, May.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics*, pages 86–90, Morristown, NJ, USA. Association for Computational Linguistics.

Collin Baker, Michael Ellsworth, and Katrin Erk. 2007. Semeval-2007 task 19: Frame semantic structure extraction. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 99–104, Prague, Czech Republic, June. Association for Computational Linguistics.

K. Bretonnel Cohen, Martha Palmer, and Lawrence Hunter. 2008. Nominalization and alternations in biomedical language. *PLoS ONE*, 3(9):e3158, 09.

Andrew Dolbey, Michael Ellsworth, and Jan Scheffczyk. 2006. Bioframenet: A domain-specific framenet extension with links to biomedical ontologies. In *Proceedings of the Second International Workshop on Formal Biomedical Knowledge Representation: "Biomedical Ontology in Action" (KR-MED 2006)*, volume 222 of *CEUR Workshop Proceedings*. CEUR-WS.org, Nov.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Sanda M. Harabagiu, Cosmin Adrian Bejan, and Paul Morarescu. 2005. Shallow semantics for relation extraction. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 1061–1066.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(1):10.

Seth Kulick, Ann Bies, Mark Liberman, Mark Mandel, Ryan McDonald, Martha Palmer, Andrew Schein, Lyle Ungar, Scott Winters, and Pete White. 2004. Integrated annotation for biomedical information extraction. In Lynette Hirschman and James Pustejovsky, editors, *HLT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases*, pages 61–68, Boston, Massachusetts, USA, May 6. Association for Computational Linguistics.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. The nombank project: An interim report. In A. Meyers, editor, *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Bjorne, Jorma Boberg, Jouni Jarvinen, and Tapio Salakoski. 2007. Bioinfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(1):50.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The conll 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, Manchester, England, August. Coling 2008 Organizing Committee.

Tuangthong Wattarujeekrit, Parantu Shah, and Nigel Collier. 2004. Pasbio: predicate-argument structures for event extraction in molecular biology. *BMC Bioinformatics*, 5(1):155.

Evaluation of the Clinical Question Answering Presentation

Yong-Gang Cao
College of Health Sciences
University of Wisconsin Milwaukee
Milwaukee, WI 53211,USA
yonggang@uwm.edu

John Ely
Carver College of Medicine
University of Iowa
Iowa, IA 52242,USA
john-ely@uiowa.edu

Lamont Antieau
College of Health Sciences
University of Wisconsin Milwaukee
Milwaukee, WI 53211,USA
antieau@uwm.edu

Hong Yu
College of Health Sciences
University of Wisconsin Milwaukee
Milwaukee, WI 53211,USA
hongyu@uwm.edu

Abstract

Question answering is different from information retrieval in that it attempts to answer questions by providing summaries from numerous retrieved documents rather than by simply providing a list of documents that requires users to do additional work. However, the quality of answers that question answering provides has not been investigated extensively, and the practical approach to presenting question answers still needs more study. In addition to factoid answering using phrases or entities, most question answering systems use a sentence-based approach for generating answers. However, many sentences are often only meaningful or understandable in their context, and a passage-based presentation can often provide richer, more coherent context. However, passage-based presentations may introduce additional noise that places greater burden on users. In this study, we performed a quantitative evaluation on the two kinds of presentation produced by our online clinical question answering system, AskHERMES (http://www.AskHERMES.org**).** The overall finding is that, although irrelevant context can hurt the quality of an answer, the passage-based approach is generally more effective in that it provides richer context and matching across sentences.

1 Introduction

Question answering is different from information retrieval in that it attempts to answer questions by providing summaries from numerous retrieved documents rather than by simply providing a list of documents for preparing the user to do even more exploration. The presentation of answers to questions is a key factor in its efficiently meeting the information needs of information users.

While different systems have adopted a variety of approaches for presenting the results of question answering, the efficacy of the use of these different approaches in extracting, summarizing, and presenting results from the biomedical literature has not been adequately investigated. In this paper, we compare the sentence-based approach and the passage-based approach by using our own system, AskHERMES, which is designed to retrieve passages of text from the biomedical literature in response to ad hoc clinical questions.

2 Background

2.1 Clinical Question Collection

The National Library of Medicine (NLM) has published a collection of 4,653 questions that can be freely downloaded from the Clinical Questions Collection website[1] and includes the questions below:

[1] http://clinques.nlm.nih.gov/JitSearch.html

Proceedings of the Workshop on BioNLP, pages 171–178,
Boulder, Colorado, June 2009.

Question 1: "The maximum dose of estradiol valerate is 20 milligrams every 2 weeks. We use 25 milligrams every month which seems to control her hot flashes. But is that adequate for osteoporosis and cardiovascular disease prevention?"

Question 2: "Child has pectus carinatum. Radiologist told Dr. X sometimes there are associated congenital heart problems. Dr. X wants to study up on this. Does the patient have these associated problems?"

Such examples show that clinicians pose complex questions of a far greater sophistication than the simple term searches that typical information retrieval systems require as input. AskHERMES, however, has been designed to handle such complexity as it encounters it.

2.2 Result Presentation

In recent years, there has been an emergence of numerous search engines – both open domain and domain-specific – as well as question answering systems, and these systems have employed a variety of methods for presenting their results, including the use of metadata, sentences, snippets, and passages. PubMed (Anon 2009a) and EAGLi (Anon 2009b), for example, use article metadata to present their results, and the combination of title, author name and publication name that they use works like the citation at the end of a paper to provide users with a general idea of what the listed article is about. On the other hand, AnswerBus (Anon 2009c) and AnswerEngine (Anon 2009d) extract sentences from relevant articles, then rank and list them one by one to answer the questions that users have. In response to a query, Google and other general search engines provide the title of a work plus a snippet of text to provide metadata as well as multiple matching hints from articles. In response to user questions, Start (Anon 2009e), Powerset(Anon 2009f) and Ask (Anon 2009g) provide a single passage as output, making them ideal for answering simple questions because they do not require users to access and read extra articles in order to answer the questions they have.

Each of these methods of presentation has strengths and weaknesses. First, a strength of using metadata is that it provides a way for discovering the general idea of an article, but it does not explain to a user why the article is relevant to the query or question, making it difficult to decide whether it is worth the time and effort to access the listed article to read more. An approach presenting a single sentence in response to a query can result in a good answer if the user is lucky but typically provides a limited idea of what the target article contains and demands that users access the source of the item to learn more. A snippet-based approach can provide a hint as to why the target article is relevant, but snippets are limited in that they are composed of segments and usually cannot be read at all; even presenting a snippet with metadata as Google does is not suitable for adequately answering many questions.

We propose a passage-based approach in which each passage is constructed by coherent sentences. The approach we propose is similar to that used by Start and Ask, but these systems have limited knowledge bases and require queries to be written using very specific question types. On the other hand, our system will be able to answer ad hoc questions (that is, questions not limited to specific types). Furthermore, the system we propose will be oriented toward answering questions in the biomedical community, a field in which automated question answering and information retrieval and extraction are in strong demand.

3 Passage-Based Approach versus Sentence-Based Approach

We define as sentence-based approaches those approaches that return a list of independently retrieved and ranked sentences. Although all the sentences are assumed to be relevant to the question, there are no assumptions of their relationship with each other. On the other hand, a passage-based approach is defined as one that returns a list of independently retrieved and ranked passages, each of which can comprise multiple tightly coupled sentences.

The passage-based approach has two benefits:

1. It provides richer context for reading and understanding.
2. It provides greater evidence for relevant ranking of the passage by matching across sentences.

For example, in Figure 1, the passage-based output of the top results of AskHERMES pertains to the question "What is the difference between the Denver ii and the regular Denver developmental screening test?" The first answer is a passage with two sentences; the first sentence in the passage informs users that there have been

criticisms of the "Denver Developmental Screening Test," and the second sentence shows that "Denver II" addressed several concerns of the "Denver Developmental Screening Test." The two sentences indicate that the article will mention several issues that answer the question. And the second passage directly shows the answer to the question: The criteria to select Denver II and the difference between the two tests.

If we use the sentence-based approach (see Figure 2), the sentences in the first passage will be ranked very low and might not appear in the results because both of them contain only one of the screening tests mentioned in the question. The second passage will be reduced to only the second sentence, which is an incomplete answer to the question; consequently, the user may remain uninformed of the selection criteria between the two screening tests without further examination of the article. Figure 2 shows the sentence-based output of the same question. A comparison of the examples in the figure clearly shows how the results of the query are affected by the two approaches. The first result is incomplete, and the second and third results are irrelevant to the question although they have many matched terms.

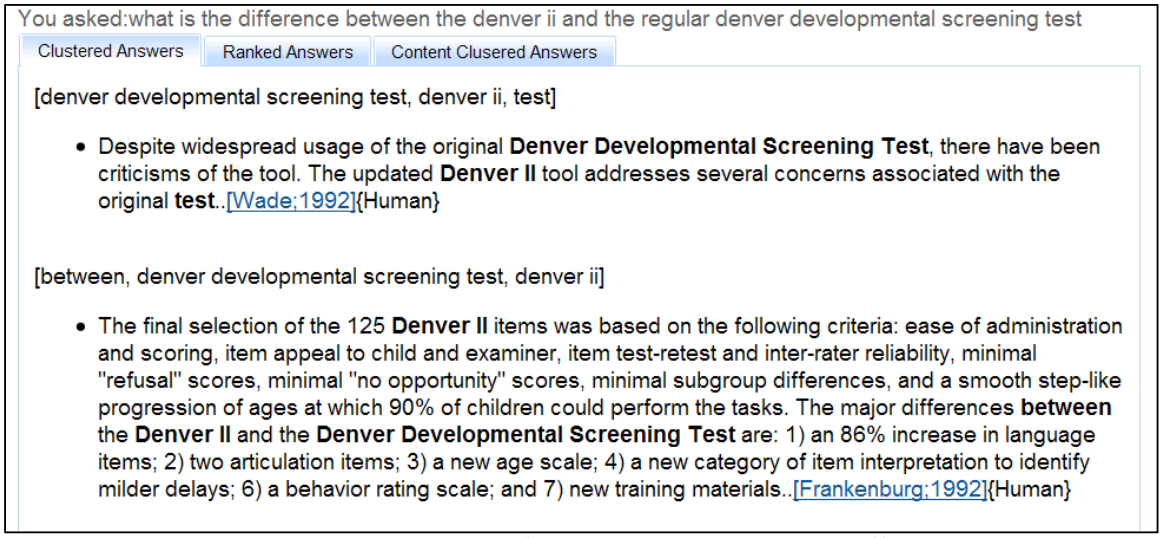


Figure 1. AskHERMES' passage-based output for the question "What is the difference between the Denver ii and the regular Denver developmental screening test?"

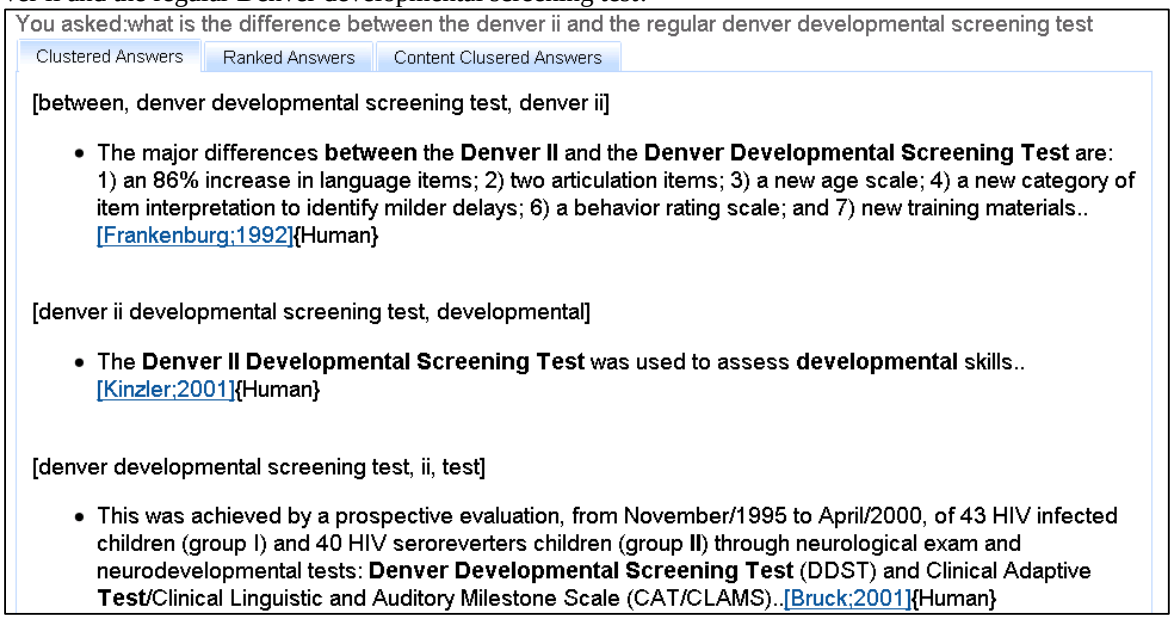


Figure 2. AskHERMES' sentence-based output for the question "What is the difference between the Denver ii and the regular Denver developmental screening test?"

While the results shown in Figures 1 and 2 suggest that a passage-based approach might be better than a sentence-based approach for question answering, this is not to say that passage-based approaches are infallible. Most importantly, a passage-based approach can introduce noisy sentences that place an additional burden on users as they search for the most informative answers to their questions. In Figure 3, the first sentence in the output of sentence-based approach answers the question. However, the passage-based approach does not answer the question until the fourth passage, and when it does, it outputs the same core answer sentence that was provided in the sentence-based approach. Additionally, the core sentence is nested within a group of sentences that on their own are only marginally relevant to the query and in effect bury the answer.

Question: which infants are candidates for synagis (respiratory-syncytial-virus immune globulin) and when should it start?

