Effectively Using Syntax for Recognizing False Entailment

Rion Snow
Computer Science Department
Stanford University
Stanford, CA 94305
rion@cs.stanford.edu

Lucy Vanderwende and Arul Menezes
Microsoft Research
One Microsoft Way
Redmond, WA 98027
{lucyv,arulm}@microsoft.com

Abstract

Recognizing textual entailment is a challenging problem and a fundamental component of many applications in natural language processing. We present a novel framework for recognizing textual entailment that focuses on the use of syntactic heuristics to recognize false entailment. We give a thorough analysis of our system, which demonstrates state-of-the-art performance on a widely-used test set.

1 Introduction

Recognizing the semantic equivalence of two fragments of text is a fundamental component of many applications in natural language processing. Recognizing textual entailment, as formulated in the recent PASCAL Challenge [1], is the problem of determining whether some *text sentence* T entails some *hypothesis sentence* H.

The motivation for this formulation was to isolate and evaluate the application-independent component of semantic inference shared across many application areas, reflected in the division of the PASCAL RTE dataset into seven distinct tasks: Information Extraction (IE), Comparable Documents (CD), Reading Comprehension (RC), Machine Translation (MT), Information Retrieval (IR), Question Answering (QA), and Paraphrase Acquisition (PP).

[1] http://www.pascal-network.org/Challenges/RTE. The examples given throughout this paper are from the first PASCAL RTE dataset, described in Section 6.

The RTE problem as presented in the PASCAL RTE dataset is particularly attractive in that it is a reasonably simple task for human annotators with high inter-annotator agreement (95.1% in one independent labeling (Bos and Markert, 2005)), but an extremely challenging task for automated systems. The highest accuracy systems on the RTE test set are still much closer in performance to a random baseline accuracy of 50% than to the inter-annotator agreement. For example, two high-accuracy systems are those described in (Tatu and Moldovan, 2005), achieving 60.4% accuracy with no task-specific information, and (Bos and Markert, 2005), which achieves 61.2% *task-dependent* accuracy, i.e. when able to use the specific task labels as input.

Previous systems for RTE have attempted a wide variety of strategies. Many previous approaches have used a logical form representation of the text and hypothesis sentences, focusing on deriving a proof by which one can infer the hypothesis logical form from the text logical form (Bayer et al., 2005; Bos and Markert, 2005; Raina et al., 2005; Tatu and Moldovan, 2005). These papers often cite that a major obstacle to accurate theorem proving for the task of textual entailment is the lack of world knowledge, which is frequently difficult and costly to obtain and encode. Attempts have been made to remedy this deficit through various techniques, including model-building (Bos and Markert, 2005) and the addition of semantic axioms (Tatu and Moldovan, 2005).

Our system diverges from previous approaches most strongly by focusing upon false entailments; rather than assuming that a given entailment is false until proven true, we make the opposite assump-

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 33–40,
New York, June 2006.

tion, and instead focus on applying knowledge-free heuristics that can act locally on a subgraph of syntactic dependencies to determine with high confidence that the entailment is false. Our approach is inspired by an analysis of the RTE dataset that suggested a syntax-based approach should be approximately twice as effective at predicting false entailment as true entailment (Vanderwende and Dolan, 2006). The analysis implied that a great deal of syntactic information remained unexploited by existing systems, but gave few explicit suggestions on how syntactic information should be applied; this paper provides a starting point for creating the heuristics capable of obtaining the bound they suggest[2].

2 System Description

Similar to most other syntax-based approaches to recognizing textual entailment, we begin by representing each text and hypothesis sentence pair in *logical forms*. These logical forms are generated using NLPWIN[3], a robust system for natural language parsing and generation (Heidorn, 2000). Our logical form representation may be considered equivalently as a set of triples of the form RELATION$(node_i, node_j)$, or as a graph of syntactic dependencies; we use both terminologies interchangeably. Our algorithm proceeds as follows:

1. Parse each sentence with the NLPWIN parser, resulting in syntactic dependency graphs for the text and hypothesis sentences.
2. Attempt an alignment of each *content* node in the dependency graph of the hypothesis sentence to some node in the graph of the text sentence, using a set of heuristics for alignment (described in Section 3).
3. Using the alignment, apply a set of syntactic heuristics for recognizing false entailment (described in Section 4); if any match, predict that the entailment is false.

