# User's Choice of Precision and Recall in Named Entity Recognition

Roman Klinger and Christoph M. Friedrich
Fraunhofer Institute for Algorithms and Scientific Computing (SCAI)
Department of Bioinformatics
Schloss Birlinghoven
53754 Sankt Augustin, Germany
{roman.klinger,christoph.friedrich}@scai.fraunhofer.de

## Abstract

Conditional Random Fields are commonly trained to maximize likelihood. The corresponding $F_\beta$ measure, the weighted harmonic mean of precision and recall, which is established for evaluation in information retrieval and text mining, is not necessarily the optimal result for the user's choice of $\beta$.

Some approaches have been published to optimize multivariate measures like $F_\beta$ to overcome this inconsistency. The limitation is that constraints like the value of $\beta$ have to be known at training time.

This publication proposes a method of multi-objective optimization of both precision and recall based on a preceding likelihood training. The output is an estimation of pareto-optimal solutions from which the user can select the best for the actual application. Evaluated on two publicly available data sets in the field of named entity recognition, nearly all $F_\beta$ values are superior to those resulting from log-likelihood training.

## Keywords

Named Entity Recognition, Conditional Random Fields, Multi-Objective Optimization, NSGA-II, $F_\beta$ measure, Recall, Precision

## 1 Introduction

In information retrieval, the $F_\beta$ measure, the weighted harmonic mean between recall and precision, is established as evaluation measure. The corresponding $\beta$ value to be chosen is application-depend. Methods for selecting $\beta$ at training time exist for Support Vector Machines [18], Logistic Regression as well as Conditional Random Fields (CRF) [11] all of which are classically optimized by means of accuracy-related measures [7, 8, 20]. A similar goal is known from the AmilCare system [4] with the main focus on user involvement.

At inference time, a parameter to select between higher precision or recall can be introduced by changing the decision threshold for an adequate decision function $d(\cdot) \in \mathbb{R}$. In sequential segmentation tasks like named entity recognition (NER), precision can be increased with this approach without retraining. Increasing recall is possible with the allowance of overlaps as demonstrated for gene and protein names [3]. This requires the computation of reliable confidences, which increasing runtime is a drawback especially during inference [5, 20].

In contrast to optimizing one special value or selecting the set of output entities in prediction phase, we propose to use an evolutionary optimization scheme to optimize recall and precision in a multi-objective way to yield different model configurations, which can be selected by an end-user depending on the respective task with higher recall or higher precision without retraining. Thereby, the non-intentional choice of precision and recall by optimization of accuracy (which is performed by maximizing the log-likelihood of the model given the training data in the case of CRFs) is avoided.

The main contribution of this paper is therefore the presentation of multi-objective optimization for Conditional Random Fields (MOCRF). The feasibility of evolutionary optimization in such models is demonstrated. The resulting possibility to choose a $\beta$ for $F_\beta$ evaluation is meaningful for information retrieval tasks often demanding for a high recall or information extraction with the need for a high precision.

## 2 Methods

### 2.1 Conditional Random Fields and Text Segmentation

Conditional Random Fields (CRF) [11, 13] are a family of probabilistic, undirected graphical models for computing the probability $P_{\vec{\lambda}}(\vec{y}|\vec{x})$ of a possible label sequence $\vec{y} = (y_0, \ldots, y_n)$ given the input sequence $\vec{x} = (x_0, \ldots, x_n)$. In the context of named entity recognition, this observation sequence $\vec{x}$ corresponds to the tokenized text. The label sequence is encoded in a label alphabet $\mathcal{L} = \{I\text{-}<entity>, O, B\text{-}<entity>\}$ where $y_i = O$ means that $x_i$ is outside an entity, $y_i = B\text{-}<entity>$ means that $x_i$ is the beginning and $y_i = I\text{-}<entity>$ means that $x_i$ is inside an entity. Using this IOB alphabet, named entity recognition is modelled as text segmentation task. An example for an input sequence and possible output sequences is shown in Table 1 taken from data of [21].