Sentence-based Approach

1. Palivizumab (**Synagis**) and **respiratory syncytial virus** intravenous **immune globulin** are licensed by the US Food and **Drug** Administration for the prevention of severe lower **respiratory** tract infections caused by **respiratory syncytial virus** in **infants** with bronchopulmonary dysplasia, **infants** with a **history** of premature birth (< or =35 weeks gestational **age**) and **children** with hemodynamically significant congenital heart **disease**..[Silvia;2005]{Human}
2. To determine outcome and length of stay (LOS) for **infants** younger than 2 **years** of **age** admitted to hospital for **respiratory syncytial virus** (RSV) infection and to perform a cost-benefit analysis of prophylaxis with RSV gamma-globulin (Respigam; CSL **Laboratories**, Melbourne, Victoria, Australia) and monoclonal antibody (**Synagis**; Abbott Australasia, Kurnell, NSW, Australia) in the Australian context..[Numa;2001]{Human}

Passage-based Approach

1. Human metapneumovirus (hMPV) is a newly recognized pathogen that **like** its better-known relative, human **respiratory syncytial virus** (hRSV), appears to be ubiquitous and an important **cause** of **respiratory disease** in diverse subpopulations. No antivirals or **vaccines** are currently approved for the **treatment** or prevention of hMPV infections. However, ribavirin is licensed to **treat** serious hRSV-induced infections in **children** and **immune globulin** designed for intravenous administration (i.v.IG) and palivizumab (**Synagis**), a humanized monoclonal antibody preparation, have been utilized as alternatives to **vaccines** for preventing or reducing the severity of infections caused by this **virus**. **Because** both ribavirin and i.v.IG have broad viral specificities, studies were performed to compare the ability of these **two** agents to inhibit the replication of hRSV and hMPV in tissue culture-based assays..[Philip;2003]{Human}
2. The Food and **Drug** Administration recently approved the use of palivizumab (pal-vizh-mäb), an intramuscularly administered monoclonal antibody preparation. Recommendations for its use are based on a **large**, randomized study demonstrating a 55% reduction in the risk of hospitalization attributable to **respiratory syncytial virus** (RSV) infections in high-risk pediatric **patients**. **Infants** and **children** with chronic lung **disease** (CLD), formerly designated bronchopulmonary dysplasia, as **well** as prematurely born **infants** without CLD experienced a reduced number of hospitalizations while receiving palivizumab compared with a placebo. Both palivizumab and **respiratory syncytial virus immune globulin** intravenous (RSV-IGIV) are **available** for protecting high-risk **children** against serious complications from RSV infections. Palivizumab is preferred for most high-risk **children because** of ease of administration (intramuscular), lack of interference with measles-mumps-rubella **vaccine** and **varicella vaccine**, and lack of complications associated with intravenous administration of human **immune globulin** products. RSV-IGIV, however, provides additional protection against other **respiratory** viral illnesses and **may** be preferred for selected high-risk **children** including those receiving replacement intravenous **immune globulin because** of underlying **immune** deficiency or human immunodeficiency **virus** infection. For premature **infants** about to be discharged from hospitals during the RSV season, physicians could consider administering RSV-IGIV for the first **month** of prophylaxis. Most of the guidelines from the American Academy of **Pediatrics** for the selection of **infants** and **children** to receive RSV-prophylaxis remain unchanged. Palivizumab has been shown to provide benefit for **infants** who were 32 to 35 weeks of gestation at birth. RSV-IGIV is contraindicated and palivizumab is not **recommended** for **children** with cyanotic congenital heart **disease**. The number of **patients** with adverse events judged to be related to palivizumab was similar to that of the placebo group (11% vs 10%, respectively); discontinuation of injections for adverse events related to palivizumab was rare..[null;1998]{Human}
3. **Respiratory syncytial virus** (RSV) continues as an emerging infectious **disease** not only among **infants** and **children**, but also for the immune-suppressed, hospitalised and the **elderly**. To date, ribavirin (Virazole, ICN Pharmaceuticals, Inc.) remains the only therapeutic agent approved for the **treatment** of RSV. However, its clinical benefits are **small** and occur only in a fraction of RSV-infected **patients**. The prophylactic administration of palivizumab (**Synagis**, MedImmune, Inc.) is problematic and costly and, therefore, only **recommended** for use in high-risk **infants**. Clearly, the need for an effective and safe **drug** remains **high**. This review discusses several different antisense approaches and compares them with traditional strategies, such as RSV-targeting antibodies and antivirals, as **well** as developments in **vaccine** research..[Hagen;2005]{Human}
4. Therefore, protection of high-risk **infants** is **possible** only by passive prophylaxis with **specific** antibodies. Palivizumab (**Synagis**) and **respiratory syncytial virus** intravenous **immune globulin** are licensed by the US Food and **Drug** Administration for the prevention of severe lower **respiratory** tract infections caused by **respiratory syncytial virus** in **infants** with bronchopulmonary dysplasia, **infants** with a **history** of premature birth (< or =35 weeks gestational **age**) and **children** with hemodynamically significant congenital heart **disease**. Palivizumab is a humanized monoclonal antibody produced by recombinant DNA technology, directed to an epitope in the A antigenic side of the F-protein of the **respiratory syncytial virus**..[Silvia;2005]{Human}

Figure 3. An example comparing the sentence-based approach and passage-based approach

4 Evaluation Design

To evaluate whether the passage-based presentation improves question answering, we plugged two different approaches into our real system by making use of either the passage-based or the sentence-based ranking and presentation unit constructor. Both of them share the same document retrieval component, and they share the same ranking and clustering strategies. In our system, we used a density-based passage retrieval strategy (Tellex et al. 2003) and a sequence sensitive ranking strategy similar to ROUGE (F. Liu and Y. Liu 2008). An in-house query-oriented clustering algorithm was used to construct the order and structure of the final hierarchical presentation. The difference between the two approaches is the unit for ranking and presentation. A passage-based approach takes the passage as its primary unit, with each passage consisting of one or more sentences. Those sentences in the passage are extracted from the adjacent matching sentences in the original article.

To evaluate the difference between the passage-based presentation and sentence-based presentation, we randomly selected 20 questions from 4,653 clinical questions. A physician (Dr. John Ely) was shown the corresponding passage-based and sentence-based outputs of every question and was then asked to judge the relevance of the output and which output had the higher quality answer. Because physicians have little time in clinical settings to be sifting through data, we presented only the top five units (sentences or passages) of output for every question.

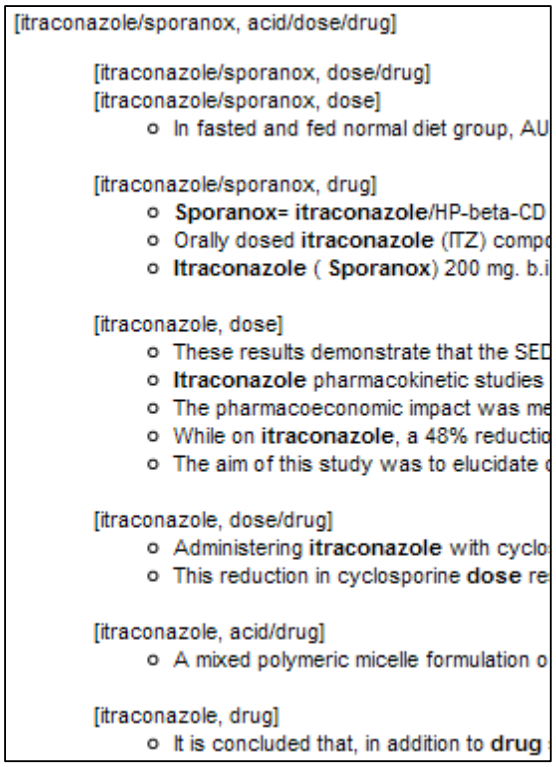


Figure 4. A partial screenshot of AskHERMES illustrating hierarchical clustering based on the question "What is the dose of sporanox?"

For answer extraction, we built a hierarchical weighted-keyword grouping model (Yu and Cao 2008;Yu and Cao 2009). More specifically, in using this model we group units based on the presence of expanded query-term categories: keywords, keyword synonyms, UMLS concepts, UMLS synonyms, and original words, and we then prioritize the groups based on their ranking. For example, units that incorporate keywords are grouped into the first cluster, followed by the cluster of units that incorporate keyword synonyms, UMLS concepts, etc. The units that appear synonymous are in the clusters with the same parent cluster. Figure 4 shows an example of the top branch of the clusters for the question "What is the dose of sporanox?" in which the answers are organized by *sporanox* and *dose* as well as their synonyms.

5 Evaluation Result and Discussion

We classify physician evaluations as being of the following four types and plot their distribution in Figure 5:

- Hard Question: The question is considered difficult because it is patient-specific or unclear (that is, it is a poorly formed question), e.g., "Multiple small ulcers on ankles and buttocks. No history of bites. I sent him for a complete blood count (cbc) and blood sugar but I don't know what these are."
- Failed Question: Neither approach can find any relevant information for the question.
- Passage Better: Passage-based approach presents more useful information for answering the question.
- Sentence Better: Sentence-based approach provides the same amount of useful information while reducing the effort required by the passage-based approach.

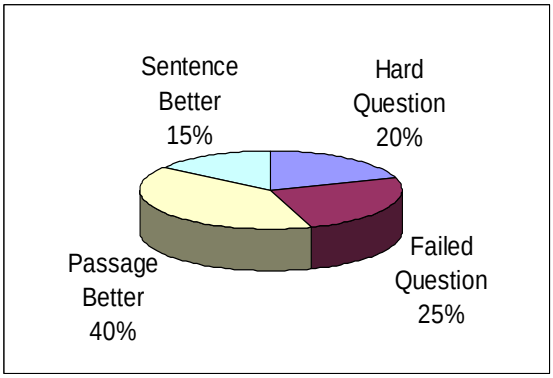


Figure 5. Distribution of the defined Evaluation categories

The evaluation data is shown in Table 1. In our study, the score range is set from 0 to 5 with the value 0 referring to answers that are totally irrelevant to the question and the value 5 meaning there is enough information to fully answer the question. Our results show that the passage-based approach is better than the sentence-based approach (p-value < 0.05).

Table 1. Quantitative measurement of the answers generated by both approaches to the 20 questions

No.	Passage-based approach score	Sentence-based approach score
1	3	1
2	2	0
3	2	0
4	0	0
5	0	0
6	1	0
7	3	1
8	3	0
9	0	0
10	0	0
11	1	2
12	1	2
13	3	4
14	0	0
15	1	0
16	2	1
17	0	0
18	1	0
19	0	0
20	0	0
mean	1.15	0.55
s.deviation	1.18	1.05
p-value	0.01	

Through further analysis of the results, we found that 70% of the sentences yielded by the sentence-based approach did not answer the question at all (the score is zero), while this was true for only 40% of the output of the passage-based approach. This indicates that the passage-based approach provides more evidence for answering questions by providing richer context and matching across sentences.

On the other hand, if the question was too general and included a plethora of detail and little focus, both approaches failed. For example, in the question "One year and 10-month-old boy removed from his home because of parental neglect. Caretaker says he often cries like he's in pain, possibly abdominal pain. Not eating, just drinking liquids, not sleeping. The big question with him: "is it something physical or all adjustment disorder?"" there is a great deal of description of the boy, and a variety of common symptoms are also provided. AskHERMES found a passage containing all of the following extracted words: "availability, because, before, between, changes, children, decrease, disorder/disorders, drug, eating, going, increase, indications/reasons, intake, laboratory, level, may, often, one, patient/patients, physical, recommended, routinely, specific, still, symptom/symptoms, two, urine, used, women, treat/treated/treating/therapy/treatment/treatments, and work." But since these words are so commonly used in a variety of scenarios, the output passage is off-topic.

For very simple questions, the sentence-based approach works well for providing answers in a very concise form. For example, the question "what is the dose of zyrtec for a 3-year-old?" can be answered by the dosage amount for the target age group, and the query resulted in this answer: "…children of both sexes aged between 2 to 6 years with allergy rhinitis (AR) were included in this study, who were randomly selected to be treated with Zyrtec (Cetirizine 2 HCL) drops 5 mg daily for 3 weeks." From a literal view, this looks like an answer to the question because it discusses the dosage of Zyrtec for the specific age group; however, it actually describes an experiment and does not necessarily provide the suggested dosage that the user is seeking. This leads to an interesting problem for clinical question answering: how should experimental data be distinguished from suggestion data for recommended daily usage? People tend to ask for the best answer instead of the possible answers. This is one of the main reasons why in Table 1, there is no perfect score (5).

Our result looks similar to the conclusion of Lin et al (Jimmy Lin et al. 2003), whose study on open-domain factoid question answering indicates a preference among users for the answer-in-paragraph approach rather than the three other types of presentation: exact-answer (that is, answer entity), answer-in-sentence, and answer-in-

document. The results of both Lin's research and our own indicate the usefulness of context, but Lin's work focuses on how surrounding context helps users to understand and become confident in answers retrieved by simple open-domain queries, while our research reveals that adjacent sentences can improve the quality of answers retrieved using complex clinical questions. Our results also indicate that context is important for relevance ranking, which has not been thoroughly investigated in previous research. Furthermore, our work places emphasis on proper passage extraction from the document or paragraph because irrelevant context can also be a burden to users, especially for physicians who have limited time for reading through irrelevant text. Our continuous sentence-based passage extraction method works well for our study, but other approaches should be investigated to improve the passage-based approach.

With respect to the quality of the answer, the content of the output is not the only important issue. Rather, the question itself and the organization of content are also important issues to consider. Luo and Tang (Luo and Tang 2008) proposed an iterative user interface to capture the information needs of users to form structured queries with the assistance of a knowledge base, and this kind of approach guides users toward a clearer and more formal representation of their questions. DynaCat (Pratt and Fagan 2000) also uses a knowledge-based approach to organize search results. Thus, applying domain-specific knowledge is promising for improving the quality of an answer, but the difficulty of the knowledge-based approach is that building and updating such knowledge bases is human labor intensive, and furthermore, a knowledge-based approach restricts the usage of the system.