[2](Vanderwende and Dolan, 2006) suggest that the truth or falsehood of 48% of the entailment examples in the RTE test set could be correctly identified via syntax and a thesaurus alone; thus by random guessing on the rest of the examples one might hope for an accuracy level of $0.48 + \frac{0.52}{2} = 74\%$.

[3]To aid in the replicability of our experiments, we have published the NLPWIN logical forms for all sentences from the development and test sets in the PASCAL RTE dataset at http://research.microsoft.com/nlp/Projects/RTE.aspx.

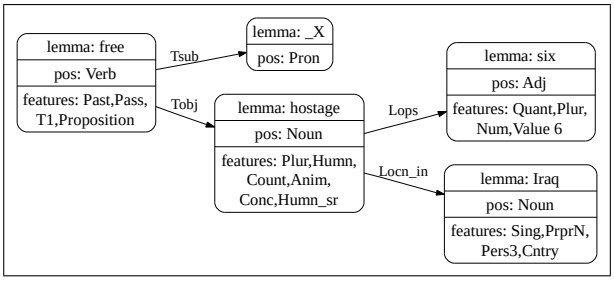

Figure 1: Logical form produced by NLPWIN for the sentence "Six hostages in Iraq were freed."

4. If no syntactic heuristic matches, back off to a lexical similarity model (described in section 5.1), with an attempt to align detected paraphrases (described in section 5.2).

In addition to the typical syntactic information provided by a dependency parser, the NLPWIN parser provides an extensive number of semantic features obtained from various linguistic resources, creating a rich environment for feature engineering. For example, Figure 1 (from Dev Ex. #616) illustrates the dependency graph representation we use, demonstrating the stemming, part-of-speech tagging, syntactic relationship identification, and semantic feature tagging capabilities of NLPWIN.

We define a *content* node to be any node whose lemma is not on a small stoplist of common stop words. In addition to content vs. non-content nodes, among content nodes we distinguish between *entities* and *nonentities*: an *entity* node is any node classified by the NLPWIN parser as being a proper noun, quantity, or time.

Each of the features of our system were developed from inspection of sentence pairs from the RTE development data set, and used in the final system only if they improved the system's accuracy on the development set (or improved F-score if accuracy was unchanged); sentence pairs in the RTE test set were left uninspected and used for testing purposes only.

3 Linguistic cues for node alignment

Our syntactic heuristics for recognizing false entailment rely heavily on the correct alignment of words and multiword units between the text and hypothesis logical forms. In the notation below, we will consider h and t to be nodes in the hypothesis H and

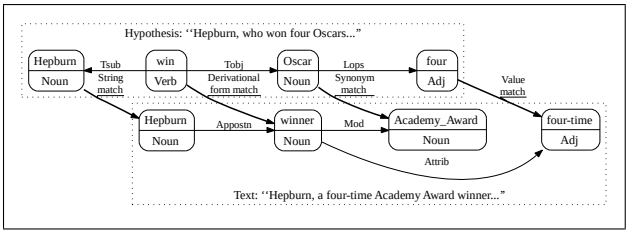

Figure 2: Example of synonym, value, and derivational form alignment heuristics, Dev Ex. #767

text T logical forms, respectively. To accomplish the task of node alignment we rely on the following heuristics:

3.1 WordNet synonym match

As in (Herrera et al., 2005) and others, we align a node $h \in H$ to any node $t \in T$ that has both the same part of speech and belongs to the same synset in WordNet. Our alignment considers multiword units, including compound nouns (e.g., we align "Oscar" to "Academy Award" as in Figure 2), as well as verb-particle constructions such as "set off" (aligned to "trigger" in Test Ex. #1983).

