Reranking for Biomedical Named-Entity Recognition

Kazuhiro Yoshida* **Jun'ichi Tsujii*†‡**
*Department of Computer Science, University of Tokyo
†School of Informatics, University of Manchester
‡National Center for Text Mining
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033 JAPAN
{kyoshida, tsujii}@is.s.u-tokyo.ac.jp

Abstract

This paper investigates improvement of automatic biomedical named-entity recognition by applying a reranking method to the COLING 2004 JNLPBA shared task of bio-entity recognition. Our system has a common reranking architecture that consists of a pipeline of two statistical classifiers which are based on log-linear models. The architecture enables the reranker to take advantage of features which are globally dependent on the label sequences, and features from the labels of other sentences than the target sentence. The experimental results show that our system achieves the labeling accuracies that are comparable to the best performance reported for the same task, thanks to the 1.55 points of F-score improvement by the reranker.

1 Introduction

Difficulty and potential application of biomedical named-entity recognition has attracted many researchers of both natural language processing and bioinformatics. The difficulty of the task largely stems from a wide variety of named entity expressions used in the domain. It is common for practical protein or gene databases to contain hundreds of thousands of items. Such a large variety of vocabulary naturally leads to long names with productive use of general words, making the task difficult to be solved by systems with naive Markov assumption of label sequences, because such systems must perform their prediction without seeing the entire string of the entities.

Importance of the treatment of long names might be implicitly indicated in the performance comparison of the participants of JNLPBA shared task (Kim et al., 2004), where the best performing system (Zhou and Su, 2004) attains their scores by extensive post-processing, which enabled the system to make use of global information of the entity labels. After the shared task, many researchers tackled the task by using conditional random fields (CRFs) (Lafferty et al., 2001), which seemed to promise improvement over locally optimized models like maximum entropy Markov models (MEMMs) (McCallum et al., 2000). However, many of the CRF systems developed after the shared task failed to reach the best performance achieved by Zhou et al. One of the reasons may be the deficiency of the dynamic programming-based systems, that the global information of sequences cannot be incorporated as features of the models. Another reason may be that the computational complexity of the models prevented the developers to invent effective features for the task. We had to wait until Tsai et al. (2006), who combine pattern-based post-processing with CRFs, for CRF-based systems to achieve the same level of performance as Zhou et al. As such, a key to further improvement of the performance of bio-entity recognition has been to employ global features, which are effective to capture the features of long names appearing in the bio domain.

In this paper, we use reranking architecture, which was successfully applied to the task of natural language parsing (Collins, 2000; Charniak and

BioNLP 2007: Biological, translational, and clinical language processing, pages 209–216,
Prague, June 2007.

Johnson, 2005), to address the problem. Reranking enables us to incorporate truly global features to the model of named entity tagging, and we aim to realize the state-of-the-art performance without depending on rule-based post-processes.

Use of global features in named-entity recognition systems is widely studied for sequence labeling including general named-entity tasks like CoNLL 2003 shared task. Such systems may be classified into two kinds, one of them uses a single classifier which is optimized incorporating non-local features, and the other consists of pipeline of more than one classifiers. The former includes Relational Markov Networks by Bunescu et al. (2004) and skip-edge CRFs by Sutton et al. (2004). A major drawback of this kind of systems may be heavy computational cost of inference both for training and running the systems, because non-local dependency forces such models to use expensive approximate inference instead of dynamic-programming-based exact inference. The latter, pipelined systems include a recent study by Krishnan et al. (2006), as well as our reranking system. Their method is a two stage model of CRFs, where the second CRF uses the global information of the output of the first CRF. Though their method is effective in capturing various non-local dependencies of named entities like consistency of labels, we may be allowed to claim that reranking is likely to be more effective in bio-entity tagging, where the treatment of long entity names is also a problem.

This paper is organized as follows. First, we briefly overview the JNLPBA shared task of bio-entity recognition and its related work. Then we explain the components of our system, one of which is an MEMM n-best tagger, and the other is a reranker based on log-linear models. Then we show the experiments to tune the performance of the system using the development set. Finally, we compare our results with the existing systems, and conclude the paper with the discussion for further improvement of the system.

