How to Make the Most of NE Dictionaries in Statistical NER

Yutaka Sasaki[2] **Yoshimasa Tsuruoka**[2] **John McNaught**[1,2] **Sophia Ananiadou**[1,2]
[1] National Centre for Text Mining
[2] School of Computer Science, University of Manchester
MIB, 131 Princess Street, Manchester, M1 7DN, UK

Abstract

When term ambiguity and variability are very high, dictionary-based *Named Entity Recognition (NER)* is not an ideal solution even though large-scale terminological resources are available. Many researches on statistical NER have tried to cope with these problems. However, it is not straightforward how to exploit existing and additional *Named Entity (NE)* dictionaries in statistical NER. Presumably, addition of NEs to an NE dictionary leads to better performance. However, in reality, the retraining of NER models is required to achieve this. We have established a novel way to improve the NER performance by addition of NEs to an NE dictionary without retraining. We chose protein name recognition as a case study because it most suffers the problems related to heavy term variation and ambiguity. In our approach, first, known NEs are identified in parallel with *Part-of-Speech (POS)* tagging based on a general word dictionary and an NE dictionary. Then, statistical NER is trained on the *tagger outputs* with correct NE labels attached. We evaluated performance of our NER on the standard JNLPBA-2004 data set. The F-score on the test set has been improved from 73.14 to 73.78 after adding the protein names appearing in the training data to the POS tagger dictionary without any model retraining. The performance further increased to 78.72 after enriching the tagging dictionary with test set protein names. Our approach has demonstrated high performance in protein name recognition, which indicates how to make the most of known NEs in statistical NER.

1 Introduction

The accumulation of online biomedical information has been growing at a rapid pace, mainly attributed to a rapid growth of a wide range of repositories of biomedical data and literature. The automatic construction and update of scientific *knowledge bases* is a major research topic in Bioinformatics. One way of populating these knowledge bases is through *named entity recognition (NER)*. Unfortunately, biomedical NER faces many problems, e.g., protein names are extremely difficult to recognize due to ambiguity, complexity and variability. A further problem in protein name recognition arises at the tokenization stage. Some protein names include punctuation or special symbols, which may cause tokenization to lose some word concatenation information in the original sentence. For example, `IL-2` and `IL - 2` fall into the same token sequence `IL - 2` as usually dash (or hyphen) is designated as a token delimiter.

Research into NER is centred around three approaches: dictionary-based, rule-based and machine learning-based approaches. To overcome the usual NER pitfalls, we have opted for a hybrid approach combining dictionary-based and machine learning approaches, which we call *dictionary-based statistical NER approach*. After identifying protein names in text, we link these to semantic identifiers, such as UniProt accession numbers. In this paper, we focus on the evaluation of our dictionary-based statistical NER.

2 Methods

Our dictionary-based statistical approach consists of two components: dictionary-based POS/PROTEIN tagging and statistical sequential labelling. First,

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 63–70,
Columbus, Ohio, USA, June 2008.

dictionary-based POS/PROTEIN tagging finds candidates for protein names using a dictionary. The dictionary maps strings to parts of speech (POS), where the POS tagset is augmented with a tag NN-PROTEIN. Then, sequential labelling applies to reduce false positives and false negatives in the POS/PROTEIN tagging results. Expandability is supported through allowing a user of the NER tool to improve NER coverage by adding entries to the dictionary. In our approach, retraining is not required after dictionary enrichment.

Recently, *Conditional Random Fields (CRFs)* have been successfully applied to sequence labelling problems, such as POS tagging and NER, and have outperformed other machine learning techniques. The main idea of CRFs is to estimate a conditional probability distribution over label sequences, rather than over local directed label sequences as with Hidden Markov Models (Baum and Petrie, 1966) and Maximum Entropy Markov Models (McCallum et al., 2000). Parameters of CRFs can be efficiently estimated through the log-likelihood parameter estimation using the forward-backward algorithm, a dynamic programming method.

2.1 Training and test data

Experiments were conducted using the training and test sets of the JNLPBA-2004 data set(Kim et al., 2004).