3.2 Numeric value match

The NLPWIN parser assigns a normalized numeric value feature to each piece of text inferred to correspond to a numeric value; this allows us to align "6th" to "sixth" in Test Ex. #1175. and to align "a dozen" to "twelve" in Test Ex. #1231.

3.3 Acronym match

Many acronyms are recognized using the synonym match described above; nonetheless, many acronyms are not yet in WordNet. For these cases we have a specialized acronym match heuristic which aligns pairs of nodes with the following properties: if the lemma for some node h consists only of capitalized letters (with possible interceding periods), and the letters correspond to the first characters of some multiword lemma for some $t \in T$, then we consider h and t to be aligned. This heuristic allows us to align "UNDP" to "United Nations Development Programme" in Dev Ex. #357 and "ANC" to "African National Congress" in Test Ex. #1300.

3.4 Derivational form match

We would like to align words which have the same root form (or have a synonym with the same root form) and which possess similar semantic meaning, but which may belong to different syntactic categories. We perform this by using a combination of the synonym and derivationally-related form information contained within WordNet. Explicitly our procedure for constructing the set of derivationally-related forms for a node h is to take the union of all derivationally-related forms of all the synonyms of h (including h itself), i.e.:

$$\text{DERIV}(h) = \cup_{s \in \text{WN-SYN}(h)} \text{WN-DERIV}(s)$$

In addition to the noun/verb derivationally-related forms, we detect adjective/adverb derivationally-related forms that differ only by the suffix 'ly'.

Unlike the previous alignment heuristics, we do not expect that two nodes aligned via derivationally-related forms will play the same syntactic role in their respective sentences. Thus we consider two nodes aligned in this way to be *soft-aligned*, and we do not attempt to apply our false entailment recognition heuristics to nodes aligned in this way.

3.5 Country adjectival form / demonym match

As a special case of derivational form match, we soft-align matches from an explicit list of place names, adjectival forms, and demonyms[4]; e.g., "Sweden" and "Swedish" in Test Ex. #1576.

3.6 Other heuristics for alignment

In addition to these heuristics, we implemented a hyponym match heuristic similar to that discussed in (Herrera et al., 2005), and a heuristic based on the string-edit distance of two lemmas; however, these heuristics yielded a decrease in our system's accuracy on the development set and were thus left out of our final system.

4 Recognizing false entailment

The bulk of our system focuses on heuristics for recognizing false entailment. For purposes of notation, we define binary functions for the existence

[4]List of adjectival forms and demonyms based on the list at: http://en.wikipedia.org/wiki/List_of_demonyms

Unaligned Entity:	$\text{ENTITY}(h) \wedge \forall t. \neg \text{ALIGN}(h,t) \rightarrow False.$
Negation Mismatch:	$\text{ALIGN}(h,t) \wedge \text{NEG}(t) \neq \text{NEG}(h) \rightarrow False.$
Modal Mismatch:	$\text{ALIGN}(h,t) \wedge \text{MOD}(t) \wedge \neg\text{MOD}(h) \rightarrow False.$
Antonym Match:	$\text{ALIGN}(h_1,t_1) \wedge \text{REL}(h_0,h_1) \wedge \text{REL}(t_0,t_1) \wedge \text{LEMMA}(t_0) \in \text{ANTONYMS}(h_0) \rightarrow False$
Argument Movement:	$\text{ALIGN}(h_1,t_1) \wedge \text{ALIGN}(h_2,t_2) \wedge \text{REL}(h_1,h_2) \wedge \neg\text{REL}(t_1,t_2) \wedge \text{REL} \in \{\text{SUBJ, OBJ, IND}\} \rightarrow False$
Superlative Mismatch:	$\neg(\text{SUPR}(h_1) \rightarrow (\text{ALIGN}(h_1,t_1) \wedge \text{ALIGN}(h_2,t_2) \wedge \text{REL}_1(h_2,h_1) \wedge \text{REL}_1(t_2,t_1)$ $\wedge \forall t_3.(\text{REL}_2(t_2,t_3) \wedge \text{REL}_2 \in \{\text{MOD,POSSR,LOCN}\} \rightarrow \text{REL}_2(h_2,h_3) \wedge \text{ALIGN}(h_3,t_3))) \rightarrow False$
Conditional Mismatch:	$\text{ALIGN}(h_1,t_1) \wedge \text{ALIGN}(h_2,t_2) \wedge \text{COND} \in \text{PATH}(t_1,t_2) \wedge \text{COND} \notin \text{PATH}(h_1,h_2) \rightarrow False$