*International Conference RANLP 2009 - Borovets, Bulgaria*, pages 192–196

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\vec{x} =$ ( | or | chicken | beta-actin | ( | cBA | ) | gene | were | injected | $)^T$ |
| $\vec{y^*} =$ ( | O | B | I | O | B | O | O | O | O | $)^T$ |
| $\vec{y'} =$ ( | O | B | I | O | O | O | O | O | O | $)^T$ |
| $\vec{y''} =$ ( | O | B | I | B | I | I | I | O | O | $)^T$ |

Table 1: Named Entity Recognition example input sequence with possible output sequences. For better perceptibility, segments have been underlined additionally. The correct sequence is $\vec{y^*}$, $\vec{y'}$ and $\vec{y'}$ are possible predictions. (annotations from [21])

| | | Correct | |
|---|---|---|---|
| | | $C_1$ | $\neg C_1$ |
| Predict | $C_1$ | TP | FP |
| | $\neg C_1$ | FN | TN |

Table 2: Contingency table for two classes $C_1$ and not $C_1$ ($\neg C_1$) used to compute different evaluation measures.

Assuming $\vec{y^*}$ to be the correct segmentation and $\vec{y''}$ to be the predicted sequence, the result is 1 TP (true positive), 1 FN (false negative) and 1 FP (false positive). Only predicting the first segment and not the second one leads to a better result with 1 TP and 1 FN (as $\vec{y'}$ in Table 1). This is a reason why it is easier to get a high precision than a high recall (compare with measures in Section 2.2). Given a predicted sequence and confidence scores, it is therefore easy to increase precision by removing unconfident entities. But it can easily be seen that adding entities to a result is not straight-forward, as searching for candidates is necessary.

A linear-chain CRF is given by $P_{\vec{\lambda}}(\vec{y}|\vec{x}) = \exp(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y_{j-1}, y_j, \vec{x}, j))/Z(\vec{x})$ with $Z(\vec{x}) = \sum_{\vec{y}\in\mathcal{Y}}\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\Big(y_{j-1}, y_j, \vec{x}, j\Big)\right)$ as normalization, where $\mathcal{Y}$ is the set of all possible label sequences. Each feature $f_i(\cdot)$ is weighted by $\lambda_i \in \mathbb{R}$. These weights are the parameters to be learned in the model. Optimization of the parameters $\lambda_i$ is typically performed with the limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS [16]) on the logarithmic likelihood, the convex function $L(\mathcal{T})$ with the training data $\mathcal{T}$, including a penalty term: $L(\mathcal{T}) = \log P_{\vec{\lambda}}(\vec{y}|\vec{x})) - \sum_{i=1}^{m}(\lambda_i^2/2\sigma^2)$.

## 2.2 Evaluation Measures

All measures used in this work are based on the contingency table shown in Table 2 [22]. The entries in the table denote frequencies of instances being true positives (TP), false positives (FP), true negatives (TN), or false negatives (FN). These values are functions of a model configuration $\vec{\lambda}$ and some data $\mathcal{D} \ni (\vec{x}, \vec{y})$ consisting of text sequences $\vec{x}$ and given label sequences $\vec{y}$. Optimizing a CRF with respect to $L(\mathcal{T})$ corresponds to maximization of accuracy which is defined as

$$\mathrm{acc}(\vec{\lambda}, \mathcal{D}) = \frac{TP + TN}{TP + FP + TN + FN}\,. \qquad (1)$$

Closely related is the precision

$$\mathrm{prec}(\vec{\lambda}, \mathcal{D}) = \frac{TP}{TP + FP} \qquad (2)$$

which is combined with recall

$$\mathrm{rec}(\vec{\lambda}, \mathcal{D}) = \frac{TP}{TP + FN} \qquad (3)$$

to form the $F_\beta$ measure

$$F_\beta(\vec{\lambda}, \mathcal{D}) = \frac{(1+\beta^2)\cdot \mathrm{prec}(\vec{\lambda}, \mathcal{D})\cdot \mathrm{rec}(\vec{\lambda}, \mathcal{D})}{\beta^2\cdot \mathrm{prec}(\vec{\lambda}, \mathcal{D}) + \mathrm{rec}(\vec{\lambda}, \mathcal{D})} \qquad (4)$$

The discrepancy between optimizing accuracy and evaluating $F_\beta$ measures is based on the fact that the first is not differentiating between false positives and false negatives nor between true positives and true negatives while the latter does by incorporating recall and precision.[1]

## 2.3 Non-dominated Sorting Genetic Algorithm II (NSGA-II)

The NSGA-II is an evolutionary optimization scheme for multi-objective optimization presented here briefly. For details, we refer to the original work [6].