Training data The training data set used in JNLPBA-2004 is a set of tokenized sentences with manually annotated term class labels. The sentences are taken from the Genia corpus (version 3.02) (Kim et al., 2003), in which 2,000 abstracts were manually annotated by a biologist, drawing on a set of POS tags and 36 biomedical term classes. In the JNLPBA-2004 shared task, performance in extracting five term classes, i.e., protein, DNA, RNA, cell line, and cell type classes, were evaluated.

Test Data The test data set used in JNLPBA-2004 is a set of tokenized sentences extracted from 404 separately collected MEDLINE abstracts, where the term class labels were manually assigned, following the annotation specification of the Genia corpus.

2.2 Overview of dictionary-based statistical NER

Figure 1 shows the block diagram of dictionary-based statistical NER. Raw text is analyzed by a POS/PROTEIN tagger based on a CRF tagging model and dictionary, and then converted into token sequences. Strings in the text that match with protein names in the dictionary will be tagged as NN-PROTEIN depending on the context around the protein names. Since it is not realistic to enumerate all protein names in the dictionary, due to their high variability of form, instead previously unseen forms are predicted to be protein names by statistical sequential labelling. Finally, protein names are identified from the POS/PROTEIN tagged token sequences via a CRF labelling model.

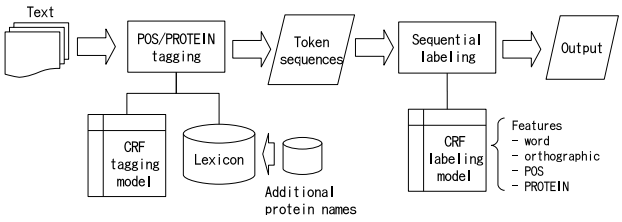

Figure 1: Block diagram of dictionary-based statistical NER

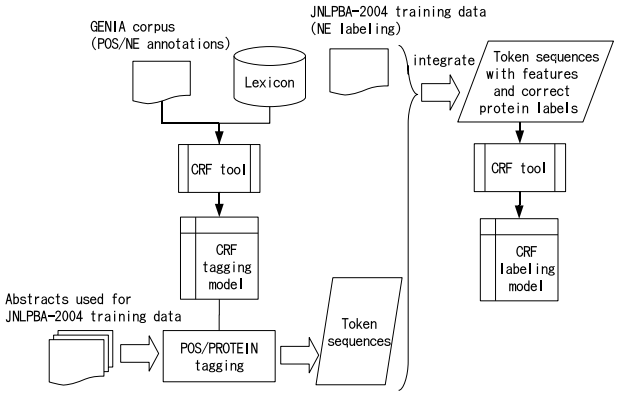

Figure 2: Block diagram of training procedure

Figure 2 shows the block diagram of the training procedure for both POS/PROTEIN tagging and sequential labelling. The tagging model is created using the Genia corpus (version 3.02) and a dictionary. Using the tagging model, MEDLINE abstracts used for the JNLPBA-2004 training data set are then POS/PROTEIN-tagged. The output token sequences over these abstracts are then integrated with the correct protein labels of the JNLPBA-2004 training data. This process results in the preparation of token sequences with features and correct protein labels. A CRF labelling model is finally generated by applying a CRF tool to these decorated token sequences.

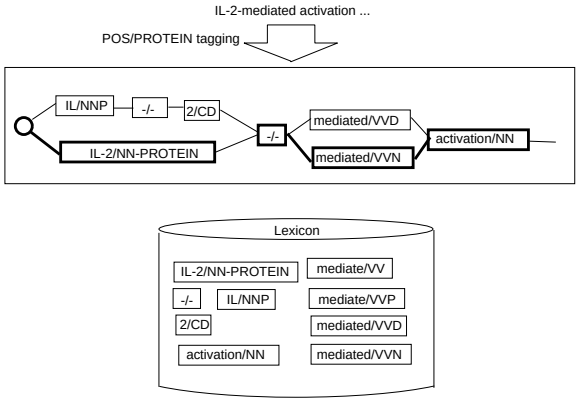

Figure 3: Dictionary based approach

2.2.1 Dictionary-based POS/PROTEIN tagging

The dictionary-based approach is beneficial when a sentence contains some protein names that conflict with general English words. Otherwise, if the POS tags of sentences are decided without considering possible occurrences of protein names, POS sequences could be disrupted. For example, in "met proto-oncogene precursor", *met* might be falsely recognized as a verb by a non dictionary-based tagger.