2 JNLPBA shared task and related work

This section overviews the task of biomedical named entity recognition as presented in JNLPBA shared task held at COLING 2004, and the systems that were successfully applied to the task. The training data provided by the shared task consisted of 2000 abstracts of biomedical articles taken from the GENIA corpus version 3 (Ohta et al., 2002), which consists of the MEDLINE abstracts with publication years from 1990 to 1999. The articles are annotated with named-entity BIO tags as an example shown in Table 1. As usual, 'B' and 'I' tags are for beginning and internal words of named entities, and 'O' tags are for general English words that are not named entities. 'B' and 'I' tags are split into 5 sub-labels, each of which are used to represent proteins, genes, cell lines, DNAs, cell types, and RNAs. The test set of the shared task consists of 404 MEDLINE abstracts whose publication years range from 1978 to 2001. The difference of publication years between the training and test sets reflects the organizer's intention to see the entity recognizers' portability with regard to the differences of the articles' publication years.

Kim et al. (Kim et al., 2004) compare the 8 systems participated in the shared task. The systems use various classification models including CRFs, hidden Markov models (HMMs), support vector machines (SVMs), and MEMMs, with various features and external resources. Though it is impossible to observe clear correlation between the performance and classification models or resources used, an important characteristic of the best system by Zhou et al. (2004) seems to be extensive use of rule-based post processing they apply to the output of their classifier.

After the shared task, several researchers tackled the problem using the CRFs and their extensions. Okanohara et al. (2006) applied semi-CRFs (Sarawagi and Cohen, 2004), which can treat multiple words as corresponding to a single state. Friedrich et al. (2006) used CRFs with features from the external gazetteer. Current state-of-the-art for the shared-task is achieved by Tsai et al. (2006), whose improvement depends on careful design of features including the normalization of numeric expressions, and use of post-processing by automatically extracted patterns.

IL-2	gene	expression	requires	reactive	oxygen	production	by	5-lipoxygenase	.
B-DNA	I-DNA	O	O	O	O	O	O	B-protein	O

Figure 1: Example sentence from the training data.

State name	Possible next state
BOS	B-* or O
B-protein	I-protein, B-* or O
B-cell_type	I-cell_type, B-* or O
B-DNA	I-DNA, B-* or O
B-cell_line	I-cell_line, B-* or O
B-RNA	I-RNA, B-* or O
I-protein	I-protein, B-* or O
I-cell_type	I-cell_type, B-* or O
I-DNA	I-DNA, B-* or O
I-cell_line	I-cell_line, B-* or O
I-RNA	I-RNA, B-* or O
O	B-* or O

Table 1: State transition of MEMM.

3 N-best MEMM tagger

As our n-best tagger, we use a first order MEMM model (McCallum et al., 2000). Though CRFs (Lafferty et al., 2001) can be regarded as improved version of MEMMs, we have chosen MEMMs because MEMMs are usually much faster to train compared to CRFs, which enables extensive feature selection. Training a CRF tagger with features selected using an MEMM may result in yet another performance boost, but in this paper we concentrate on the MEMM as our n-best tagger, and consider CRFs as one of our future extensions.

Table 1 shows the state transition table of our MEMM model. Though existing studies suggest that changing the tag set of the original corpus, such as splitting of O tags, can contribute to the performances of named entity recognizers (Peshkin and Pfefer, 2003), our system uses the original tagset of the training data, except that the 'BOS' label is added to represent the state before the beginning of sentences.

Probability of state transition to the i-th label of a sentence is calculated by the following formula:

$$P(l_i|l_{i-1}, S) = \frac{exp(\sum_j \lambda_j f_j(l_i, l_{i-1}, S))}{\sum_l exp(\sum_j \lambda_j f_j(l, l_{i-1}, S))}. \quad (1)$$

Features used	Forward tagging	Backward tagging
unigrams, bigrams and previous labels	(62.43/71.77/66.78)	(66.02/74.73/70.10)
unigrams and bigrams	(61.64/71.73/66.30)	(65.38/74.87/69.80)
unigrams and previous labels	(62.17/71.67/66.58)	(65.59/74.77/69.88)
unigrams	(61.31/71.81/66.15)	(65.61/75.25/70.10)

Table 2: (Recall/Precision/F-score) of forward and backward tagging.

where l_i is the next BIO tag, l_{i-1} is the previous BIO tag, S is the target sentence, and f_j and l_j are feature functions and parameters of a log-linear model (Berger et al., 1996). As a first order MEMM, the probability of a label l_i is dependent on the previous label l_{i-1}, and when we calculate the normalization constant in the right hand side (i.e. the denominator of the fraction), we limit the range of l to the possible successors of the previous label. This probability is multiplied to obtain the probability of a label sequence for a sentence:

$$P(l_{1...n}|S) = \prod_i P(l_i|l_{i-1}). \quad (2)$$

The probability in Eq. 1. is estimated as a single log-linear model, regardless to the types of the target labels.