As usual in evolutionary computation [1], main aspects are recombination, mutation and selection of a population of individuals representing solutions of a problem. Each has one or more assigned objective values. For multi-objective optimization the population is maintained to consist of diverse solutions. The result of the process is a population of non-dominated individuals near the real pareto-optimal front. Domination means that a solution has at least one better and no worse objective value than another solution.

In each iteration of the optimization procedure, sorting of the individuals is necessary with respect to the non-domination. The result is a partition of the population into domination fronts, i.e., each individual $I$ has an assigned rank $\mathrm{r}(I) \in \mathbb{N}$.

As mentioned, the population needs to be divers and cover the pareto-front with a good spread. This is achieved by assigning a *crowding distance* $c(I) \in \mathbb{R}^+$ to each individual. This measure represents the average distance to the individuals with most similar objective values in the same front.

[1] A state-of-the-art approach to overcome this is the minimum classification error (MCE) framework [20].

These two values are used to define the comparator $\prec$ and sort the individuals of a population:

$$I_1 \prec I_2 \text{ if } \begin{cases} & (r(I_1) < r(I_2)) \\ \text{or} & (r(I_1) = r(I_2) \text{ and } c(I_1) > c(I_2)) \,. \end{cases} \tag{5}$$

This operator is used to select the individuals to form the succeeding population; in the original work, a tournament selection [15] is proposed.

The general workflow is as follows: First, the initial parent and offspring population is generated. In the evaluation step, the individuals are sorted with respect to $\prec$. By selection of the $q$ first individuals, the succeeding population is created. If the stop criterion (e.g. based on iteration number or values of objective functions) is not satisfied, this population is used in the next iteration to generate offspring by recombination and mutation and so on. The final set of solutions is defined by the last population.

### 2.4 Multi-Objective Optimization of CRFs (MOCRF)

To apply NSGA-II to optimize precision and recall we need to define initialization, recombination and mutation operators manipulating the parameters $\vec{\lambda} = \{\lambda_1, \ldots, \lambda_m\}$ of a CRF. Each individual in the following is represented by such a vector, therefore we refer to them as $\vec{\lambda}_k$ $(1 \leq k \leq q)$.

For initializing, a maximization of log-likelihood of an individual $\vec{\lambda}_1$ via L-BFGS is performed until convergence of the training algorithm. The initial population $P = \{\vec{\lambda}_1, \ldots, \vec{\lambda}_q\}$ consists of this individual and $n-1$ copies of the resulting parameters. The individuals $\vec{\lambda}_2, \ldots, \vec{\lambda}_q$ are modified with the mutation operator $\mathrm{mut}(\vec{\lambda})$: We add a normally distributed random value to each parameter:

$$\mathrm{mut}(\lambda_k) = \lambda_k + \mathcal{N}(0, \sigma) \,, \tag{6}$$

with $\mathcal{N}(\mu, \sigma)$ as a normally distributed random number with expectation value $\mu$ and standard deviation $\sigma \in \mathbb{R}$.

The recombination operator creates offspring from two parents (chosen by tournament selection). Two crossover variants are incorporated, in each application of recombination one is selected randomly: Intermediate recombination $\mathrm{im}(\vec{\lambda}_1, \vec{\lambda}_2)$ or one-point crossover $\mathrm{co}(\vec{\lambda}_1, \vec{\lambda}_2)$ [1] ($\lambda_{i,j}$ denotes component $j$ of individual $\vec{\lambda}_i$; $r \in [1, n] \subset \mathbb{N}$ a uniformly distributed random variable):

$$\mathrm{im}(\vec{\lambda}_1, \vec{\lambda}_2) = \Big((\lambda_{1,1} + \lambda_{2,1})/2, \ldots, (\lambda_{1,n} + \lambda_{2,n})/2\Big)^T \,, \tag{7}$$

$$\mathrm{co}(\vec{\lambda}_1, \vec{\lambda}_2) = \Big(\lambda_{1,1}, \ldots, \lambda_{1,r}, \lambda_{2,r+1}, \ldots, \lambda_{2,n}\Big)^T \,. \tag{8}$$

The objective functions are $\mathrm{prec}(\vec{\lambda}, \mathcal{D})$ and $\mathrm{rec}(\vec{\lambda}, \mathcal{D})$.

The implementation used in this work is based on [14]. It should be noted, that computing the objective functions can easily be done in parallel to decrease duration of the optimization process.