6 Conclusion and Future Work

In this study, we performed a quantitative evaluation on the two kinds of presentation produced by our online clinical question answering system, AskHERMES. Although there is some indication that sentence-based passages are more effective for some question types, the overall finding is that by providing richer context and matching across sentences, the passage-based approach is generally a more effective approach for answering questions. Compared to Lin's study on open-domain factoid questions (Jimmy Lin et al. 2003), our study addresses the usefulness of context for answering complex clinical questions and its ability to improve answer quality instead of just adding surrounding context to the specific answer.

While conducting this investigation, we noticed that simple continuous sentence-based passage constructions have limitations in that they have no semantic boundary and will form too long a passage if the question contains many common words. Therefore, we will take advantage of recent advances we have made in HTML page analysis components to split documents into paragraphs and use the paragraph as the maximum passage, that is, a passage will only group sentences that appear in the same paragraph. Furthermore, by setting the boundary at a single paragraph, we can loosen the adjacency criterion of our current approach, which requires that the sentences in a passage be next to each other in the original source, and instead adopt a requirement that they only be in the same paragraph. This will enable us to build a model consisting of one or more core sentences as well as several satellite sentences that could be used to make the answer more complete or understandable.

Acknowledgments

The authors acknowledge support from the National Library of Medicine to Hong Yu, grant number 1R01LM009836-01A1. Any opinions, findings, or recommendations are those of the authors and do not necessarily reflect the views of the NIH.

References

Anon. 2009a. PubMed Home. http://www.ncbi.nlm.nih.gov/pubmed/ (Accessed: 10. March 2009).

Anon. 2009b. EAGLi: the EAGL project's biomedical question answering and information retrieval interface. http://eagl.unige.ch/EAGLi/ (Accessed: 6. March 2009).

Anon. 2009c. AnswerBus Question Answering System. http://www.answerbus.com/index.shtml (Accessed: 6. March 2009).

Anon. 2009d. Question Answering Engine. http://www.answers.com/bb/ (Accessed: 6. March 2009).

Anon. 2009e. The START Natural Language Question Answering System. http://start.csail.mit.edu/ (Accessed: 6. March 2009).

Anon. 2009f. Powerset. http://www.powerset.com/ (Accessed: 19. April 2009).

Anon. 2009g. Ask.com Search Engine - Better Web Search. http://www.ask.com/ (Accessed: 6. March 2009).

Lin, Jimmy, Dennis Quan, Vineet Sinha, Karun Bakshi, David Huynh Boris, Boris Katz and David R Karger. 2003. What Makes a Good Answer? The Role of Context in Question Answering Jimmy Lin, Dennis Quan, Vineet Sinha, Karun Bakshi, *PROCEEDINGS OF INTERACT 2003*: 25--32. doi:10.1.1.4.7644, .

Liu, F. and Y. Liu. 2008. Correlation between rouge and human evaluation of extractive meeting summaries. In: *The 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2008)*.

Luo, Gang and Chunqiang Tang. 2008. On iterative intelligent medical search. In: *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 3-10. Singapore, Singapore: ACM. doi:10.1145/1390334.1390338, http://portal.acm.org/citation.cfm?id=1390338 (Accessed: 13. March 2009).

Pratt, Wanda and Lawrence Fagan. 2000. The Usefulness of Dynamically Categorizing Search Results. *Journal of the American Medical Informatics Association* 7, Nr. 6 (December): 605–617.

Tellex, S., B. Katz, J. Lin, A. Fernandes and G. Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 41-47. ACM New York, NY, USA.

Yu, Hong and Yong-Gang Cao. 2008. Automatically extracting information needs from ad hoc clinical questions. *AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium*: 96-100.

Yu, Hong and Yong-Gang Cao. 2009. Using the weighted keyword models to improve information retrieval for answering biomedical questions. In: *To appear in AMIA Summit on Translational Bioinformatics*.

Clustering semantic spaces of suicide notes and newsgroups articles

P. Matykiewicz[1,2], **W. Duch**[2], **J. Pestian**[1]
[1]Cincinnati Children's Hospital Medical Center, University of Cincinnati,
[2]Nicolaus Copernicus University, Toruń, Poland.

Abstract

Historically, suicide risk assessment has relied on question-and-answer type tools. These tools, built on psychometric advances, are widely used because of availability. Yet there is no known tool based on biologic and cognitive evidence. This absence often cause a vexing clinical problem for clinicians who question the value of the result as time passes. The purpose of this paper is to describe one experiment in a series of experiments to develop a tool that combines Biological Markers (B_m) with Thought Markers (T_m), and use machine learning to compute a real-time index for assessing the likelihood repeated suicide attempt in the next six-months. For this study we focus using unsupervised machine learning to distinguish between actual suicide notes and newsgroups. This is important because it gives us insight into how well these methods discriminate between real notes and general conversation.

1 Introduction

It is estimated that each year 800,000 die by suicide worldwide (World Health Organization, 2001). In the United States, suicide ranks second as the leading cause of death among 25-34 year-olds and the third leading cause of death among 15-25 year-olds (Kung et al., 2008). The challenge for those who care for suicide attempters, such as an Emergency Medicine clinicians, is to assess the likelihood of another attempt, a more lethal one. We believe to fully asses this risk a tool must be developed that measures both the biological and cognitive state of the patient. Such a tool will include Biological Markers (B_m): measured by the concentration of certain biochemical markers, Thought Markers (T_m): measured by artifacts of thought that have been reduced to writing or transcribe speech, and Clinical Markers (C_m): measured by traditional clinical risk factors. In this study we focus on the T_m because of BioNLP's important role. Here, we employ machine-learning analysis to examine suicide notes and how these notes compare to newsgroups. This is one experiment in a series of experiments that are intended to provide insight into how best to apply linguistic tools when responding to suicidal patients.

To gain insight into the suicidal mind, researchers have suggested empirically analyzing national mortality statistics, psychological autopsies, nonfatal suicide attempts and documents such as suicide notes (Shneidman and Farberow, 1957; Maris, 1981). Most suicide notes analysis has focused on classification and theoretical-conceptual analysis. Content analysis has been limited to extracting explicit information from a suicide note, e.g., length of the message, words, and parts of speech (Ogilvie et al., 1969). Classification analysis uses data such as age, sex, marital status, educational level, employment status and mental disorder (Ho et al., 1998; Girdhar et al., 2004; Chavez et al., 2006; Demirel et al., 2007). Only a very few studies have utilized theoretical-conceptual analysis , despite the assertion in the first formal study of suicide notes (Shneidman and Farberow, 1957) that such an analysis has much promise. So, the inconclusive nature of the methods of analysis has limited their application to patient care.

Proceedings of the Workshop on BioNLP, pages 179–184,
Boulder, Colorado, June 2009.

Our own research has taken a different approach. In particular we first wanted to determine if modern machine learning methods could be applied to free-text from those who committed suicide. Our first experiment focused on the the ability of machine learning to distinguish between real suicide notes and elicited suicide notes as well as mental health professionals. This is an important question since all current care is based on a mental health profession's interpretation. Our findings showed that mental health professionals accurately selected genuine suicide notes 50% of the time and the supervised machine learning methods were accurate 78% (Pestian et al., 2008). In this study we shift from supervised to unsupervised machine learning methods. Even though these methods have rich history we know of no research that has applied them to suicide notes. Our rationale for this study, then, is that since our ultimate goal is to create a Suicide Risk Index that incorporates biological and thought markers it is important to determine if unsupervised methods can distinguish between suicidal and non-suicidal writings. To conduct this research we developed a corpus of over 800 suicide notes from individuals who had committed suicide, as opposed to those who attempted or ideated about suicide. This is an important contribution and, as far as we know, it is the largest ever developed. It spans 70 years of notes, and now includes multiple languages. Details of this corpus are described below. We also created a corpus of data from various newsgroups that acted as non-suicidal writings. These corpora were used to conduct the analysis. The sections below describe the cluster analysis process and results.

2 Data

Suicide Notes Corpus

Data for the suicide note database were collected from around the United States. They were either in a hand written or typed written form. Once the note was acquired it was scanned into the database. Optical character recognition was attempted on the typed written notes, but not accurate, so the notes were read from the scanned version and type into the database exactly as seen. A second person reviewed what was typed. There were limitation in collecting deceased demographics. The table 1 provides various descriptive statistics.

Newsgroup Corpus

Newsgroup data was selected because it was convenient and as close to normal discourse as we could find. We understood that and ideal comparison group would be composed of Internet blogs or e-mails that were written by suicide ideators. True, a Google query of "suicide blog" yields millions of response, a review of many of these responses shows that the data are of little use for this analysis. In our opinion, the next suitable corpora was found in a 20 newsgroup collection from the University of California in Irvine (UCI) machine learning repository[1]. Most of the newsgroups have no relevance to suicide notes. Since our hypothesis is that unsupervised learning methods can tell the difference between suicidal and non-suicidal writing we selected discussions that we believed may have some similarity to suicide writings. This selection was based on reviewing the newsgroups with experts. We had conjectured that if an unsupervised method could distinguish between similar clusters those methods could distinguish between dissimilar clusters. The newsgroups ultimately selected were *talk.politics.guns*, *talk.politics.mideast*, *talk.politics.misc*, *talk.religion.misc*. Each newsgroup contains 1000 articles (newsgroup postings). Headers and quotes from other postings were removed.

3 Methods

Basic statistics are calculated using variables extracted by Linguistic Inquiry and Word Count version 2007 software (LIWC2007) (Chung and Pennebaker, 2007). J. W. Pennebaker, C. K. Chung, M. Ireland, A. Gonzales, and R. J. Booth created an annotated dictionary. Each word in the dictionary is assigned to at least one of the following high level category: linguistic process, psychological process, personal concern, or spoken category. These categories provide an efficient and effective method for studying the various emotional, cognitive, and structural components present in individuals' verbal and written speech samples (Chung and Pennebaker, 2007; Pennebaker et al., 2001). Here it is used to analyze differences between suicide notes and news-

[1] http://archive.ics.uci.edu/ml/

group articles.

Feature space was prepared using open source algorithms available in *Perl* language[2]. First, Brian Duggan spell checking software that uses aspell library was used (*Text::SpellChecker* module[3]). Then, tokenizer created by Aaron Coburn was used (*Lingua::EN::Tagger* module[2]) to extract words was applied. After that, words were filtered with 319 element stop word list [4]. Next, the Richardson/Franz English stemmer was included in the pre-processing stage (*Lingua::Stem* module[2]). Features that appeared in less than 10 documents or in more than 500 documents were removed. Documents that had less than 10 features or more than 500 were removed. Finally, columns and rows were normalized to have unitary lengths. These last steps of pre-processing are used to reduce outliers.

Calculations are done using open source software called *R*[5]. Clustering is done with the following algorithms: expectation maximization (EM) (Witten and Frank, 2000), simple k-means with euclidean distance (SKM) (Witten and Frank, 2000), and sequential information bottleneck algorithm (sIB) (Slonim et al., 2002). The last approach has been shown to work well work well when clustering documents. Specificity, sensitivity and F1 measure are used as performance measures (Rijsbergen, 1979). Multidimensional scaling with euclidean distance measures is used for visualization purposes (Cox and Cox, 1994).

To extract features that represent each cluster, Pearson correlation coefficient is used. The correlation coefficient r is calculated between each feature and each cluster separately $r(w_i, c_j)$ where w_i is ith word and c_j is jth cluster. N best features with the highest values for each cluster are selected as most representative.

4 Results

Descriptive statistics for the data sets are listed in table 1. It shows syntactic differences between language use in suicide notes and newsgroups when *Lingua::EN::Tagger* is used.

[2]http://www.perl.org

[3]http://search.cpan.org

[4]http://www.dcs.gla.ac.uk/idom/ir_resources/
/linguistic_utils/stop_words

[5]http://www.r-project.org

Table 1: Descriptive statistics of suicide note corpus and newsgroups.

	suicide corpus	**newsgroups**
Sample Size	866	4000 (1000 per group)
Collection Years	1945-2009	1992-1993
Avg tokens per record (SD)	105 (154)	243 (582)
Range of tokens per record	1-1837	0-11024
Average (SD) nouns	25.21 (34.81)	77.19 (181.63)
Average (SD) pronouns	16.58 (26.69)	18.05 (63.18)
Average (SD) verbs	21.07 (32.82)	41.31 (109.23)
Average (SD) adjectives	6.43 (9.81)	16.92 (36.45)

Table 2 summarizes information about the linguistic and psychological processes of the data. The idea of "process" is derived from the Linguistic Inquiry and Word Count (LIWC2007) software (Chung and Pennebaker, 2007). This software conducts traditional natural language processing by placing various word into categories. For example, *sixltrs* includes words that are at least six letters in length. A full description of this software, dictionaries, reliability and validity tests can be found on LIWC's website. [6]. Table 2 shows that suicide notes are, in many ways, different than normal text. For our study this provides inspiration for continued research.