Table 1: Summary of heuristics for recognizing false entailment

of each semantic node feature recognized by NLPWIN; e.g., if h is negated, we state that $\text{NEG}(h) = \text{TRUE}$. Similarly we assign binary functions for the existence of each syntactic relation defined over pairs of nodes. Finally, we define the function $\text{ALIGN}(h,t)$ to be true if and only if the node $h \in H$ has been 'hard-aligned' to the node $t \in T$ using one of the heuristics in Section 3. Other notation is defined in the text as it is used. Table 1 summarizes all heuristics used in our final system to recognize false entailment.

4.1 Unaligned entity

If some node h has been recognized as an entity (i.e., as a proper noun, quantity, or time) but has not been aligned to any node t, we predict that the entailment is false. For example, we predict that Test Ex. #1863 is false because the entities "Suwariya", "20 miles", and "35" in H are unaligned.

4.2 Negation mismatch

If any two nodes (h,t) are aligned, and one (and only one) of them is negated, we predict that the entailment is false. Negation is conveyed by the NEG feature in NLPWIN. This heuristic allows us to predict false entailment in the example "Pertussis is not very contagious" and "...pertussis, is a highly contagious bacterial infection" in Test Ex. #1144.

4.3 Modal auxiliary verb mismatch

If any two nodes (h,t) are aligned, and t is modified by a modal auxiliary verb (e.g, *can*, *might*, *should*, etc.) but h is not similarly modified, we predict that the entailment is false. Modification by a modal auxiliary verb is conveyed by the MOD feature in NLPWIN. This heuristic allows us to predict false entailment between the text phrase "would constitute a threat to democracy", and the hypothesis phrase "constitutes a democratic threat" in Test Ex. #1203.

4.4 Antonym match

If two aligned noun nodes (h_1,t_1) are both subjects or both objects of verb nodes (h_0,t_0) in their respective sentences, i.e., $\text{REL}(h_0,h_1) \wedge \text{REL}(t_0,t_1) \wedge \text{REL} \in \{\text{SUBJ,OBJ}\}$, then we check for a verb antonym match between (h_0,t_0). We construct the set of verb antonyms using WordNet; we consider the antonyms of h_0 to be the union of the antonyms of the first three senses of $\text{LEMMA}(h_0)$, or of the nearest antonym-possessing hypernyms if those senses do not themselves have antonyms in WordNet. Explicitly our procedure for constructing the antonym set of a node h_0 is as follows:

1. $\text{ANTONYMS}(h_0) = \{\}$
2. For each of the first three listed senses s of $\text{LEMMA}(h_0)$ in WordNet:
   (a) While $|\text{WN-ANTONYMS}(s)| = 0$
      i. $s \leftarrow \text{WN-HYPERNYM}(s)$
   (b) $\text{ANTONYMS}(h_0) \leftarrow \text{ANTONYMS}(h_0) \cup \text{WN-ANTONYMS}(s)$
3. return $\text{ANTONYMS}(h_0)$

In addition to the verb antonyms in WordNet, we detect the prepositional antonym pairs (*before/after*, *to/from*, and *over/under*). This heuristic allows us to predict false entailment between "Black holes can lose mass..." and "Black holes can regain some of their mass..." in Test Ex. #1445.