## 3 Experiments

In this section, the results for the proposed optimization approach are evaluated on two data sets from the field of named entity recognition. Parameter settings via cross-validation or bootstrapping are not a topic of this paper due to page limitations.

The standard deviation $\sigma$ (step size) used for mutating the individuals representing solutions is set to $\sigma = 0.01$. Greater step sizes would lead to a better exploration but a worse approximation of the real pareto-front. All experiments are performed with a population size of $q = 100$ and 100 iterations of the multi-objective optimization.

### 3.1 Data Sets

The results and evaluations are shown on the basis of two data sets with slightly different configurations of the CRF.

The BioCreative 2 Gene Mention Task data (BC2) contains entities of the class *Gene/Protein* with the specialty of acceptance of several boundaries for entities [21]. We incorporate the configuration of the CRF as described in a participating system using only the shortest possible annotation as exact true positive per entity [10, 19].

The ConLL data [17] is an annotation of the Reuters corpus [12] containing the classes *person*, *organization*, *locations* and *misc*. We use an order-one CRF with offset conjunction combining features of one preceding and succeeding token for each position in the text sequence. The feature set is fairly standard with Word-As-Class, prefix and suffix generation of length two, three and four as well as several regular expressions detecting capital letters, numbers, dashes and dots separately and as parts of tokens. The combination of the provided sets "train" and "testa" is used for training and "testb" for testing.

In both settings, a feature selection based on information gain is performed (namely *IG-OAA* [9]). For CoNLL, we use 38095 features and 22993 for BC2.

### 3.2 Results

Figure 1 depicts the final population for both data sets. The estimated pareto-fronts for the training and test sets are shown, each individual forming one position in the plot on each front is connected with a line. The boxes show the results of the initial individual trained to maximize log-likelihood. The blue, green and red line show the individual with highest $F_2$, $F_1$ and $F_{0.5}$ measure respectively.

The pareto-front on the training set is the one determined by MOCRF. The results shown as pareto-front on the test set are the results of the same individuals connected by a line. The absence of crossings to a

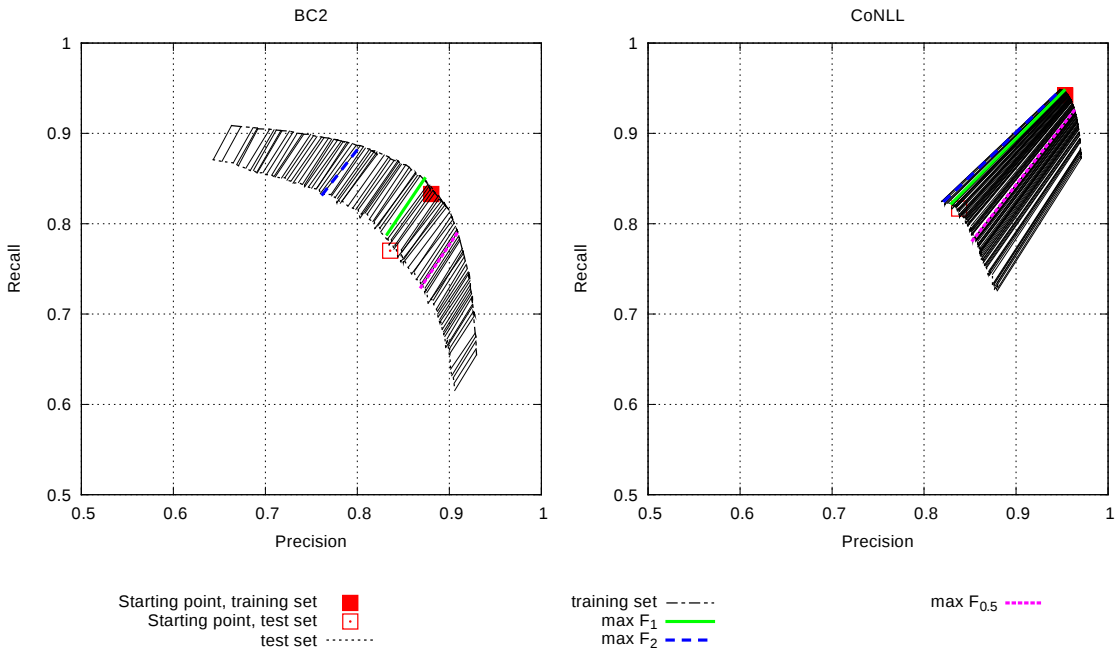


Figure 1: Results of the final population for $\sigma = 0.01$ without smoothing. Best $F_1$, $F_{0.5}$ and $F_2$ values are shown in bold colored lines, selected on the training set with the according values on the test set.