Table 2: Mean and standard deviation in linguistic and psychological processes. Selected categories with smallest p-values ($<$0.0001) are shown.

	suicide	**guns**	**mideast**	**politics**	**religion**
artcl	3.31 (2.79)	7.80 (3.52)	7.37 (3.34)	7.21 (3.40)	7.07 (3.51)
sixltrs	14.20 (7.34)	21.22 (6.32)	23.24 (7.03)	22.41 (7.13)	21.37 (7.87)
prnoun	16.75 (6.82)	11.96 (5.15)	10.64 (4.92)	11.77 (5.18)	13.21 (5.76)
prepos	10.61 (4.35)	12.13 (3.97)	12.89 (3.89)	12.21 (3.97)	11.75 (4.07)
verb	14.69 (5.99)	12.75 (4.72)	11.54 (4.74)	12.72 (4.63)	13.54 (4.97)
biolog	2.70 (3.04)	0.93 (1.27)	0.85 (1.50)	1.59 (2.08)	1.10 (1.75)
affctiv	7.71 (5.39)	4.83 (2.87)	4.77 (3.45)	4.90 (3.18)	5.10 (3.93)
cognitv	12.68 (5.76)	16.14 (5.93)	14.72 (5.62)	16.00 (5.49)	17.14 (6.17)
social	10.45 (5.86)	8.10 (4.20)	8.43 (4.71)	8.76 (4.37)	9.06 (5.17)

The four newsgroup data sets are combined as follows: *talk.politics.guns* + suicide notes = guns, *talk.politics.mideast* + suicide notes = mideast, *talk.politics.misc* + suicide notes = politics,

[6]http://www.liwc.net/liwcdescription.php#index1

talk.religion.misc + suicide notes = religion. Each data set contained 1866 documents before document and feature selection is applied. Table 3 has final number of features while table 4 has final number of documents. In general sIB clustering algorithm performed best for all data sets with respect to F1 measure (mean = 0.976, sd = 0.008). The average score also did not change when the number of clusters varied from two to six (mean = 0.973, sd = 0.012). Performance of k-means and expectation maximization algorithm was much worse. If number of clusters was varied between two and six for different data sets the algorithms achieved F1 measure 0.146 lower than sIB (SKM mean = 0.831, sd = 0.279, EM mean = 0.824, sd = 0.219). Table 3 summarizes performance of best algorithms for each data set if two clusters are chosen.

Table 3: Best clustering algorithms for each newsgroup when clustered with suicide notes in case of two clusters (alg = clustering algorithm, sens = sensitivity, spec = specificity, F1 = F1 measure, #f = number of features, sIB = sequential information bottleneck, SKM = simple k-means).

dataset	alg	sens	spec	F1	#f
guns	sIB	.9689	.9834	.9721	1658
mideast	sIB	.9837	.9942	.9877	2023
politics	SKM	.9705	.9889	.9769	1694
religion	sIB	.9787	.9700	.9692	1553

If the desired number of clusters is increased to four then two major sub-groups are discovered in suicide notes: emotional (represented by words like: *love*, *forgive*, *hope*, and *want*) and non-emotional (represented by words like: *check*, *bank*, and *notify*). Example of the first type of note might be (suicide note was annonymized and misspellings left unchanged):

Jane I am bitterly sorry for what I have done to you. Please try to forgive me. I can't live without you and you don't want me. I can't blame you though. But I love you very much. I didn't act like it but I did and still do. Please try to be happy, Jane. That is all I ask. I try hope for the best for you and I guess that is all there is for me to say. Good by. John Johnson. Please mail this to Mom. Mrs. Jane Johnson. Cincinnati, OH.

Example of a non-emotional suicide note might be:

There is no use living in pains. That arthritis and hardening of the arteries are too much for me. There are two hundred and five dollars in the bank, and here are fifty- five dollars and eight cents. I hope that will be enough for my funeral. You have to notify the Old Age Assistance Board. Phone - 99999.

Table 4 shows best five ranked features for each cluster for each data set according to correlation coefficient CC. Features are in the order of rank so that feature with the highest CC is first. Even though that we use different newsgroups as control groups same sub-groups of suicide notes are discovered. sIB is the most stable and best performing algorithm in this experiment so it was used to discover those clusters. Stemmed word that appear in best five ranked features in at least three data sets are marked bold.

Figures 1, 2, 3, and 4 show high-dimensional document/stemmed word feature space projected on a two dimensional plane using multidimensional scaling (MDS) initialized by principal component analysis. Each figure has different rotation but the shapes are similar. In addition MDS shows very little mixing of suicide notes and newsgroups which is also explained by results in the table 3.

Figure 1: MDS showing suicide notes and *talk.politics.guns* articles (**s** character in the figure means suicide note while **a** character depicts newsgroup article, colors are used as cluster numbers).

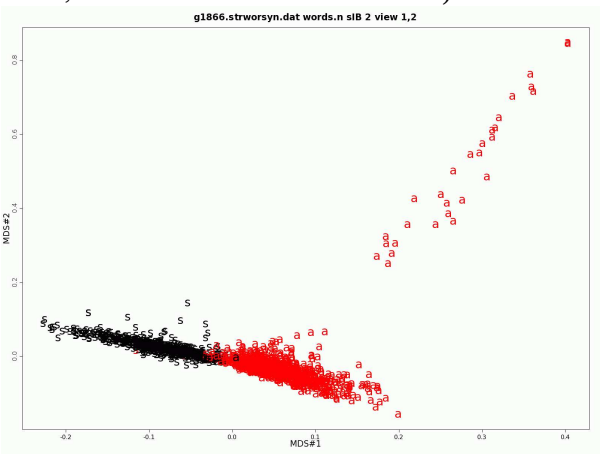


Table 4: Best five features when four clusters are created by the sIB algorithm (#c = cluster number, #a = number of newsgroup articles in a cluster, #s = number of suicide notes in a cluster). Stemmed word that appear in best five ranked features in at least three data sets are marked bold.

dataset	#c	stemmed words	#a	#s
guns	1	address, **bank**, bond, **notifi**, testam	28	204
guns	2	clinton, fbi, foreign, jim, spea	318	2
guns	3	**forgiv**, god, **hope**, **love**, **want**	4	381
guns	4	crime, firearm, gun, law, weapon	541	8
mideast	1	appressian, armenia, armenian, ohanu, proceed	464	5
mideast	2	arab, congress, isra, israel, jew	379	4
mideast	3	**bank**, **check**, funer, insur, testam	10	233
mideast	4	**forgiv**, good, **hope**, **love**, **want**	2	355
politics	1	compound, disclaim, fbi, govern, major	593	12
politics	2	clayton, cramer, optilink, relat, uunet	274	1
politics	3	**bank**, box, **check**, funer, **notifi**	11	258
politics	4	**forgiv**, good, **hope**, life, **love**	11	330
religion	1	**bank**, bond, **check**, **notifi**, paper	36	192
religion	2	frank, object, observ, theori, valu	279	0
religion	3	activ, christian, jesu, koresh, net	502	10
religion	4	**forgiv**, **hope**, **love**, sorri, **want**	12	395

Figure 2: MDS showing suicide notes and *talk.politics.mideast* articles (**s** character in the figure means suicide notes while **a** character depicts newsgroup article, colors are used as cluster numbers).

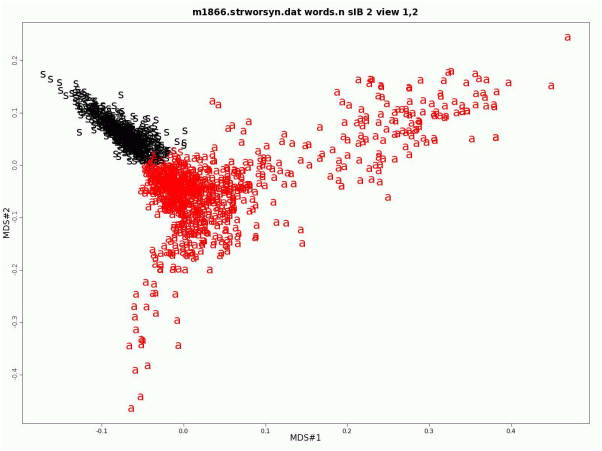
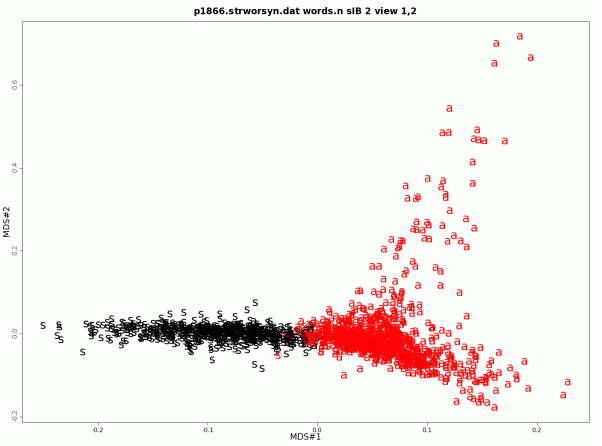


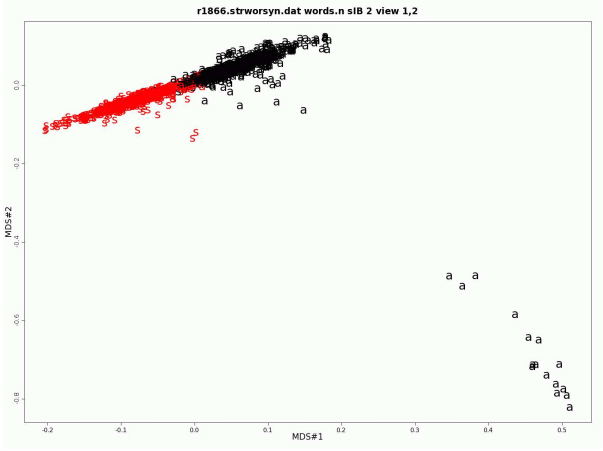
Figure 3: MDS showing suicide notes and *talk.politics.misc* articles (**s** character in the figure means suicide note while **a** character depicts newsgroup article, colors are used as cluster numbers).



5 Conclusions

Our findings suggest that unsupervised methods can distinguish between suicide notes and newsgroups, our proxy for general discussion. This is important because it is helpful in determining if NLP can be useful when integrating thought markers with biological and clinical markers ($f(B_m, T_m, C_m)$). In other words, can an NLP tools accurately distinguish between suicidal and normal thought markers ($T_m^S \neq T_m^N$)? Moreover these unsupervised methods have shown an ability to find sub-groups of suicide notes even when other types of newsgroups are present. In our analysis, one subgroup showed no emotional content while the other was emotionally charged. This finding is consistent with Tuckman's, 1959 work that showed suicide notes fall into six emotional categories: emotionally neutral, emotionally positive, emotionally negative directed inward, emotionally negative directed outward, emotionally negative directed inward and outward (Tuckman et al., 1959). The next step in developing a Suicide Risk Index is to conduct a clinical trail in the Emergency Department that will collect B_m, T_m, C_m and test multiple methods for computing the Suicide

Figure 4: MDS showing suicide notes and *talk.religion.misc* articles (**s** character in the figure means suicide note while **a** character depicts newsgroup article, colors are used as cluster numbers).



Risk Index.

References

A. Chavez, D. Paramo-Castillo, A. Leenaars, and L. Leenaars. 2006. Suicide notes in mexico: What do they tell us? Suicide and Life-Threatening Behavior, 36:709–715.

C.K. Chung and J.W. Pennebaker, 2007. The psychological functions of function words, pages 343–359. New York: Psychology Press.

T. F. Cox and M. A. A. Cox. 1994. Multidimensional Scaling. Chapman and Hall.

B. Demirel, T. Akar, A. Sayin, S. Candansayar, and A Leenaars. 2007. Farewell to the world: Suicide notes from turkey. Suicide and Life-Threatening Behavior, 38:123–128.

S. Girdhar, A. Leenaars, T.D. Dogra, L. Leenaars, and G. Kumar. 2004. Suicide notes in india: what do they tell us? Archives of Suicide Research, 8:179–185.

T. Ho, P. Yip, C. Chiu, and P. Halliday. 1998. Suicide notes: what do they tell us? Acta Psychiatrica Scandinavica, 98:467–473.

Hsiang-Ching Kung, Donna L. Hoyert, Jiaquan Xu, and Sherry L. Murphy. 2008. Deaths: Final data for 2005. National Vital Statistics Report, 56:1–121.

R. Maris. 1981. Pathways to suicide. John Hopkins University Press, Baltimore, MD.

D Ogilvie, P. Stone, and E. Shneidman. 1969. Some characteristics of genuine versus simulated suicide notes. Bulletin of Suicidology, 1:17–26.

J. W. Pennebaker, M. E. Francis, and R. J. Booth. 2001. Linguistic Inquiry and Word Count: LIWC. Lawrence Erlbaum Associates, Mahwah, NJ, 2nd edition.

J. P. Pestian, P. Matykiewicz, J. Grupp-Phelan, S. Arszman-Lavanier, J. Combs, and Robert Kowatch. 2008. Using natural language processing to classify suicide notes. In AMIA Annual Symposium Proceedings, volume 2008. American Medical Informatics Association.

C. J. Van Rijsbergen. 1979. Information Retrieval. Butterworth-Heinemann, Newton, MA, USA.

E Shneidman and N Farberow. 1957. Clues to Suicide. McGraw Hill Paperbacks.

Noam Slonim, Nir Friedman, and Naftali Tishby. 2002. Unsupervised document classification using sequential information maximization. In Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 129–136.

Jacob Tuckman, Robert J. Kleiner, and Martha Lavell. 1959. Emotional content of suicide notes. Am J Psychiatry, 116(1):59–63.

Ian H. Witten and Eibe Frank. 2000. Data mining: practical machine learning tools and techniques with Java implementations. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

World Health Organization, 2001. Burden of mental and behavioral disorders, pages 19–45. World Health Organization, Geneva.

TEXT2TABLE: Medical Text Summarization System based on Named Entity Recognition and Modality Identification

Eiji ARAMAKI
The university of Tokyo
eiji.aramaki@gmail.com

Yasuhide MIURA
Fuji Xerox
Yasuhide.Miura@fujixerox.co.jp

Masatsugu TONOIKE
Fuji Xerox
masatsugu.tonoike@fujixerox.co.jp

Tomoko OHKUMA
Fuji Xerox
ohkuma.tomoko@fujixerox.co.jp

Hiroshi MASHUICHI
Fuji Xerox
hiroshi.masuichi@fujixerox.co.jp

Kazuhiko OHE
The university of Tokyo Hospital
kohe@hcc.h.u-tokyo.ac.jp

Abstract

With the rapidly growing use of electronic health records, the possibility of large-scale clinical information extraction has drawn much attention. It is not, however, easy to extract information because these reports are written in natural language. To address this problem, this paper presents a system that converts a medical text into a table structure. This system's core technologies are (1) medical event recognition modules and (2) a negative event identification module that judges whether an event actually occurred or not. Regarding the latter module, this paper also proposes an SVM-based classifier using syntactic information. Experimental results demonstrate empirically that syntactic information can contribute to the method's accuracy.