N-best tag sequences of input sentences are obtained by well-known combination of the Viterbi algorithm and A* algorithm. We implemented two methods for thresholding the best sequences: N-*best* takes the sequences whose ranks are higher than N, and θ-*best* takes the sequences that have probability higher than that of the best sequences with a factor θ, where θ is a real value between 0 and 1. The θ-best method is used in combination with N-best to limit the maximum number of selected sequences.

3.1 Backward tagging

There remains one significant choice when we develop an MEMM tagger, that is, the direction of tagging. The results of the preliminary experiment with forward and backward MEMMs with word unigram and bigram features are shown in Table 2. (The evaluation is done using the same training and development set as used in Section 5.) As can be seen, the backward tagging outperformed forward tagging by a margin larger than 3 points, in all the cases.

One of the reasons of these striking differences may be long names which appear in biomedical texts. In order to recognize long entity names, forward tagging is preferable if we have strong clues of entities which appear around their left boundaries, and backward tagging is preferable if clues appear at right boundaries. A common example of this effect is a gene expression like 'XXX YYY gene.' The right boundary of this expression is easy to detect because of the word 'gene.' For a backward tagger, the remaining decision is only 'where to stop' the entity. But a forward tagger must decide not only 'where to start,' but also 'whether to start' the entity, before the tagger encounter the word 'gene.' In biomedical named-entity tagging, right boundaries are usually easier to detect, and it may be the reason of the superiority of the backward tagging.

We could have partially alleviated this effect by employing head-word triggers as done in Zhou et al. (2004), but we decided to use backward tagging because the results of a number of preliminary experiments, including the ones shown in Table 2 above, seemed to be showing that the backward tagging is preferable in this task setting.

3.2 Feature set

In our system, features of log-linear models are generated by concatenating (or combining) the 'atomic' features, which belong to their corresponding atomic feature classes. Feature selection is done by deciding whether to include combination of feature classes into the model. We ensure that features in the same atomic feature class do not co-occur, so that a single feature-class combination generates only one feature for each event. The following is a list of atomic feature classes implemented in our system.

Label features The target and previous labels. We also include the coarse-grained label distinction to distinguish five 'I' labels of each entity classes from the other labels, expecting smoothing effect.

Word-based features Surface strings, base forms, parts-of-speech (POSs), word shapes[1], suffixes and prefixes of words in input sentence. These features are extracted from five words around the word to be tagged, and also from the words around NP-chunk boundaries as explained bellow.

Chunk-based features Features dependent on the output of shallow parser. Word-based features of the beginning and end of noun phrases, and the distances of the target word from the beginning and end of noun phrases are used.

4 Reranker

Our reranker is based on a log-linear classifier. Given n-best tag sequences $L_i (1 \leq i \leq n)$, a log-linear model is used to estimate the probability

$$P(L_i|S) = \frac{exp(\sum_j \lambda_j f_j(L_i, S))}{\sum_k exp(\sum_j \lambda_j f_j(L_k, S))}. \qquad (3)$$

From the n-best sequences, reranker selects a sequence which maximize this probability.

The features used by the reranker are explained in the following sections. Though most of the features are binary-valued (i.e. the value of f_j in Eq. 3. is exclusively 1 or 0), the logarithm of the probability of the sequence output by the n-best tagger is also used as a real-valued feature, to ensure the reranker's improvement over the n-best tagger.

4.1 Basic features

Basic features of the reranker are straightforward extension of the features used in the MEMM tagger. The difference is that we do not have to care the locality of the features with regard to the labels.

Characteristics of words that are listed as word-based features in the previous section is also used for the reranker. Such features are chiefly extracted from around the left and right boundaries of entities. In our experiments, we used five words around the leftmost and rightmost words of the entities. We also use the entire string, affixes, word shape, concatenation of POSs, and length of entities. Some of our

[1]The shape of a word is defined as a sequence of character types contained in the word. Character types include uppercase letters, lowercase letters, numerics, space characters, and the other symbols.

features depend on two adjacent entities. Such features include the word-based features of the words between the entities, and the verbs between the entities. Most of the features are used in combination with entity types.