4.5 Argument movement

For any two aligned verb nodes (h_1,t_1), we consider each noun child h_2 of h_1 possessing any of

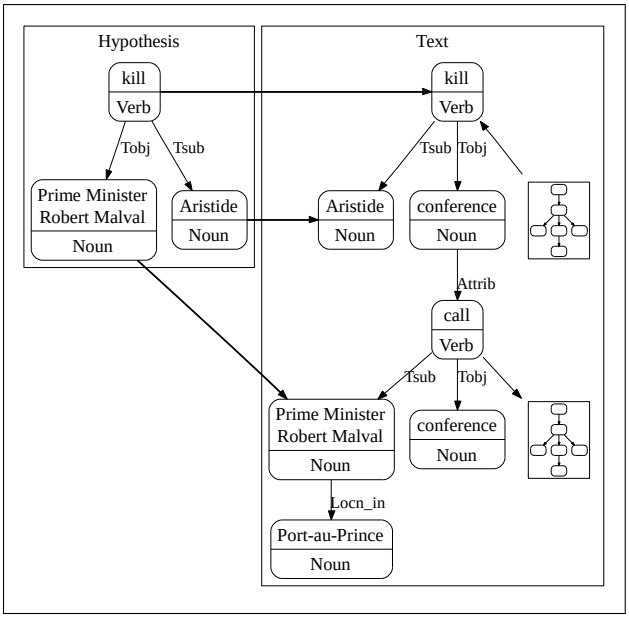

Figure 3: Example of object movement signaling false entailment

the subject, object, or indirect object relations to h_1, i.e., there exists $\text{REL}(h_1, h_2)$ such that $\text{REL} \in \{\text{SUBJ}, \text{OBJ}, \text{IND}\}$. If there is some node t_2 such that $\text{ALIGN}(h_2, t_2)$, but $\text{REL}(t_1, t_2) \neq \text{REL}(h_1, h_2)$, then we predict that the entailment is false.

As an example, consider Figure 3, representing subgraphs from Dev Ex. #1916:

T: ...U.N. officials are also dismayed that Aristide killed a conference called by Prime Minister Robert Malval...

H: Aristide kills Prime Minister Robert Malval.

Here let (h_1, t_1) correspond to the aligned verbs with lemma *kill*, where the object of h_1 has lemma *Prime Minister Robert Malval*, and the object of t_1 has lemma *conference*. Since h_2 is aligned to some node t_2 in the text graph, but $\neg\text{OBJ}(t_1, t_2)$, the sentence pair is rejected as a false entailment.

4.6 Superlative mismatch

If some adjective node h_1 in the hypothesis is identified as a superlative, check that all of the following conditions are satisfied:

1. h_1 is aligned to some superlative t_1 in the text sentence.

2. The noun phrase h_2 modified by h_1 is aligned to the noun phrase t_2 modified by t_1.

3. Any additional modifier t_3 of the noun phrase t_2 is aligned to some modifier h_3 of h_2 in the hypothesis sentence (reverse subset match).

If any of these conditions are not satisfied, we predict that the entailment is false. This heuristic allows us to predict false entailment in (Dev Ex. #908):

T: Time Warner is the world's largest media and Internet company.

H: Time Warner is the world's largest company.

Here "largest media and Internet company" in T fails the reverse subset match (condition 3) to "largest company" in H.

4.7 Conditional mismatch

For any pair of aligned nodes (h_1, t_1), if there exists a second pair of aligned nodes (h_2, t_2) such that the shortest path $\text{PATH}(t_1, t_2)$ in the dependency graph T contains the conditional relation, then $\text{PATH}(h_1, h_2)$ must also contain the conditional relation, or else we predict that the entailment is false. For example, consider the following false entailment (Dev Ex. #60):

T: If a Mexican approaches the border, he's assumed to be trying to illegally cross.

H: Mexicans continue to illegally cross border.

Here, "Mexican" and "cross" are aligned, and the path between them in the text contains the conditional relation, but does not in the hypothesis; thus the entailment is predicted to be false.