1 Introduction

The use of electronic texts in hospitals is increasing rapidly everywhere. This study specifically examines discharge summaries, which are reports generated by medical personnel at the end of a patient's hospital stay. They include massive clinical information about a patient's health, such as the frequency of drug usage, related side-effects, and correlation between a disease and a patient's actions (e.g., smoking, drinking), which enables unprecedented large-scale research, engendering promising findings.

Nevertheless, it is not easy to extract clinical information from the reports because these reports are written in natural language. An example of a discharge summary is presented in Table 1. The table shows records that are full of medical jargon, acronyms, shorthand notation, misspellings, and sentence fragments (Tawanda et al., 2006).

To address this problem, this paper presents a proposal of a system that extracts medical events and date times from a text. It then converts them into a table structure. We designate this system ***TEXT2TABLE***, which is available from a web site[1]. The extraction method, which achieves a high accuracy extraction, is based on Conditional Random Fields (CRFs) (Lafferty et al., 2001).

Another problem is posed by events that do not actually occur, i.e., future scheduled events, events that are merely intended to take place, or hypothetical events. As described herein, we call such non-actual events **negative events**. Negative events are frequently mentioned in medical records; actually, in our corpus, 12% of medical events are negative. Several examples of negative events (in italic letters) are presented below:

(1) *no* **headache**
(2) keep *appointment* of **radiotherapy**
(3) .. *will* have **intravenous fluids**

[1] http://lab0.com/

Proceedings of the Workshop on BioNLP, pages 185–192,
Boulder, Colorado, June 2009.

(4) .. came *for* **radiotherapy**
(4') .. came *for* **headache**
(5) Every week **radiation therapy** and chemical therapy *are scheduled*
(6) *Please* call Dr. Smith with worsening **headache** or back pain, or any other concern.

Negative events have two characteristics. First, various words and phrases indicate that an event is negative. For this study, such a word or phrase that makes an event negative is called a *negative trigger*. For instance, a negation word "*no*" is a negative trigger in (1). A noun "*appointment*" in (2) is a negative trigger. Similarly, the auxiliary "*will*" in (3) signals negation. More complex phenomena are presented in (4) and (4'). For instance, "**radiotherapy**" in (4) is a negative event because the therapy will be held in the future. In contrast, "**headache**" in (4') is not negative because a patient actually has a "**headache**". These indicate that a simple rule-based approach (such as a list of triggers) can only imply classification of whether an event is negative or not, and that information of the event category (e.g., a therapy or symptom) is required.

Another characteristic is a long scope of a negative trigger. Although negative triggers are near the descriptive words of events in (1)–(4), there could alternatively be a great distance of separation, as portrayed in (5) and (6). In (5), a noun coordination separates a negative trigger from the event. In (6), the trigger "*please*" renders all events in that sentence negative. These indicate that neighboring words are insufficient to determine whether an event is negative or not. To deal with (5), syntactic information is helpful because the trigger and the event are neighboring in the dependency structure, as portrayed in Fig. 2. To deal with (6), bag-of-word (BOW) information is desired.

Because of the observation described above, this paper presents a proposal of a classifier: whether an event is negative or not. The proposed classifier uses various information, the event category, neighboring words, BOW, and dependent phrases.

The point of this paper is two-fold: (1) We propose a new type of text-summarizing system (TEXT2TABLE) that requires a technique for a negative event identification. (2) We investigate what kind of information is helpful for negative event identification.

The experiment results revealed that, in spite of the risk of parsing error, syntactic information can contribute to performance, demonstrating the feasibility of the proposed approach.

Although experiments described in this paper are related to Japanese medical reports, the proposed method does not depend on specific languages or domains.

Table 1: A Health Record Sample.

BRIEF RESUME OF HOSPITAL COURSE : 57 yo with NSCLCa with back pain and headache . Transferred from neurosurgery for additional mgmt with palliative XRT to head . Pt initially presented with cough and hemoptysis to his primary MD . On CXR he was found to have a upper left lobe mass . He subsequently underwent bronchoscopy and bx revealed non-small cell adeno CA. STaging revealed multiple bony mets including skull, spine with MRI revealing mild compression of vertebral bodies at T9, T11, T12 . T9 with encroachment of spinal cord underwent urgent XRT with no response so he was referred to neurosurgery for intervention . MRI-rt. frontal, left temporal, rt cerebellar hemorrhagic enhancing lesions- most likely extensive intracranial mets– T-spine surgery considered second priority and plan to radiate cranially immediately with steroid and anticonvulsant . He underwent simulation on 3/28 to whole brain and T3-T7 fields with plan for rx to both sites over 2.5 weeks. Over the past 2 weeks he has noted frontal and occipital HA with left eyelid swelling, ptosis, and denies CP, SOB, no sig. BM in past 5 days, small amt of stool after suppository. Neuro–He was Dilantin loaded and a level should be checked on 3/31 . He is to continue Decadron . Onc–He is to receive XRT on 3/31 and daily during that week . Pain control–Currently under control with MS contin and MSIR prn. regimen . Follow HA, LBP. ENDO–Glucose control monitored while on decadron with SSRI coverage . Will check HgbA1C prior to discharge . GI–Aggressive bowel regimen to continue at home . Pt is Full Code . ADDITIONAL COMMENTS: Please call Dr. Xellcaugh with worsening headache or back pain, or any other concern . Keep appointment as scheduled with XRT . Please check fingerstick once a day, and record, call MD if greater than 200 .

Figure 1: Visualization result (Left), magnified (Right).

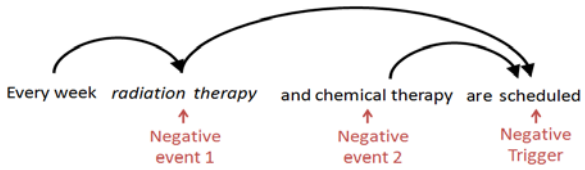


Figure 2: Negative Triggers and Events on a Dependency Structure.

Table 2: Corpora and Modalities

CORPUS	MODALITY
ACE	asserted, or other
TIMEML	must, may, should, would, or could
Prasad et al., 2006	assertion, belief, facts or eventualities
Saurí et al., 2007	certain, probable, possible, or other
Inui et al., 2008	affirm, infer, doubt, hear, intend, ask, recommend, hypothesize, or other
THIS STUDY	S/O, necessity, hope, possible, recommend, intend

Table 3: Markup Scheme (Tags and Definitions)

Tag	Definition (Examples)
R	Remedy, Medical operation (e.g. radiotherapy)
T	Medical test, Medical examination (e.g., CT, MRI)
D	Deasese, Symptom (e.g., Endometrial cancer, headache)
M	Medication, administration of a drug (e.g., Levofloxacin, Flexeril)
A	patient action (e.g., admitted to a hospital)
V	Other verb (e.g., cancer spread to ...)

2 Related Works

2.1 Previous Markup Schemes

In the NLP field, fact identification has not been studied well to date. Nevertheless, similar analyses can be found in studies of sentence modality.

The Automatic Content Extraction (ACE)[2] information extraction program deals with event extraction, by which each event is annotated with temporal and modal markers.

A similar effort is made in the TimeML project (Pustejovsky et al., 2003). This project specifically examines temporal expressions, but several modal expressions are also covered.

Prasad et al. (2006) propose four factuality classifications (certain, probable...etc.) for the Penn Discourse TreeBank (PDTB)[3].

Saurí et al. (2007) propose three modal categories for text entailment tasks.

Among various markup schemes, the most recent one is *Experience Mining* (Inui et al., 2008), which collects personal experiences from the web. They also distinguish whether an experience is an actual one or not, which is a similar problem to that confronting us.

Table 2 portrays a markup scheme adopted by each project. Our purpose is similar to that of Experience Mining. Consequently, we fundamentally adopt its markup scheme. However, we modify the label to suit medical mannerisms. For example, "doubt" is modified into "(S/O) suspicion of". Rare modalities such as "hear" are removed.

2.2 Previous Algorithms

Negation is a traditional topic in medical fields. Therefore, we can find many previous studies of the topic in the relevant literature.

An algorithm, *NegEx*[4] was proposed by Chapman et al. (Chapman et al., 2001a; Chapman et al., 2001b). It outputs an inference of whether a term is positive or negative. The original algorithm is based on a list of negation expressions. Goldin et al. (2003) incorporate machine learning techniques (Naïve Bayes and decision trees) into the algorithm. The extended version (ConText) was also proposed (Chapman et al., 2007).

Elkin et al. (2005) use a list of negation words and a list of negation scope-ending words to iden-

[2] http://projects.ldc.upenn.edu/ace/
[3] http://www.seas.upenn.edu/~pdtb/
[4] http://www.dbmi.pitt.edu/chapman/NegEx.html

tify negated statements and their scope. Their technique was used in The MAYO Clinic Vocabulary Server (MCVS)[5], which encodes clinical expressions into medical ontology (SNOMED-CT) and identifies whether the event is positive or negative.

Mutalik et al. (2001) earlier developed *Negfinder* to recognize negated patterns in medical texts. Their system uses regular expressions to identify words indicating negation. Then it passes them as special tokens to the parser, which makes use of the single-token look-ahead strategy.

Huang and Lowe (2007) implemented a hybrid approach to automated negation detection. They combined regular expression matching with grammatical parsing: negations are classified based on syntactic categories. In fact, they are located in parse trees. Their hybrid approach can identify negated concepts in radiology reports even when they are located distantly from the negative term.

The Medical Language Extraction and Encoding (MedLEE) system was developed as a general natural language processor to encode clinical documents in a structured form (Friedman et al., 1994). Negated concepts and certainty modifiers are also encoded within the system.

Veronika et al. (2008) published a negation scope corpus[6] in which both negation and uncertainty are addressed.

Although their motivations are identical to ours, two important differences are apparent. (1) Previous (except for Veronika et al., 2008) methods deal with the two-way problem (positive or negative), whereas the analyses proposed herein tackle more fine-grained modalities. (2) Previous studies (except for Huang et al., 2007) are based on BOW approaches, whereas we use syntactic information.

3 Medical Text Summarization System: TEXT2TABLE

Because the core problem of this paper is to identify negative events, this section briefly presents a description of the entire system, which consists of four steps. The detailed algorithm of negative identification is explained in Section 4.

STEP 1: Event Identification

First, we define the event discussed in this paper. We deal with events of six types, as presented in Table 3. Two of the four are Verb Phrases (base VPs); the others are noun phrases (base-NPs). Because this task is similar to Named Entity Recognition (NER), we use the state-of-the art NER method, which is based on the IOB2 representation and Conditional Random Fields (CRFs). In learning, we use standard features, as shown in Table 4.

Table 4: Features for Event Identification

Lexicon and Stem	Current target word (and its stem) and its surrounding words (and stem). The window size is five words (-2, -1, 0, 1, 2).
POS	Part of speech of current target word and its surrounding words (-2, -1, 0, 1, 2). The part of speech is analyzed using a POS tagger[7].
DIC	A fragment for the target word appears in the medical dictionary (Ito et al., 2003).

STEP 2: Normalization

As described in Section 1, a term in a record is sometimes an acronym: shorthand notation. Such abbreviations are converted into standard notation through (1) date time normalization or (2) event normalization.

(1) Date Time Normalization

As for date time expressions, relative date expressions are converted into YYYY/MM/DD as follows.

On Dec Last year → 2007/12/XX
10 Dec 2008 → 2008/12/10

These conversions are based on heuristic rules.

(2) Event Normalization

Medical terms are converted into standard notation (dictionary entry terms) using orthographic disambiguation (Aramaki et al., 2008).

STEP 3: TIME–EVENT Relation Identification

Then, each event is tied with a date time. The current system relies on a simple rule (i.e., an event is tied with the latest date time).

STEP 4: Negative Identification

The proposed SVM classifier distinguishes negative events from other events. The detailed algorithm is described in the next section.

4 Modality Identification Algorithm

First, we define the negative. We classify modality events into eight types (Table 5). These classifications are motivated by those used in previous stud-

[5] http://mayoclinproc.highwire.org/content/81/6/741.figures-only

[6] www.inf.u-szeged.hu/rgai/bioscope

[7] http://chasen-legacy.sourceforge.jp/

ies (Inui et al., 2008). However, we simplify their scheme because several categories are rare in this domain.

These classes are not exclusive. For that reason, they sometimes lead to multiple class events. For example, given "No **chemotherapy** is planned", an event "**chemotherapy**" belongs to two classes, which are "NEGATION" and "FUTURE".

Training Phase

Using a corpus with modality annotation, we train a SVM classifier for each category. The training features come from four parts:

(1) Current phrases: words included in a current event. We also regard their STEMs, POSs, and the current event category as features.

(2) Surrounding phrases: words included in the current event phrase and its surrounding two phrases (p1, p2, n1, n2, as depicted in Fig. 3). The unit of the phrase is base-NP/VP, which is produced by the Japanese parser (Kurohashi et al., 1994). Its window size is two in the neighboring phrase (p1, p2, c, n1, n2). We also deal with their STEMs and POSs.

(3) Dependent phrases: words included in the parent phrase of the current phrase (d1 in Fig. 3), and grandparent phrases (d2 in Fig. 3). We also deal with their STEMs and POSs.

(4) Previous Event: words (with STEMs and POSs) included in the previous (left side) events. Additionally, we deal with the previous event category and the modality class.

(5) Bag-of-words: all words (with STEMs and POSs) in the sentence.

TEST Phrase

During the test, each SVM classifier runs.

Although this task is multiclass labeling, several class combinations are unnatural, such as FUTURE and S/O. We list up possible label combinations (that have at least one occurrence in the corpora); if such a combination appears in a text, we adapt a high confidence label (using a marginal distance).