4.2 N-best distribution features

N-best tags of sentences other than the target sentence is available to the rerankers. This information is sometimes useful for recognizing the names in the target sentence. For example, proteins are often written as 'XXX protein' where XXX is a protein name, especially when they are first introduced in an article, and thereafter referred to simply as 'XXX.' In such cases, the first appearance is easily identified as proteins only by local features, but the subsequent ones might not, and the information of the first appearance can be effectively used to identify the other appearances.

Our system uses the distribution of the tags of the 20 neighboring sentences of the target sentence to help the tagging of the target sentence. Tag distributions are obtained by marginalizing the n-best tag sequences. Example of an effective feature is a binary-valued feature which becomes 1 when the candidate entity names in the target sentence is contained in the marginal distribution of the neighboring sentences with a probability which is above some threshold.

We also use the information of overlapping named-entity candidates which appear in the target sentence. When there is an overlap between the entities in the target sequence and any of the named-entity candidates in the marginal distribution of the target sentence, the corresponding features are used to indicate the existence of the overlapping entity and its entity type.

5 Experiments

We evaluated the performance of the system on the data set provided by the COLING 2004 JNLPBA shared-task. which consists of 2000 abstracts from the MEDLINE articles. GENIA tagger [2], a biomedical text processing tool which automatically annotates POS tags, shallow parses and named-entity tags is used to preprocess the corpus, and POS and shallow parse information is used in our experiments.

Features used	(Recall/Precision/F-score)
full set	(73.90/77.58/75.69)
w/o shallow parser	(72.63/76.35/74.44)
w/o previous labels	(72.06/75.38/73.68)

Table 3: Performance of MEMM tagger.

We divided the data into 20 contiguous and equally-sized sections, and used the first 18 sections for training, and the last 2 sections for testing while development (henceforth the training and development sets, respectively). The training data of the reranker is created by the n-best tagger, and every set of 17 sections from the training set is used to train the n-best tagger for the remaining section (The same technique is used by previous studies to avoid the n-best tagger's 'unrealistically good' performance on the training set (Collins, 2000)). Among the n-best sequences output by the MEMM tagger, the sequence with the highest F-score is used as the 'correct' sequence for training the reranker.

The two log-linear models for the MEMM tagger and reranker are estimated using a limited-memory BFGS algorithm implemented in an open-source software Amis[3]. In both models, Gaussian prior distributions are used to avoid overfitting (Chen and Rosenfeld, 1999), and the standard deviations of the Gaussian distributions are optimized to maximize the performance on the development set. We also used a thresholding technique which discards features with low frequency. This is also optimized using the development set, and the best threshold was 4 for the MEMM tagger, and 50 for the reranker [4]. For both of the MEMM tagger and reranker, combinations of feature classes are manually selected to improve the accuracies on the development set. Our final models include 49 and 148 feature class combinations for the MEMM tagger and reranker, respectively.

Table 3 shows the performance of the MEMM tagger on the development set. As reported in many

[2]http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/. The tagger is trained on the GENIA corpus, so it is likely to show very good performance on both training and development sets, but not on the test set.

[3]http://www-tsujii.is.s.u-tokyo.ac.jp/amis/.

[4]We treated feature occurrences both in positive and negative examples as one occurrence.

Features used	(Recall/Precision/F-score)
oracle	(94.62/96.07/95.34)
full set	(75.46/78.85/77.12)
w/o features that depend on two entities	(74.67/77.99/76.29)
w/o n-best distribution features	(74.99/78.38/76.65)
baseline	(73.90/77.58/75.69)

Table 4: Performance of the reranker.

of the previous studies (Kim et al., 2004; Okanohara et al., 2006; Tzong-Han Tsai et al., 2006), features of shallow parsers had a large contribution to the performance. The information of the previous labels was also quite effective, which indicates that label unigram models (i.e. 0th order Markov models, so to speak) would have been insufficient for good performance.

Then we developed the reranker, using the results of 50-best taggers as training data. Table 4 shows the performance of the reranker pipelined with the 50-best MEMM tagger, where the 'oracle' row shows the upper bound of reranker performance. Here, we can observe that the reranker successfully improved the performance by 1.43 points from the baseline (i.e. the one-best of the MEMM tagger). It is also shown that the global features that depend on two adjacent entities, and the n-best distribution features from the outside of the target sentences, are both contributing to this performance improvement.