| | L-BFGS | | | | | MOCRF | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set | $F_{0.25}$ | $F_{0.5}$ | $F_1$ | $F_2$ | $F_4$ | $F_{0.25}$ | $F_{0.5}$ | $F_1$ | $F_2$ | $F_4$ |
| BC2 | 0.83 | 0.82 | 0.80 | 0.78 | 0.78 | 0.88 | 0.84 | 0.81 | 0.82 | 0.85 |
| CoNLL | 0.84 | 0.83 | 0.83 | 0.82 | 0.82 | 0.87 | 0.84 | 0.83 | 0.83 | 0.83 |

Table 3: Results for classic L-BFGS training in comparison to MOCRF. Given are the best available $F_\beta$ measures for $\beta = \{0.25, 0.5, 1, 2, 4\}$, as well as the result for L-BFGS for different data sets. All results are equal or better than for L-BFGS training which does not optimize with respect to a special $\beta$ value. These results are shown graphically in Figure 2.

large extent shows that the generalization from the results on the training set to the results on the test set is feasible.

It is noticeable, that the fronts seem to be differently well explored in BC2 and CoNLL data. On BC2 data, precision as well as recall can be increased at the expense of the other measure: The starting point is an $F_1$ measure of 0.86 with a precision of 0.88 and a recall of 0.83 on training data, highest possible precision is 0.93 (difference 0.05), highest possible recall is 0.90 (difference to start: 0.07). On ConLL data, the starting point is an $F_1$ measure of 0.94 with a precision of 0.95 and a recall of 0.94 on training data, highest possible precision is 0.97 (difference 0.02), highest possible recall is 0.95 (difference to start: 0.01). This difference between the data sets is founded by the structure of the problem and the different dependencies of the objective functions on the data sets. In both cases, a spread set of solutions is made available by the proposed method.

Assuming a user asking for a model characterized by an $F_\beta$ measure with fixed $\beta$, the provided system multi-objectively trained exhibits better performance than the one trained to maximize log-likelihood. This is shown in Table 3 and Figure 2. On BC2 data, the results are better for all $\beta$ values, for CoNLL data the results are the same for $F_1$, but superior for all other values.

On both data sets, the precision is higher than the recall for the model trained on log-likelihood. Therefore, $F_\beta$ is monotonically decreasing for that method. For MOCRF, higher values of precision than for recall are achieved. On BC2 data, this even leads to a minimum of $F_\beta$ for $\beta = 1$ as the same precision and recall are more difficult to achieve than other weightings. On CoNLL data, the exploration of recall is not as successful as on BC2 data.

# 4 Conclusions and Future Work

This paper presents the application of multi-objective optimization via NSGA-II to maximize precision and recall in Conditional Random Fields for named entity

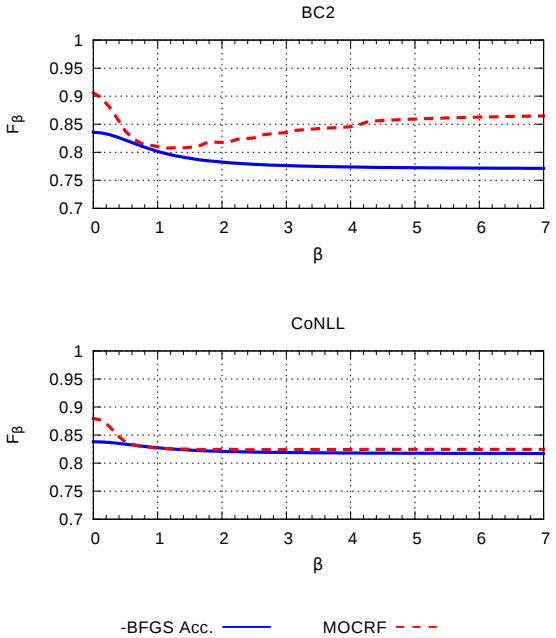


Figure 2: Results in $F_\beta$ for different $\beta$ on the result obtained via L-BFGS training w.r.t. log-likelihood and MOCRF.

recognition. It is shown on both data sets that $F_\beta$ measures for nearly all $\beta$ could be increased in comparison to classical maximization of log-likelihood via L-BFGS. This enables an end-user to choose a model with higher recall or precision without retraining or time-consuming computation of confidence measures. Possible applications include information retrieval with the need for a high recall to find most of the possible results, e.g. documents from a database as well as information extraction, where a high precision can help to detect correct relations between named entities.