5 Experiments

We investigate what kind of information contributes to the performance in various machine learning algorithms.

Table 5: Classification of Modalities

NEGATION	An event with negation words such as "not" or "no".
FUTURE	An event that is scheduled for execution in the future.
PURPOSE	An event that is planed by a doctor, but its time schedule is ambiguous (just a hope/intention).
S/O	An event (usually a disease) that is suspected. For example, given "suspected microscopic tumor in ...", "microscopic tumor" is an S/O event."
NECESSITY	An event (usually a remedy or medical test) that is required.
INTEND	An event that is hoped for by a patient. Note that if the event is hoped by a doctor, we regard is a PURPOSE or FUTURE. For example, given "He hoped for chemical therapy", "chemical therapy" is INTEND.
POSSIBLE	An event (usually remedy) that is possible under the current situation.
RECOMMEND	An event (usually remedy) that is recommended by other doctor(s).

5.1 Corpus and Setting

We collected 435 Japanese discharge summaries in which events and the modality are annotated. For training, we used the CRF toolkit[8] with standard parameters. In this experiment setting, the input is an event with its contexts. The output is an event modality class (positive of negative in two-way) (or more detailed modality class in nine-way).

The core problem addressed in this paper is modality classification. Therefore, this task setting assumes that all events are identified correctly. Table 6 presents the event identification accuracy. Except for the rare class V (the other verb), we got more than 80% *F*-scores. It is true that the accuracy is not perfect. Nevertheless, most of the remaining problems in this step will be solved using a larger corpus.

5.2 Comparable Methods

We conducted experiments in the 10-fold cross validation manner. We investigated the perform-

[8] http://crfpp.sourceforge.net/

ance in various feature combinations and the following machine learning methods.

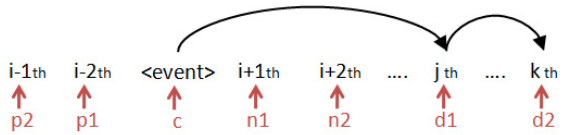


Figure 3: Features

Table 6: Event Identification Result. Tag precision recall *F*-score.

	#	P	R	F
A (ACTION)	1,556	94.63	91.04	92.80
V (VERB)	1,047	84.64	74.89	79.47
D (DISEASE)	3,601	85.56	80.24	82.82
M (MEDICINE)	1,045	86.99	81.34	84.07
R (REMEDY)	1,699	84.50	76.36	80.22
T (TEST)	2,077	84.74	76.68	80.51
ALL	11,025	84.74	76.68	80.51

Table 7: Various Machine Learning Method

SVM	Support Vector Machine (Vapnik, 1999). We used TinySVM[9] with a polynomial kernel (degree=2).
AP	Averaged Perceptron (Collins, 2002)
PA1	Passive Aggressive I (Crammer et al., 2006)*
PA2	Passive Aggressive II (Crammer et al., 2006)*
CW	Confidence Weighted (Dredze et al., 2008)*

* The online learning library[10] is used for AP PA1,2 CW .

5.3 Evaluation Metrics

We adopt evaluation of two types:

(1) Two-way: positive or negative:

(2) Nine-way: positive or one of eight modality categories.

Recall and *F*-measure are investigated in both for evaluation precision.

5.4 Results

The results are shown in Table 8 (Two-Way) and in Table 9 (Nine-Way).

Current Event Category

The results in ID0–ID1 indicate that the current event category (CAT) is useful. However, events are sometimes misestimated in real settings. We must check more practical performance in the future.

Bag-of-words (BOW) Information

Results in ID1–ID2 indicate that BOW is important.

Surrounding Phrase Contribution

The results appearing in ID2–ID9 represent the contribution of each feature position. From ID3, ID4, and ID7 results, next phrases (n1, n2) and parent phrases (d1) were able to boost the accuracy. Despite the risk of parsing errors, parent phrases (d1) are helpful, which is an insight of this study.

In contrast, we can say that the following features had little contribution: previous phrases (p1, p2 from ID5 and ID6), grandparent phrases (d2 from ID8), and previous events (e from ID9).

Regarding p1 and p2, these modalities are rarely expressed in the previous parts in Japanese.

As for d2, the grandparent phrases might be too removed from the target events.

As for e, because texts in health records are fragmented, each event might have little relation.

However, the above features are also helpful in cases with a stronger learning algorithm.

In fact, among ID10–ID14, the SVM-based classifier achieved the best accuracy with all features (ID14).

Table 8: Two-way Results

ID	Learning Algorithm	CAT	BOW	P2	P1	c	n1	n2	d1	d2	e	Precision	Recall	F-measure
0	CW					●						61.25	50.10	54.25
1	CW	●				●						63.73	51.18	56.76
2	CW	●	●			●						78.55	70.21	74.14
3	CW	●	●			●	●					84.40	78.08	81.11
4	CW	●	●			●	●	●				**85.71**	77.72	81.51
5	CW	●	●		●	●	●	●				84.61	75.53	79.81
6	CW	●	●	●	●	●	●	●				85.34	71.85	78.01
7	CW	●	●			●	●	●	●			85.48	**78.22**	**81.68**
8	CW	●	●			●	●	●	●	●		83.94	78.08	80.80
9	CW	●	●			●	●	●	●		●	83.20	78.04	80.53
10	SVM	●	●			●	●	●				88.57	80.84	84.52
11	SVM	●	●			●	●	●	●			89.24	81.05	84.94
12	SVM	●	●			●	●	●	●	●		89.29	81.54	85.23
13	SVM	●	●			●	●	●	●		●	89.30	81.54	85.24
14	SVM	●	●	●	●	●	●	●	●	●	●	**89.40**	**82.50**	**85.81**
15	AP	●	●			●	●	●	●			92.28	64.56	75.97
16	PA1	●	●			●	●	●	●			93.08	68.03	78.60
17	PA2	●	●			●	●	●	●			92.99	67.67	78.33

● indicates the used feature. c are features from the current phrase. p1, p2, n1, n2 are features from surrounding phrases. e are features from a previous event. **BOW** is a bag-of-words using features from an entire sentence. **CAT** is the category of the current event.

[9] http://chasen.org/ taku/software/TinySVM/

[10] http://code.google.com/p/oll

Learning Methods
Regarding the learning algorithms, all online learning methods (ID7 and ID15–17) showed lower accuracies than SVM (ID11), indicating that this task requires heavy learning.

Nine-way Results
Table 9 presents the accuracies of each class. Fundamentally, we can obtain high performance in the frequent classes (such as NEGATION, PURPOSE, and S/O). In contrast, the classifier suffers from low frequent classes (such as FUTURE). How to handle such examples is a subject of future study.

Table 9: Two-way Results

	#	Precision	Recall	*F*-measure
NEGATION	441	84.19	77.36	80.63
PURPOSE	346	91.35	63.87	75.17
S/O	242	90.74	72.39	80.53
FUTURE	97	23.31	55.96	32.91
POSSIBLE	36	83.33	40.55	54.55
INTEND	32	76.66	29.35	42.44
RECOMMEND	21	95.71	38.57	54.98
NECESSITY	4	100	0	0

4.5 Future Works
In this section, we will discuss several remaining problems. First, as described, the classifier suffers from low frequent modality classes. To give more examples for such classes is an important problem.
Our final goal is to realize precise information extraction from health records. Our IE systems are already available at the web site (http://lab0.com). Comprehensive evaluation of those systems is required.

6 Conclusions

This paper presented a classifier that identified whether an event has actually occurred or not. The proposed SVM-based classifier uses both BOW information and dependency parsing results. The experimental results demonstrated 85.8 *F*-measure% accuracy and revealed that syntactic information can contribute to the method's accuracy. In the future, a method of handling low-frequency events is strongly desired.

Acknowledgments

Part of this research is supported by Grant-in-Aid for Scientific Research (A) of Japan Society for the Promotion of Science Project Number: 20680006 F.Y.2008-20011 and the Research Collaboration Project with Fuji Xerox Co. Ltd.

References

Wendy Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce Buchanan. 2001a. Evaluation of negation phrases in narrative clinical reports. In Proceedings of AMIA Symp, pages 105-109.

Wendy Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce Buchanan. 2001b. A simple algorithm for identifying negated findings and diseases in discharge summaries. Journal of Biomedical Informatics, 5:301-310.

Wendy Chapman, John Dowling and David Chu. 2007. ConText: An algorithm for identifying contextual features from clinical text. Biological, translational, and clinical language processing (BioNLP2007), pp. 81–88.

Eiji Aramaki, Takeshi Imai, Kengo Miyo, and Kazuhiko Ohe: Orthographic Disambiguation Incorporating Transliterated Probability International Joint Conference on Natural Language Processing (IJCNLP2008), pp.48-55, 2008.

Peter L. Elkin, Steven H. Brown, Brent A. Bauer, Casey S. Husser, William Carruth, Larry R. Bergstrom, and Dietlind L. Wahner Roedler. A controlled trial of automated classification of negation from clinical notes. BMC Medical Informatics and Decision Making 5:13.

C. Friedman, P.O. Alderson, J.H. Austin, J.J. Cimino, and S.B. Johnson. 1994. A general natural language text processor for clinical radiology. Journal of the American Medical Informatics Association, 1(2):161-174.

L. Gillick and S.J. Cox. 1989. Some statistical issues in the comparison of speech recognition algorithms. In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pages 532-535.

Ilya M. Goldin and Wendy Chapman. 2003. Learning to detect negation with not in medical texts. In Workshop at the 26th ACM SIGIR Conference.

Yang Huang and Henry J. Lowe. 2007. A novel hybrid approach to automated negation detection in clinical radiology reports. Journal of the American Medical Informatics Association, 14(3):304-311.

Kentaro Inui, Shuya Abe, Hiraku Morita, Megumi Eguchi, Asuka Sumida, Chitose Sao, Kazuo Hara, Koji Murakami, and Suguru Matsuyoshi. 2008. Experience mining: Building a large-scale database of personal experiences and opinions from web documents. In Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence, pages 314-321.

M. Ito, H. Imura, and H. Takahisa. 2003. Igaku- Shoin's Medical Dictionary. Igakusyoin.

Sadao Kurohashi and Makoto Nagao. 1994. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. Computational Linguistics, 20(4).

Pradeep G. Mutalik, Aniruddha Deshpande, and Prakash M. Nadkarni. 2001. Use of general purpose negation detection to augment concept indexing of medical documents: A quantitative study using the umls. Journal of the American Medical Informatics Association, 8(6):598-609.

J. Lafferty, A. McCallum, and F. Pereira: Conditional random fields: Probabilistic models for segmenting and labeling sequence data, In Proceedings of the International Conference on Machine Learning (ICML2001), pp.282-289, 2001.

R. Prasad, N. Dinesh, A. Lee, A. Joshi and B. Webber: Annotating Attribution in the Penn Discourse TreeBank, In Proceedings of the International Conference on Computational Linguistics and the Annual Conference of the Association for Computational Linguistics (COLING/ACL2006) Workshop on Sentiment and Subjectivity in Text, pp.31-38 (2006).

R. Saurí, and J. Pustejovsky: Determining Modality and Factuality for Text Entailment, Proceedings of ICSC2007, pp. 509-516 (2007).

Gaizauskas, A. Setzer, G. Katz, and D.R. Radev. 2003. New Directions in Question Answering: Timeml: Robust specification of event and temporal expressions in text. AAAI Press.

SNOMED-CT. 2002. SNOMED Clinical Terms Guide. College of American Pathologists.

Sibanda Tawanda, Tian He, Peter Szolovits, and Uzuner Ozlem. 2006. Syntactically informed semantic category recognizer for discharge summaries. In Proceedings of the Fall Symposium of the American Medical Informatics Association (AMIA 2006), pages 11-15.

Sibanda Tawanda and Uzuner Ozlem. 2006. Role of local context in automatic deidentification of ungrammatical, fragmented text. In Proceedings of the Human Language Technology conference and the North American chapter of the Association for Computational Linguistics (HLT-NAACL2006), pages 65-73.

Veronika Vincze, Gyorgy Szarvas, Richard Farkas, Gyorgy Mora, and Janos Csirik. 2008. The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. BMC Bioinformatics, 9(11).

Semantic Annotation of Papers: Interface & Enrichment Tool (SAPIENT)

Maria Liakata[†], **Claire Q**[††], **Larisa N. Soldatova**[†††]
Department of Computer Science
University of Wales, Aberystwyth
SY23 3DB UK
†mal@aber.ac.uk, ††ceq08@aber.ac.uk, †††lss@aber.ac.uk

Abstract

In this paper we introduce a web application (SAPIENT) for sentence based annotation of full papers with semantic information. SAPIENT enables experts to annotate scientific papers sentence by sentence and also to link related sentences together, thus forming spans of interesting regions, which can facilitate text mining applications. As part of the system, we developed an XML-aware sentence splitter (SSSplit) which preserves XML markup and identifies sentences through the addition of in-line markup. SAPIENT has been used in a systematic study for the annotation of scientific papers with concepts representing the Core Information about Scientific Papers (CISP) to create a corpus of 225 annotated papers.

1 Introduction

Given the rapid growth in the quantity of scientific literature, particularly in the Biosciences, there is an increasing need to work with full papers rather than abstracts, both to identify their key contributions and to provide some automated assistance to researchers (Karamanis et al., 2008; Medlock and Briscoe, 2007). Initiatives like OTMI[1], which aim to make full papers available to researchers for text mining purposes is further evidence that relying solely on abstracts presents important limitations for such tasks. A recent study on whether information retrieval from full text is more effective than searching abstracts alone (Lin Jimmy, 2009) showed that the former is indeed the case. Their experimental results suggested that span-level analysis is a promising strategy for taking advantage of the full papers, where spans are defined as paragraphs of text assessed by humans and deemed to be relevant to one of 36 pre-defined topics. Therefore, when working with full papers, it is important to be able to identify and annotate spans of text. In previous research, sentence based annotation has been used to identify text regions with scientific content of interest to the user (Wilbur et al., 2006; Shatkay et al., 2008) or zones of different rhetorical status (AZ) (Teufel and Moens, 2002). Sentences are the structural units of paragraphs and can be more flexible than paragraphs for text mining purposes other than information retrieval.