4.8 Other heuristics for false entailment

In addition to these heuristics, we additionally implemented an IS-A mismatch heuristic, which attempted to discover when an IS-A relation in the hypothesis sentence was not implied by a corresponding IS-A relation in the text; however, this heuristic yielded a loss in accuracy on the development set and was therefore not included in our final system.

5 Lexical similarity and paraphrase detection

5.1 Lexical similarity using MindNet

In case none of the preceding heuristics for rejection are applicable, we back off to a lexical similarity model similar to that described in (Glickman et al., 2005). For every content node $h \in H$

not already aligned by one of the heuristics in Section 3, we obtain a similarity score $\text{MN}(h,t)$ from a similarity database that is constructed automatically from the data contained in MindNet[5] as described in (Richardson, 1997). Our similarity function is thus:

$$sim(h,t) = \begin{cases} 1 & \text{if ANY-ALIGN}(h,t) \\ \text{MN}(h,t) & \text{if MN}(h,t) > min \\ min & \text{otherwise} \end{cases}$$

Where the minimum score min is a parameter tuned for maximum accuracy on the development set; $min = 0.00002$ in our final system. We then compute the entailment score:

$$score(H,T) = \frac{1}{|H|} \prod_{h \in H} \max_{t \in T} sim(h,t)$$

This approach is identical to that used in (Glickman et al., 2005), except that we use alignment heuristics and MindNet similarity scores in place of their web-based estimation of lexical entailment probabilities, and we take as our score the geometric mean of the component entailment scores rather than the unnormalized product of probabilities.

5.2 Measuring phrasal similarity using the web

The methods discussed so far for alignment are limited to aligning pairs of single words or multiple-word units constituting single syntactic categories; these are insufficient for the problem of detecting more complicated paraphrases. For example, consider the following true entailment (Dev Ex. #496):

T: ...Muslims believe there is only one God.

H: Muslims are monotheistic.

Here we would like to align the hypothesis phrase "are monotheistic" to the text phrase "believe there is only one God"; unfortunately, single-node alignment aligns only the nodes with lemma "Muslim". In this section we describe the approach used in our system to approximate phrasal similarity via distributional information obtained using the MSN Search search engine.

We propose a metric for measuring phrasal similarity based on a phrasal version of the distributional hypothesis: we propose that a phrase template P_h (e.g. 'x_h are monotheistic') has high semantic similarity to a template P_t (e.g. "x_t believe there is only one God"), with possible "slot-fillers" x_h and x_t, respectively, if the overlap of the sets of observed slot-fillers $X_h \cap X_t$ for those phrase templates is high in some sufficiently large corpus (e.g., the Web).

To measure phrasal similarity we issue the surface text form of each candidate phrase template as a query to a web-based search engine, and parse the returned sentences in which the candidate phrase occurs to determine the appropriate slot-fillers. For example, in the above example, we observe the set of slot-fillers $X_t = \{$Muslims, Christians, Jews, Saivities, Sikhs, Caodaists, People$\}$, and $X_h \cap X_t = \{$Muslims, Christians, Jews, Sikhs, People$\}$.

Explicitly, given the text and hypothesis logical forms, our algorithm proceeds as follows to compute the phrasal similarity between all phrase templates in H and T:

1. For each pair of aligned single node and unaligned leaf node (t_1, t_l) (or pair of aligned nodes (t_1, t_2)) in the text T:
   (a) Use NLPWIN to generate a surface text string S from the underlying logical form $\text{PATH}(t_1, t_2)$.
   (b) Create the surface string template phrase P_t by removing from S the lemmas corresponding to t_1 (and t_2, if path is between aligned nodes).
   (c) Perform a web search for the string P_t.
   (d) Parse the resulting sentences containing P_t and extract all non-pronoun slot fillers $x_t \in X_t$ that satisfy the same syntactic roles as t_1 in the original sentence.
2. Similarly, extract the slot fillers X_h for each discovered phrase template P_h in H.
3. Calculate paraphrase similarity as a function of the overlap between the slot-filler sets X_t and X_h, i.e: $score(P_h, P_t) = \frac{|X_h \cap X_t|}{|X_t|}$.