Main future work is to evaluate other multi-objective optimization heuristics to improve the result in terms of a higher spread of solutions and possibly a better approximation of the real pareto-front. An integration of the initial training into the multi-objective optimization is also desirable.

# References


[1] T. Bäck, D. Fogel, and Z. Michaelewicz, editors. *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxforf University Press, Bristol. UK, 1997.

[2] A. L. Berger, S. D. Pietra, and V. J. D. Pietra. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71, 1996.

[3] B. Carpenter. LingPipe for 99.99 In *Proceedings of the 2nd BioCreative workshop*, Madrid, Spain, 2007.

[4] F. Ciravegna and D. Petrelli. User involvement in customizing adaptive Information Extraction: position paper. In *Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, Seattle, US, August 2001.

[5] A. Culotta and A. McCallum. Confidence Estimation for Information Extraction. In *Proceedings of HLT-NAACL*, pages 109–112, 2004.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[7] M. Jansche. Maximum Expected F-Measure Training of Logistic Regression Models. In *Proceedings of HLT/EMNLP*, Vancouver, October 2005.

[8] T. Joachims. A Support Vector Method for Multivariate Performance Measures. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 377–384, Bonn, 2005.

[9] R. Klinger and C. M. Friedrich. Feature Subset Selection in Conditional Random Fields for Named Entity Recognition. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria, 2009.

[10] R. Klinger, C. M. Friedrich, J. Fluck, and M. Hofmann-Apitius. Named Entity Recognition with Combinations of Conditional Random Fields. In *Proc. of the Second BioCreative Challenge Evaluation Workshop*, pages 89–91, 2007.

[11] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 282–289. Morgan Kaufmann Publishers, 2001.

[12] D. D. Lewis, Y. Yang, T. Rose, and F. Li. A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[13] R. McDonald and F. Pereira. Identifying Gene and Protein Mentions in Text Using Conditional Random Fields. *BMC Bioinformatics*, 6 Suppl 1:S6, 2005.

[14] J. Melcher. Java Non-Dominated Sorting Genetic Algorithm II (JNSGA II) – Implementation. Software, 2007. `http://sourceforge.net/projects/jnsga2`.

[15] B. L. Miller and D. E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9:193–212, 1995.

[16] J. Nocedal. Updating Quasi-Newton Matrices with Limited Storage. *Mathematics of Computation*, 35(151):773–782, July 1980.

[17] E. F. T. K. Sang and F. De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In W. Daelemans and M. Osborne, editors, *Proc. of CoNLL-2003*, pages 142–147. Edmonton, Canada, 2003.

[18] B. Schölkopf and A. J. Smola. *Learning with Kernels*. The MIT Press, 2002.

[19] L. Smith, L. K. Tanabe, R. J. nee Ando, C.-J. Kuo, I.-F. Chung, C.-N. Hsu, Y.-S. Lin, R. Klinger, C. M. Friedrich, K. Ganchev, M. Torii, H. Liu, B. Haddow, C. A. Struble, R. J. Povinelli, A. Vlachos, W. A. Baumgartner, L. Hunter, B. Carpenter, R. T.-H. Tsai, H.-J. Dai, F. Liu, Y. Chen, C. Sun, S. Katrenko, P. Adriaans, C. Blaschke, R. Torres, M. Neves, P. Nakov, A. Divoli, M. Maa-Lpez, J. Mata, and W. J. Wilbur. Overview of BioCreative II gene mention recognition. *Genome Biol*, 9 Suppl 2:S2, 2008.

[20] J. Suzuki, E. McDermott, and H. Isozaki. Training Conditional Random Fields with Multivariate Evaluation Measures. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 217–224. Association for Computational Linguistics, July 2006.

[21] J. Wilbur, L. Smith, and L. Tanabe. BioCreative 2. Gene Mention Task. In *Proc. of the Second BioCreative Challenge Evaluation Workshop*, pages 7–9, 2007.

[22] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.