Current general purpose systems for linguistic annotation such as Callisto[2] allow the creation of a simple annotation schema that is a tag set augmented with simple (e.g. string) attributes for each tag. Knowtator (Ogren, 2006) is a plug-in of the knowledge representation tool Protégé[3], which works as a general purpose text annotation tool and has the advantage that it can work with complex ontology-derived schemas. However, these systems are not particularly suited to sentence by sentence annotation of full papers, as one would need to highlight entire sentences manually. Also these systems work mainly with plain text, so they do not necessarily interpret the structural information already available in the paper, which can be crucial to annotation decisions for the type of high level annotation mentioned

[1] http://opentextmining.org/wiki/Main_Page

[2] http://callisto.mitre.org/manual/use.html

[3] http://protege.stanford.edu/

Proceedings of the Workshop on BioNLP, pages 193–200,
Boulder, Colorado, June 2009.

above. The OSCAR3 (Corbett et al., 2007) tool for the recognition and annotation of chemical named entities fully displays underlying paper information in XML but is not suited to sentence by sentence annotation.

To address the above issues, we present a system (SAPIENT) for sentence by sentence annotation of scientific papers which supports ontology-motivated concepts representing the core information about scientific papers (CISP) (Soldatova and Liakata, 2007). An important aspect of the system is that although annotation is sentence based, the system caters for identifiers, which link together sentences pertaining to the same concept. This way spans of interest or key regions are formed. SAPIENT also incorporates OSCAR3 capability for the automatic recognition of chemical named entities and runs within a browser, which makes it platform independent. SAPIENT takes as input full scientific papers in XML, splits them into individual sentences, displays them and allows the user to annotate each sentence with one of 11 CISP concepts as well as link the sentence to other sentences referring to the same instance of the concept selected. The system is especially suitable for so called multi-dimensional annotation (Shatkay et al., 2008) or ontology-motivated annotation, where a label originates from a class with properties. SAPIENT is currently being employed by 16 Chemistry experts to develop a corpus of scientific papers (ART Corpus) annotated with Core Information about Scientific Papers (CISP) covering topics in Physical Chemistry and Biochemistry.

2 SAPIENT System Description

We chose to implement SAPIENT as a web application, so as to make it platform independent and easier to incorporate as part of an online workflow. We have used state of the art web technologies to develop SAPIENT, namely Java, Javascript (with Asynchronous JavaScript and XML (AJAX) functionality), XSLT, CSS and XML. The system has a client-server architecture (see Figure 1), with papers being uploaded and stored on the server but functionality for annotation contained in Javascript, which runs client-side in the browser. This is inspired by but in contrast with OSCAR3 (Corbett et al., 2007), which also allows manual annotation alongside the automated annotation of chemical named entities, but where each minor edit is saved to the server, writing to a file. We chose to make more of the functionality client-side in order to reduce the number of server requests, which could become problematic if the system became widely distributed.

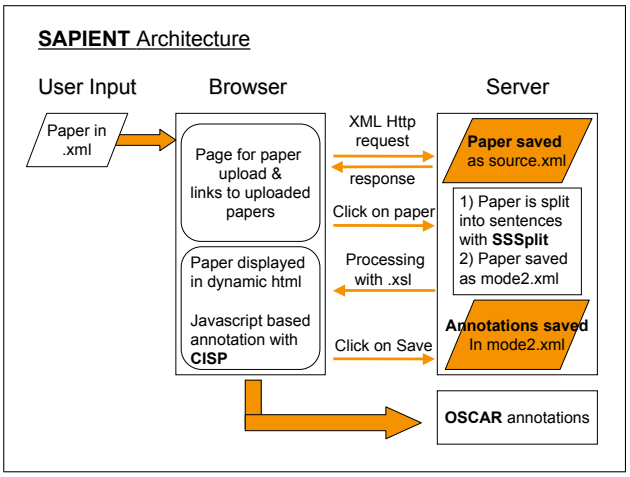


Figure 1: Architecture of the SAPIENT System

SAPIENT has been designed to take as input full papers in XML, conforming to the SciXML schema (Rupp et al., 2006)(see Section 3).

To view or annotate a paper, a user must first upload it. The index page of SAPIENT shows a list of papers already uploaded (available as links) and an interface for uploading more papers (See Figure 2). Once the user selects a link to a paper, the paper is split into sentences using the XML-aware sentence splitter SSSplit which we have developed (See section 4) and is included in the server-side Java. The resultant XML file is stored alongside the original upload. Sentence splitting involves detecting the boundaries of sentences and, in this context, marking the latter by inline <s></s> tags added to the original XML. The <s></s> tags contain an id attribute enumerating the sentence.

After sentence splitting, the new XML file containing sentence boundaries marked by <s id=#NUM>< /s> tags is parsed by XSLT into HTML, so that it displays in the browser. In the HTML interface dynamically generated in this way, Javascript annotation drop-downs are available for

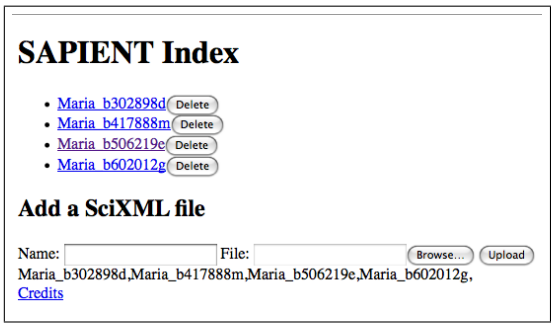


Figure 2: Index page of the SAPIENT System

each sentence. The user can perform annotations by selecting items from the drop-downs and all the corresponding annotation information is stored in Javascript until a request to save is made by the user.

The Javascript drop-downs allow annotation at two levels (Figure 3), enabling a sentence to have a semantic label (type) with properties (subtypes) and an identifier (conceptID).

In the current implementation of SAPIENT, The **type** drop-down value corresponds to the selection of one out of 11 general scientific concepts (Liakata and Soldatova, 2008), namely ('Background', 'Conclusion', 'Experiment', 'Goal of the Investigation', 'Hypothesis','Method', 'Model', 'Motivation', 'Object of the Investigation', 'Observation', 'Result'). These labels originate from a set of meta-data (The Core Information about Scientific Concepts (CISP) (Soldatova and Liakata, 2007) which were constructed using an ontology methodology, based on an ontology of experiments EXPO (Soldatova and King, 2006). Because these labels map to ontology classes, they can also have properties. For example, 'Method' has the property 'New'/'Old','Advantage'/'Disadvantage'. These properties are dependent on the type selected and are expressed in terms of the **subtype** drop-down. The third drop-down, **concept ID** allows a user to provide a **concept identifier**. The latter is an entity formed by the name of a concept and a number (e.g. "Res2"). Concept identifiers uniquely identify an instance of a concept (e.g. the second Result), but not a sentence. That is, concept identifiers designate and link together instances of the same semantic concept, spread across different sentences, which can be in different parts of the paper. For example, the second result ("Res2") can be referred to by 1 sentence in the abstract, 5 sentences in the Discussion and 2 sentences in the Conclusion sections.

The distinction between sentence identifiers and concept identifiers is an important characteristic of the system. It means that the system does not necessarily assume a '1-1' correspondence between a sentence and a concept, but rather that concepts can be represented by spans of often disjoint text. Therefore, SAPIENT indirectly allows the annotation of discourse segments beyond the sentence level and also keeps track of co-referring sentences.

2.1 SAPIENT Usability

Even though SAPIENT has been primarily designed to work with CISP concepts, it can be used to annotate papers according to any sentence based annotation scheme. Changes required can be easily performed by modifying the XSL sheet which dynamically generates HTML from XML and organises the structure of drop-down menus. Automated noun-phrase based annotation from existing ontologies is available to SAPIENT users through OSCAR3 (Corbett et al., 2007), since SAPIENT incorporates OSCAR3 functionality for chemical named entity recognition. The latter is implemented as a link which when selected calls the OSCAR3 workflow (integrated in the system) to automatically recognise chemical named entities (NEs) (See Figure 5).

When all annotations (both sentence based and chemical NEs) are saved to the server, a new version of the XML file is produced, which contains in-line annotation for sentences as well as extra in-line annotation for the semantic concepts and NEs embedded within <s></s> tags. These annotation tags are compliant with the SciXML schema (Rupp et al., 2006) and in the case of sentence-based annotations are of the form:

```
<annotationART atype=''GSC''
               type=#TYPE
               conceptID=#CONCEPTID
               novelty=''Yes/No''
               advantage=''Yes/No''
</annotationART>
```

(See Figure 4). The attribute **type**, stands for the CISP concept selected for the sentence in question. The **conceptID** attribute is an enumerator of the particular concept, which the sentence refers to. For

example, two different sentences will have different sentence ids but if they refer to the same concept (e.g. the same "Conclusion") , they will be assigned the same concept ID (e.g. "Con3"). The attributes **novelty** and **advantage**, are properties of the concepts assigned to a sentence and depend on the concept selection. They take boolean values or the dummy value "None" if the properties are not defined for a particular concept. For example, these attributes are relevant when the concept selected is a 'Method', in which case the method can be "New/Old" and/or have an "Advantage/Disadvantage". The novelty and advantage attributes co-exist in the annotation (as can be seen in Figure 4) but they are not set by the system at the same time. For instance, if a sentence refers to a new method, it will be given the type 'Method' and the subtype "New"; this sets the novelty attribute in the underlying XML to "Yes" and leaves the advantage attribute set to the default "None". The sentence will also be given a conceptID, e.g. "Met1". If another sentence refers to an advantage of this method, then the new sentence will be assigned the type 'Method', the subtype "Advantage" (which sets the underlying advantage attribute to "Yes") and the same conceptID "Met1". The novelty attribute value is then inherited from the novelty attribute value of the first coreferring sentence, which in this case is "New".

3 Input: Paper in XML

SAPIENT currently accepts as input papers in XML, especially ones compliant with the SciXML schema (Rupp et al., 2006). SciXML is ideally suited for this purpose as it was developed for representing the logical structure of scientific research papers. Tags used in the schema serve the purpose of paper identification (e.g. <TITLE>,<AUTHOR>), defining sections of the paper (e.g. <DIV>,<HEADER>), text sections with specific function and formatting (e.g. <ABSTRACT>, <EQUATION>), paragraph tags <P>, references, tables, figures and footnotes, lists, bibliography. SAPIENT operates only on the <TITLE>, <ABSTRACT> ,<BODY> and <P> tags, leaving out any list elements following the body, such as acknowledgements, figures or references at the end of the paper. This is because we make the assumption that only the abstract and the body contain sentences with semantic content of any importance to the research carried out in the paper. This would have been different if SAPIENT annotated figures as well, but such provision is not currently made. Tags such as <REF>, citations in the text, are included within the sentence boundaries.

Even though SAPIENT was developed with the SciXML schema in mind, it will work with any well formed XML document that has <PAPER> as the root node and which also contains an <ABSTRACT> and <BODY> node. Therefore, it is relatively easy to adapt SAPIENT to other XML schemas.

4 SSSplit: Sapient Sentence Splitting

4.1 Sentence Matching

The reason for developing our own sentence splitter was that sentence splitters widely available could not handle XML properly. The XML markup contains useful information about the document structure and formatting in the form of inline tags, which is important for determining the logical structure of the paper. The latter is worth preserving for our purposes, since it can influence the annotation of individual sentences. XML markup (e.g. <ABSTRACT>,<REF>,<EQUATION>) needs to be combined carefully with tags designating sentence boundaries (<s></s>), so that the resulting document is in well formed XML. Current sentence splitters ignore XML markup, which means that any document formatting/information would have to be removed in order to use them. RASP (Briscoe et al., 2006), the sentence splitter used in the Sciborg project[4] at the University of Cambridge, can deal with XML but has to be compiled for different operating systems, which would result in compromising the platform independence of SAPIENT. A recent MPhil thesis (Owusu, 2008) has also developed an XML-aware sentence splitter but the code is in Microsoft C#.Net and therefore not platform independent.

We have written the XML-aware sentence splitter SSSplit in the platform-independent Java language (version 1.6), based on and extending open source Perl code[5] for handling plain text. In or-

[4] http://www.cl.cam.ac.uk/research/nl/sciborg/www/

[5] http://search.cpan.org/ tgrose/HTML-Summary-0.017/

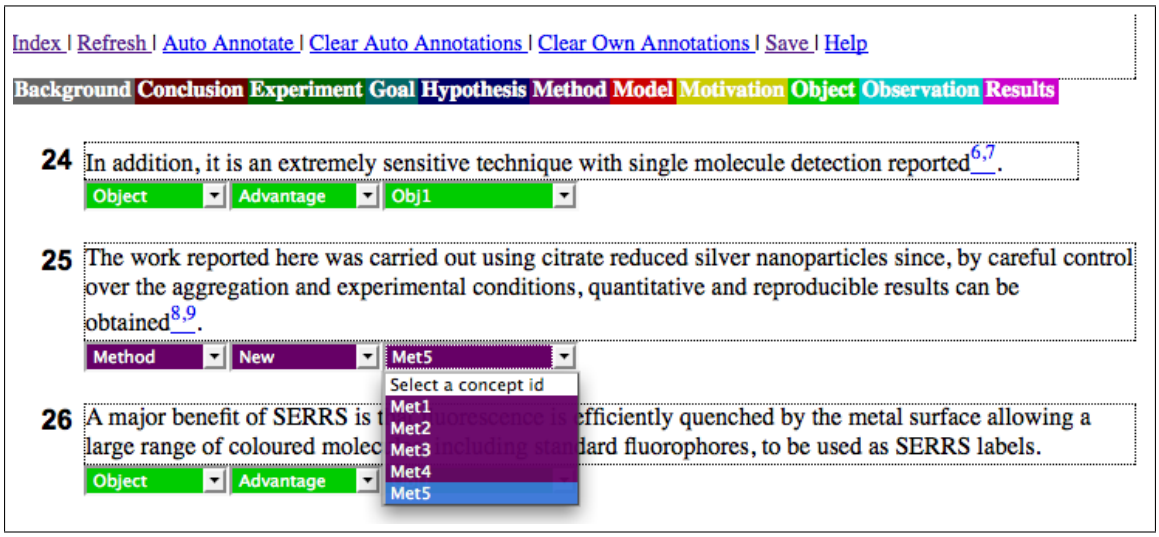


Figure 3: Example of SAPIENT annotation through selection from drop-down menu.