We then incorporate paraphrase similarity within the lexical similarity model by allowing, for some unaligned node $h \in P_h$, where $t \in P_t$:

$$sim(h,t) = \max(\text{MN}(h,t), score(P_h, P_t))$$

[5] http://research.microsoft.com/mnex

Our approach to paraphrase detection is most similar to the TE/ASE algorithm (Szpektor et al., 2004), and bears similarity to both DIRT (Lin and Pantel, 2001) and KnowItAll (Etzioni et al., 2004). The chief difference in our algorithm is that we generate the surface text search strings from the parsed logical forms using the generation capabilities of NLPWIN (Aikawa et al., 2001), and we verify that the syntactic relations in each discovered web snippet are isomorphic to those in the original candidate paraphrase template.

6 Results and Discussion

In this section we present the final results of our system on the PASCAL RTE-1 test set, and examine our features in an ablation study. The PASCAL RTE-1 development and test sets consist of 567 and 800 examples, respectively, with the test set split equally between true and false examples.

6.1 Results and Performance Comparison on the PASCAL RTE-1 Test Set

Table 2 displays the accuracy and confidence-weighted score[6] (CWS) of our final system on each of the tasks for both the development and test sets.

Our overall test set accuracy of 62.50% represents a 2.1% absolute improvement over the task-independent system described in (Tatu and Moldovan, 2005), and a 20.2% relative improvement in accuracy over their system with respect to an uninformed baseline accuracy of 50%.

To compute confidence scores for our judgments, any entailment determined to be false by any heuristic was assigned maximum confidence; no attempts were made to distinguish between entailments rejected by different heuristics. The confidence of all other predictions was calculated as the absolute value in the difference between the output $score(H, T)$ of the lexical similarity model and the threshold $t = 0.1285$ as tuned for highest accuracy on our development set. We would expect a higher CWS to result from learning a more appropriate confidence function; nonetheless our overall test set CWS of 0.6534 is higher than previously-reported task-independent systems (however, the task-dependent system reported in (Raina et al., 2005) achieves a CWS of 0.686).

[6] As in (Dagan et al., 2005) we compute the confidence-weighted score (or "average precision") over n examples $\{c_1, c_2, ..., c_n\}$ ranked in order of decreasing confidence as $cws = \frac{1}{n} \sum_{i=1}^{n} \frac{(\#\text{correct-up-to-rank-}i)}{i}$

	Dev Set		Test Set	
Task	**acc**	**cws**	**acc**	**cws**
CD	0.8061	0.8357	0.7867	0.8261
RC	0.5534	0.5885	0.6429	0.6476
IR	0.6857	0.6954	0.6000	0.6571
MT	0.7037	0.7145	0.6000	0.6350
IE	0.5857	0.6008	0.5917	0.6275
QA	0.7111	0.7121	0.5308	0.5463
PP	0.7683	0.7470	0.5200	0.5333
All	0.6878	0.6888	0.6250	0.6534

Table 2: Summary of accuracies and confidence-weighted scores, by task

Alignment Feature	**Dev**	**Test**
Synonym Match	0.0106	0.0038
Derivational Form	0.0053	0.0025
Paraphrase	0.0053	0.0000
Lexical Similarity	0.0053	0.0000
Value Match	0.0017	0.0013
Acronym Match	0.0017	0.0013
Adjectival Form[7]	0.0000	0.0063
False Entailment Feature	**Dev**	**Test**
Negation Mismatch	0.0106	0.0025
Argument Movement	0.0070	0.0250
Conditional Mismatch	0.0053	0.0037
Modal Mismatch	0.0035	0.0013
Superlative Mismatch	0.0035	-0.0025
Entity Mismatch	0.0018	0.0063

Table 3: Feature ablation study; quantity is the accuracy loss obtained by removal of single feature

6.2 Feature analysis

Table 3 displays the results of our feature ablation study, analyzing the individual effect of each feature.