We also conducted experimental comparison of two thresholding methods which are described in Section 3. Since we can train and test the reranker with MEMM taggers that use different thresholding methods, we could make a table of the performance of the reranker, changing the MEMM tagger used for both training and evaluation[5].

Tables 5 and 6 show the F-scores obtained by various MEMM taggers, where the 'oracle' column again shows the performance upper bound. (All of the θ-best methods are combined with 200-best thresholding.) Though we can roughly state that the reranker can work better with n-best taggers which are more ambiguous than those used for their training, the differences are so slight to see clear tendencies (For example, the columns for the reranker trained using the 10-best MEMM tagger seems to be a counter example against the statement).

We may also be able to say that the θ-best methods are generally performing slightly better, and it could be explained by the fact that we have better oracle performance with less ambiguity in θ-best methods.

However, the scores in the column corresponding to the 50-best training seems to be as high as any of the scores of the θ-best methods, and the best score is also achieved in that column. The reason may be because our performance tuning is done exclusively using the 50-best-trained reranker. Though we could have achieved better performance by doing feature selection and hyper-parameter tuning again using θ-best MEMMs, we use the reranker trained on 50-best tags run with 70-best MEMM tagger as the best performing system in the following.

5.1 Comparison with existing systems

Table 7 shows the performance of our n-best tagger and reranker on the official test set, and the best reported results on the same task. As naturally expected, our system outperformed the systems that cannot accommodate truly global features (Note that one point of F-score improvement is valuable in this task, because inter-annotator agreement rate of human experts in bio-entity recognition is likely to be about 80%. For example, Krauthammer et al. (2004) report the inter-annotater agreement rate of 77.6% for the three way bio-entity classification task.) and the performance can be said to be at the same level as the best systems. However, in spite of our effort, our system could not outperform the best result achieved by Tsai et al. What makes Tsai et al.'s system perform better than ours might be the careful treatment of numeric expressions.

It is also notable that our MEMM tagger scored 71.10, which is comparable to the results of the systems that use CRFs. Considering the fact that the tagger's architecture is a simple first-order MEMM which is far from state-of-the-art, and it uses only POS taggers and shallow parsers as external resources, we can say that simple machine-learning-based method with carefully selected features could

[5]These results might not be a fair comparison, because the feature selection and hyper-parameter tuning are done using a reranker which is trained and tested with a 50-best tagger.

			Thresholding method for training						
Thresholding method for testing	oracle	avg. # of answers	10-best	20-best	30-best	40-best	50-best	70-best	100-best
10-best	91.00	10	76.51	76.53	76.85	76.73	77.01	76.68	76.86
20-best	93.31	20	76.40	76.55	76.83	76.62	76.95	76.68	76.85
30-best	94.40	30	76.34	76.52	76.91	76.63	77.06	76.75	76.90
40-best	94.94	40	76.39	76.58	76.91	76.71	77.14	76.75	76.92
50-best	95.34	50	76.37	76.58	76.90	76.65	77.12	76.78	76.92
70-best	95.87	60	76.38	76.57	76.91	76.71	**77.16**	76.81	76.97
100-best	96.26	70	76.38	76.59	76.95	76.74	77.10	76.82	76.98

Table 5: Comparison of the F-scores of rerankers trained and evaluated with various N-best taggers.

			Thresholding method for training						
Thresholding method for testing	oracle	avg. # of answers	0.05-best	0.02-best	0.008-best	0.004-best	0.002-best	0.0005-best	0.0002-best
0.05-best	91.65	10.7	76.70	76.80	76.93	76.64	77.02	76.78	76.52
0.02-best	93.45	17.7	76.79	76.91	77.07	76.79	77.09	76.89	76.70
0.008-best	94.81	27.7	76.79	77.01	77.05	76.80	**77.14**	76.88	76.73
0.004-best	95.55	37.5	76.79	76.98	76.97	76.74	77.12	76.86	76.71
0.002-best	96.09	49.3	76.79	76.98	76.96	76.73	77.13	76.85	76.72
0.0005-best	96.82	77.7	76.79	76.98	76.96	76.73	77.13	76.85	76.70
0.0002-best	97.04	99.2	76.83	77.01	76.96	76.71	77.13	76.88	76.70

Table 6: Comparison of the F-scores of rerankers trained and evaluated with various θ-best taggers.