Given a sentence, the dictionary-based approach extracts protein names as follows. Find all word sequences that match the lexical entries, and create a token graph (i.e., *trellis*) according to the word order. Estimate the score of every path using the weights of node and edges estimated by training using Conditional Random Fields. Select the best path.

Figure 3 shows an example of our dictionary-based approach. Suppose that the input is "IL-2-mediated activation". A trellis is created based on the lexical entries in a dictionary. The selection criteria for the best path are determined by the CRF tagging model trained on the Genia corpus. In this example, `IL-2/NN-PROTEIN -/- mediated/VVN activation/NN` is selected as the best path. Following Kudo et al. (Kudo et al., 2004), we adapted the core engine of the CRF-based morphological analyzer, MeCab[1], to our POS/PROTEIN tagging task. MeCab's dictionary databases employ double arrays (Aoe, 1989) which enable efficient lexical look-ups.

The features used were:

- POS
- PROTEIN
- POS-PROTEIN
- bigram of adjacent POS
- bigram of adjacent PROTEIN
- bigram of adjacent POS-PROTEIN

During the construction of the trellis, white space is considered as the delimiter unless otherwise stated within dictionary entries. This means that unknown tokens are character sequences without spaces.

2.2.2 Dictionary construction

A dictionary-based approach requires the dictionary to cover not only a wide variety of biomedical terms but also entries with:

- all possible capitalization
- all possible linguistic inflections

We constructed a freely available, wide-coverage English word dictionary that satisfies these conditions. We did consider the MedPost pos-tagger package[2] which contains a free dictionary that has downcased English words; however, this dictionary is not well curated as a dictionary and the number of entries is limited to only 100,000, including inflections.

Therefore, we started by constructing an English word dictionary. Eventually, we created a dictionary with about 266,000 entries for English words (systematically covering inflections) and about 1.3 million entries for protein names.

We created the general English part of the dictionary from WordNet by semi-automatically adding POS tags. The POS tag set is a minor modification of the Penn Treebank POS tag set[3], in that protein names are given a new POS tag, NN-PROTEIN. Further details on construction of the dictionary now follow.

Protein names were extracted from the BioThesaurus[4]. After selecting only those terms clearly stated as protein names, 1,341,992 protein names in total were added to the dictionary.

[1] http://sourceforge.net/project/showfiles.php?group_id=177856/`biothesaurus/`

[2] `ftp://ftp.ncbi.nlm.nih.gov/pub/lsmith/MedPost/`

[3] `ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz`

[4] `http://pir.georgetown.edu/iprolink/`

Nouns were extracted from WordNet's noun list. Words starting with lower case and upper case letters were determined as NN and NNP, respectively. Nouns in NNS and NNPS categories were collected from the results of POS tagging articles from Plos Biology Journal[5] with TreeTagger[6].

Verbs were extracted from WordNet's verb list. We manually curated VBD, VBN, VBG and VBZ verbs with irregular inflections based on WordNet. Next, VBN, VBD, VBG and VBZ forms of regular verbs were automatically generated from the WordNet verb list.

Adjectives were extracted from WordNet's adjective list. We manually curated JJ, JJR and JJS of irregular inflections of adjectives based on the WordNet irregular adjective list. Base form (JJ) and regular inflections (JJR, JJS) of adjectives were also created based on the list of adjectives.

Adverbs were extracted from WordNet's adverb list. Both the original and capitalised forms were added as RB.

Pronouns were manually curated. PRP and PRP$ words were added to the dictionary.

Wh-words were manually curated. As a result, WDT, WP, WP$ and WRB words were added to the dictionary.

Words for other parts of speech were manually curated.

2.2.3 Statistical prediction of protein names

Statistical sequential labelling was employed to improve the coverage of protein name recognition and to remove false positives resulting from the previous stage (dictionary-based tagging).

We used the JNLPBA-2004 training data, which is a set of tokenized word sequences with IOB2(Tjong Kim Sang and Veenstra, 1999) protein labels. As shown in Figure 2, POSs of tokens resulting from tagging and tokens of the JNLPBA-2004 data set are integrated to yield training data for sequential labelling. During integration, when the single token of a protein name found after tagging corresponds to a sequence of tokens from JNLPBA-2004, its POS is given as NN-PROTEIN1, NN-PROTEIN2,..., according to the corresponding token order in the JNLPBA-2004 sequence.