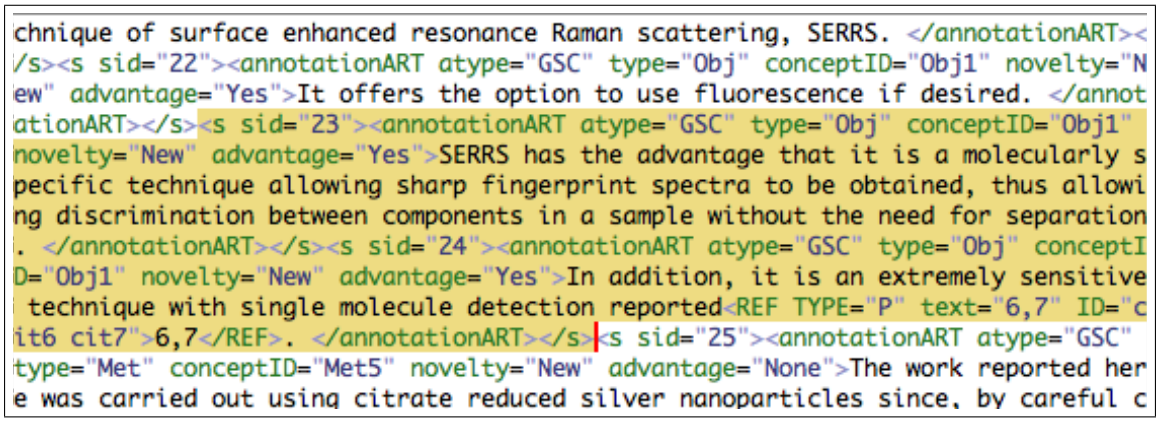


Figure 4: Behind the scenes: Example XML fragment of a paper annotated using SAPIENT.

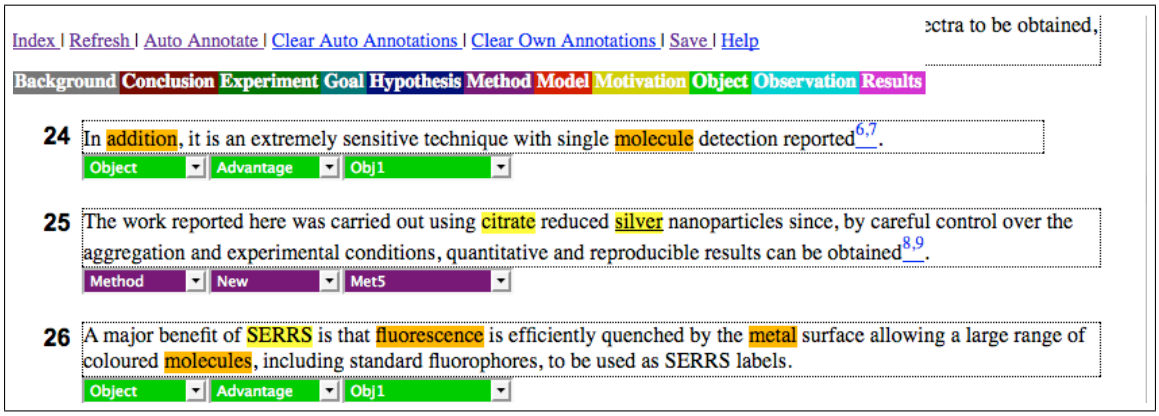


Figure 5: Incorporation of OSCAR3 annotations in SAPIENT, after selecting the link “Auto Annotate”

der to make our sentence splitter XML aware, we translated the Perl regular expression rules into Java and modifed them to make them compatible with the SciXML(Rupp et al., 2006) schema. We then further improved the rules, by training on a set of 14 papers in SciXML. This involved displaying the papers, checking whether the XML was well formed and making corrections accordingly. We would observe cases of oversplit and undersplit sentences and amend the rules while keeping them as general as possible. The rules in SSSplit were evaluated by comparing the system output against a gold standard of 41 papers, where sentence boundaries had been provided by human experts (See section 4.2). The sentence splitter is integrated within the SAPIENT system but is also available as a separate package ("SSSplit"). This should enable any future work to easily incorporate or extend it. It is currently trained for splitting papers in SciXML, but can be easily ported to any other kind of XML, as discussed in section 3.

4.2 SSSplit Evaluation

SAPIENT and SSSplit have been have been employed by more than 20 different users to successfully display 270 full papers. For a more accurate evaluation of the quality of the sentences produced by SSSplit, we used a Perl script which compared the sentence boundaries (start and end) generated by SSSplit, to sentence tags in a set of 41 papers (SciXML files) annotated manually by human experts. If both the start and end of a sentence matched up in the generated and manual versions, we considered this a true positive result. In the case where a sentence did not match in the two versions, we first searched for a matching end in our generated set of sentences and then in the hand annotated version. If the 'true' end of the sentence (as defined by the manual annotation) was found in later sentences in the SSSplit version, this meant that the system had split a sentence too early, or "oversplit". This we considered to be a false positive, since we had detected a sentence boundary where in reality there was none. This would result in the following sentence being matched at the end only, which also counts as a false positive. In the case where the end of the SSSplit sentence was found in a later sentence, within the set of 'true' sentences, it meant that our sentence spanned too wide, or that the system had "undersplit". These cases we considered to be false negatives, as we had failed to detect a sentence boundary where there was one.

	RASP	Owusu	SSSplit
Precision	0.994	0.996	0.964
Recall	0.983	0.990	0.994
F-measure	0.988	0.992	0.978

Table 1: Comparison of sentence splitters in RASP, Owusu and SSSplit.

Our training consisted of 14 papers in the fields of physical chemistry and biochemistry. A different set of 41 papers distinct from the training set but from the same thematic domain was used as a test set. Out of these 41 papers, 36 feature as a test set (with n-fold validation) also for the sentence splitters RASP (Briscoe et al., 2006) and the XML-aware sentence splitter developed by (Owusu, 2008). The results for all three systems, obtained as medians of Precision, Recall and F-measure for the 36 papers are shown in Table 1.

Precision is the proportion of true positives over all end and start tags returned, giving a measure of the number of boundaries identified correctly. Recall is the proportion of true positives over all the relevant start and end tags in the hand-annotated papers, giving a measure of the number of boundaries actually found. F-Measure combines Precision and Recall to give a more balanced view on the system performance.

In comparison with RASP and the XML-Aware splitter of (Owusu, 2008), SSSplit performed well, though it did not outperform these systems. Their highest result for precision was 0.996 (vs 0.964 for SSSplit) and for recall 0.990 (vs 0.994 for SSSplit). We can explain their higher results somewhat by their use of n-fold cross-validation on 36 out of the same 41 papers that we used, which can allow information from the test set to leak into the training data. We did not perform n-fold cross-validation, as this would have involved going through each of the papers and removing any potential influence on our regular expression rules of the sentences included within, which is a non-trivial process. Our test data was completely unseen, which meant that our eval-

	Training (1979 sentences)	Testing (5002 sentences)
Precision	0.961	0.964
Recall	0.995	0.994
F-measure	0.96875	0.978

Table 2: Comparison of SSSplit on the training and testing papers. The training set consisted of 14 papers (1979 sentences) and the testing set of 41 papers (5002 sentences).

uation is stricter, avoiding any influence from the training data.

In addition to the comparison between SSSplit and the other two XML-aware sentence splitters, we also performed a comparison between our training and testing sets, depicted in Table 2.

As can be seen in Table 2, recall was only slightly better on the training set than the test set, but precision was worse on the training set, presumably because of lack of attention being paid to the oversplitting in a particular paper ("b103844n"). This shows that we have not overfitted to the training set in developing our splitter. Our recall is particularly high, indicating that our splitter makes very few false negative errors. We can attribute many of the false positive errors to our somewhat small set of abbreviations considered, resulting in oversplit sentences. We would like to incorporate a more sophisticated approach to abbreviations in the future.

5 Performing CISP Annotations

Within the context of the ART project (Soldatova et al., 2007), SAPIENT has been used by 16 Chemistry experts to annotate 265 papers from RSC Publishing journals, covering topics in Physical Chemistry and Biochemistry. Experts have been annotating the papers sentence by sentence, assigning each sentence one of 11 core scientific concepts and linking together sentences across a paper which refer to the same instance of a concept. The aim is to create a corpus of annotated papers (ART-corpus) with regions of scientific interest identified by CISP concepts ("Result","Conclusion", "Observation","Method" and so on).

A preliminary evaluation of the experts' agreement on the ART Corpus, based on a sample of 41 papers, annotated by the 16 experts in non-overlapping groups of 3, shows significant agreement between annotators, given the difficulty of the task (an average kappa co-efficient of 0.55 per group). The details of this work are beyond the scope of the current paper, but the preliminary results underline the usability of both the CISP meta-data and SAPIENT. In the future, we plan to further evaluate the ART Corpus by incorporating existing machine learning algorithms into SAPIENT and automating the generation of CISP meta-data. This would make SAPIENT a very useful tool and would indeed add a lot more value to the meta-data, since training and paying annotators is a costly process and manually annotating papers is incredibly time consuming.

6 Conclusion and Future Work

We have presented SAPIENT, a web-based tool for the annotation of full papers, sentence by sentence, with semantic information. We have also discussed how these annotations result in the indirect definition of regions of interest within the paper. The system has been already tested in a systematic study and has been employed for the creation of a corpus of papers annotated with CISP concepts (ART Corpus). In the future we plan to extend SAPIENT so that the system can itself suggest annotation labels to users. We also plan to target the needs of particular users such as authors of papers, reviewers and editors.

SAPIENT, SSSplit and their documentation are both available for download from http://www.aber.ac.uk/compsci/Research/bio/art/sapient/.

Acknowledgments

We would like to thank Peter Corbett, Amanda Clare, Jem Rowland and Andrew Sparkes for reading and commenting on earlier versions of this paper. We would also like to thank the anonymous reviewers for their useful comments. This work was part of the ART Project (http://www.aber.ac.uk/compsci/Research/bio/art/), funded by the U.K. Higher Education Joint Information Services Committee (JISC).

References

E. Briscoe, J. Carroll and R. Watson 2006. The Second Release of the RASP System. *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, Sydney, Australia.*

P. Corbett, P. Batchelor and S. Teufel. 2007. Annotation of Chemical Named Entities. *Proc. BioNLP*.

Nikiforos Karamanis, Ruth Seal, Ian Lewin, Peter McQuilton, Andreas Vlachos, Caroline Gasperin, Rachel Drysdale and Ted Briscoe. 2008. Natural Language Processing in aid of FlyBase curators. *BMC Bioinformatics*, 9:193.

Maria Liakata and Larisa N. Soldatova. 2008. Guidelines for the annotation of General Scientific Concepts *JISC Project Report*, http://ie-repository.jisc.ac.uk/.

Jimmy Lin 2009. Is Searching Full Text More Effective Than Searching Abstracts? *BMC Bioinformatics*, 10:46.

Ben Medlock and Ted Briscoe. 2007. Weakly supervised learning for hedge classification in scientific literature. *45th Annual Meeting of the Association for Computational Linguistics*, 23-30 Jun 2007, Prague, Czech Republic.

P. Ogren. 2006. Knowtator: a Protégé plug-in for annotated corpus construction. *Proceedings of the 2006 Conference of the North American Chapter of the Association For Computational Linguistics on Human Language Technology: Companion Volume: Demonstrations*, New York Press, New York, June 04 - 09, 2006.

Lawrence Owusu. 2008. XML-Aware Sentence Splitter. *MPhil thesis*, Cambridge, UK.

CJ Rupp, Ann Copestake, Simone Teufel and Ben Waldron. 2006. Flexible Interfaces in the Application of Language Technology to an eScience Corpus. *Proceedings of the UK e-Science Programme All Hands Meeting 2006 (AHM2006), Nottingham, UK*

Hagit Shatkay, Fengxia Pan, Andrey Rzhetsky and W. John Wilbur. 2008. Multi-dimensional classification of biomedical text: Toward automated, practical provision of high-utility text to diverse users. *Bioinformatics*, 24(18):2086–2093.

Larisa N. Soldatova and Maria Liakata. 2007. An ontology methodology and CISP - the proposed Core Information about Scientific Papers. *JISC Project Report*, http://ie-repository.jisc.ac.uk/137/.

L. Soldatova, C. Batchelor, M. Liakata, H. Fielding, S. Lewis and R. King 2007. ART: An ontology based tool for the translation of papers into Semantic Web format. *Proceedings of the SIG/ISMB07 ontology workshop.*, p.33–36.

Larisa N. Soldatova and Ross D. King. 2006. An Ontology of Scientific Experiments. *Journal of the Royal Society Interface*, 3:795–803.

S. Teufel and M. Moens. 2002. Summarizing Scientific Articles – Experiments with Relevance and Rhetorical Status. *Computational Linguistics*, 28(4). (preprint)

W. Wilbur, A. Rzhetsky and H. Shatkay. 2006. New Directions in Biomedical Text Annotations: Deifinitions, Guidelines and Corpus Construction. *BMC Bioinformatics*, 7:356.

Author Index