	F-score	Method
This paper	71.10	MEMM
	72.65	reranking
Tsai et al. (2006)	72.98	CRF, post-processing
Zhou et al. (2004)	72.55	HMM, SVM, post-processing, gazetteer
Friedrich et al. (2006)	71.5	CRF, gazetteer
Okanohara et al. (2006)	71.48	semi-CRF

Table 7: Performance comparison on the test set.

be sufficient practical solutions for this kind of tasks.

6 Conclusion

This paper showed that the named-entity recognition, which have usually been solved by dynamic-programming-based sequence-labeling techniques with local features, can have innegligible performance improvement from reranking methods. Our system showed clear improvement over many of the machine-learning-based systems reported to date, and also proved comparable to the existing state-of-the-art systems that use rule-based post-processing.

Our future plans include further sophistication of features, such as the use of external gazetteers which is reported to improve the F-score by 1.0 and 2.7 points in (Zhou and Su, 2004) and (Friedrich et al., 2006), respectively. We expect that reranking architecture can readily accommodate dictionary-based features, because we can apply elaborated string-matching algorithms to the qualified candidate strings available at reranking phase.

We also plan to apply self-training of n-best tagger which successfully boosted the performance of one of the best existing English syntactic parser (McClosky et al., 2006). Since the test data of the shared-task consists of articles that represent the different publication years, the effects of the publication years of the texts used for self-training would be interesting to study.

References

Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1).

R. Bunescu and R. Mooney. 2004. Relational markov networks for collective information extraction. In *Proceedings of ICML 2004*.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In *Proceedings of ACL 2005*.

S. Chen and R. Rosenfeld. 1999. A Gaussian prior for smoothing maximum entropy models. In *Technical Report CMUCS*.

Michael Collins. 2000. Discriminative Reranking for Natural Language Parsing. In *Proceedings of 17th International Conference on Machine Learning*, pages 175–182. Morgan Kaufmann, San Francisco, CA.

Christoph M. Friedrich, Thomas Revillion, Martin Hofmann, and Juliane Fluck. 2006. Biomedical and Chemical Named Entity Recognition with Conditional Random Fields: The Advantage of Dictionary Features. In *Proceedings of the Second International Symposium on Semantic Mining in Biomedicine*.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the Bio-Entity Recognition Task at JNLPBA. In *Proceedings of the International Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-04)*, pages 70–75, Geneva, Switzerland.

Michael Krauthammer and Goran Nenadic. 2004. Term identification in the biomedical literature. *Journal of Biomedical Informatics*, 37(6).

Vijay Krishnan and Christopher D. Manning. 2006. An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition. In *Proceedings of ACL 2006*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of 18th International Conference on Machine Learning*, pages 282–289.

Andrew McCallum, Dayne Freitag, and Fernando Pereira. 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *ICML 2000*.

D. McClosky, E. Charniak, and M. Johnson. 2006. Effective self-training for parsing. In *Proceedings of NAACL 2006*.

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 2002)*, March.

Daisuke Okanohara, Yusuke Miyao, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2006. Improving the Scalability of Semi-Markov Conditional Random Fields for Named Entity Recognition. In *Proceedings of ACL 2006*, Sydney, Australia, July.

Leonid Peshkin and Avi Pfefer. 2003. Bayesian Information Extraction Network. In *Proceedings of the Eighteenth International Joint Conf. on Artificial Intelligence*.

S. Sarawagi and W. Cohen. 2004. Semimarkov conditional random fields for information extraction. In *Proceedings of ICML 2004*.

Charles Sutton and Andrew McCallum. 2004. Collective Segmentation and Labeling of Distant Entities in Information Extraction. Technical report, University of Massachusetts. Presented at ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields.

Richard Tzong-Han Tsai, Cheng-Lung Sung, Hong-Jie Dai, Hsieh-Chuan Hung, Ting-Yi Sung, and Wen-Lian Hsu. 2006. NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition. In *BMC Bioinformatics 2006, 7(Suppl 5):S11*.

GuoDong Zhou and Jian Su. 2004. Exploring deep knowledge resources in biomedical name recognition. In *Proceedings of the International Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-04)*, pages 96–99.