Of the seven heuristics used in our final system for node alignment (including lexical similarity and paraphrase detection), our ablation study showed

[7] As discussed in Section 2, features with no effect on development set accuracy were included in the system if and only if they improved the system's unweighted F-score.

that five were helpful in varying degrees on our test set, but that removal of either MindNet similarity scores or paraphrase detection resulted in no accuracy loss on the test set.

Of the six false entailment heuristics used in the final system, five resulted in an accuracy improvement on the test set (the most effective by far was the "Argument Movement", resulting in a net gain of 20 correctly-classified false examples); inclusion of the "Superlative Mismatch" feature resulted in a small net loss of two examples.

We note that our heuristics for false entailment, where applicable, were indeed significantly more accurate than our final system as a whole; on the set of examples predicted false by our heuristics we had 71.3% accuracy on the training set (112 correct out of 157 predicted), and 72.9% accuracy on the test set (164 correct out of 225 predicted).

7 Conclusion

In this paper we have presented and analyzed a system for recognizing textual entailment focused primarily on the recognition of *false* entailment, and demonstrated higher performance than achieved by previous approaches on the widely-used PASCAL RTE test set. Our system achieves state-of-the-art performance despite not exploiting a wide array of sources of knowledge used by other high-performance systems; we submit that the performance of our system demonstrates the unexploited potential in features designed specifically for the recognition of false entailment.

Acknowledgments

We thank Chris Brockett, Michael Gamon, Gary Kacmarick, and Chris Quirk for helpful discussion. Also, thanks to Robert Ragno for assistance with the MSN Search API. Rion Snow is supported by an NDSEG Fellowship sponsored by the DOD and AFOSR.

References

Takako Aikawa, Maite Melero, Lee Schwartz, and Andi Wu. 2001. Multilingual Sentence Generation. In *Proc. of 8^{th} European Workshop on Natural Language Generation*.

Samuel Bayer, John Burger, Lisa Ferro, John Henderson, and Alexander Yeh. 2005. MITRE's Submissions to the EU Pascal RTE Challenge. In *Proc. of the PASCAL Challenges Workshop on RTE 2005*.

Johan Bos and Katja Markert. 2005. Recognizing Textual Entailment with Logical Inference. In *Proc. HLT-EMNLP 2005*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on RTE 2005*.

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2004. Web-scale information extraction in KnowItAll. In *Proc. WWW 2004*.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.

Oren Glickman, Ido Dagan, and Moshe Koppel. 2005. Web Based Probabilistic Textual Entailment. In *Proc. of the PASCAL Challenges Workshop on RTE 2005*.

George E. Heidorn. 2000. Intelligent Writing Assistance. In R. Dale, H. Moisl, and H. Somers (eds.), *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*. Marcel Dekker, New York. 181-207.

Jesús Herrera, Anselmo Peñas, and Felisa Verdejo. 2005. Textual Entailment Recognision Based on Dependency Analysis and WordNet. In *Proc. of the PASCAL Challenges Workshop on RTE 2005*.

Dekang Lin and Patrick Pantel. 2001. DIRT - Discovery of Inference Rules from Text. In *Proc. KDD 2001*.

Rajat Raina, Andrew Y. Ng, and Christopher D. Manning. 2005. Robust textual inference via learning and abductive reasoning. In *Proc. AAAI 2005*.

Stephen D. Richardson. 1997. Determining Similarity and Inferring Relations in a Lexical Knowledge Base. Ph.D. thesis, The City University of New York.

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling Web-based Acquisition of Entailment Relations. In *Proc. EMNLP 2004*.

Marta Tatu and Dan Moldovan. 2005. A Semantic Approach to Recognizing Textual Entailment. In *Proc. HLT-EMNLP 2005*.

Lucy Vanderwende and William B. Dolan. 2006. What Syntax Can Contribute in the Entailment Task. In *MLCW 2005*, LNAI 3944, pp. 205–216. J. Quinonero-Candela et al. (eds.). Springer-Verlag.