Following the data format of the JNLPBA-2004 training set, our training and test data use the IOB2 labels, which are "B-protein" for the first token of the target sequence, "I-protein" for each remaining token in the target sequence, and "O" for other tokens. For example, "Activation of the IL 2 precursor provides" is analyzed by the POS/PROTEIN tagger as follows.

```
Activation      NN
of              IN
the             DT
IL 2 precursor  NN-PROTEIN
provides        VVZ
```

The tagger output is given IOB2 labels as follows.

```
Activation NN          O
of         IN          O
the        DT          O
IL         NN-PROTEIN1 B-protein
2          NN-PROTEIN2 I-protein
precursor  NN-PROTEIN3 I-protein
provides   VVZ         O
```

We used CRF models to predict the IOB2 labels. The following features were used in our experiments.

- word feature
- orthographic features
  - the first letter and the last four letters of the word form, in which capital letters in a word are normalized to "A", lower case letters are normalized to "a", and digits are replaced by "0", *e.g.*, the word form of IL-2 is AA-0.
  - postfixes, the last two and four letters
- POS feature
- PROTEIN feature

The window size was set to ± 2 of the current token.

3 Results and discussion

[5] http://biology.plosjournals.org/

[6] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html/

Table 1: Experimental Rusults

Tagging		R	P	F
(a) POS/PROTEIN tagging	Full	52.91	43.85	47.96
	Left	61.48	50.95	55.72
	Right	61.38	50.87	55.63
Sequential Labelling		**R**	**P**	**F**
(b) Word feature	Full	63.23	70.39	66.62
	Left	68.15	75.86	71.80
	Right	69.88	77.79	73.63
(c) (b) + orthographic feature	Full	77.17	67.52	72.02
	Left	82.51	72.20	77.01
	Right	84.29	73.75	78.67
(d) (c) + POS feature	Full	76.46	68.41	72.21
	Left	81.94	73.32	77.39
	Right	83.54	74.75	78.90
(e) (d) + PROTEIN feature	Full	77.58	69.18	73.14
	Left	82.69	73.74	77.96
	Right	84.37	75.24	79.54
(f) (e) + after adding protein names in the training set to the dictionary	Full	**79.85**	**68.58**	**73.78**
	Left	84.82	72.85	78.38
	Right	86.60	74.37	80.02

3.1 Protein name recognition performance

Table 1 shows our protein name recognition results, showing the differential effect of various combinations of strategies. Results are expressed according to recall (R), precision (P), and F-measure (F), which here measure how accurately our various experiments determined the left boundary (Left), the right boundary (Right), and both boundaries (Full) of protein names. The baseline for tagging (row (a)) shows the protein name detection performance of our dictionary-based tagging using our large protein name dictionary, where no training for protein name prediction was involved. The F-score of this baseline tagging method was 47.96.

The baseline for sequential labelling (row (b)) shows the prediction performance when using only word features where no orthographic and POS features were used. The F-score of the baseline labelling method was 66.62. When orthographic feature was added (row (c)), the F-score increased by 5.40 to 72.02. When the POS feature was added (row (d)), the F-score increased by 0.19 to 72.21. Using all features (row (e)), the F-score reached 73.14. Surprisingly, adding protein names appearing in the *training data* to the dictionary further improved the F-score by 0.64 to 73.78, which is the second best score for protein name recognition using the JNLPBA-2004 data set.

Table 2: After Dictionary Enrichment

Method		R	P	F
Tagging (+test set protein names)	Full	79.02	61.87	69.40
	Left	82.28	64.42	72.26
	Right	80.96	63.38	71.10
Labelling (+test set protein names)	full	**86.13**	**72.49**	**78.72**
	Left	89.58	75.40	81.88
	Right	90.23	75.95	82.47

Tagging and labelling speeds were measured using an unloaded Linux server with quad 1.8 GHz Opteron cores and 16GB memory. The dictionary-based POS/PROTEIN tagger is very fast even though the total size of the dictionary is more than one million. The processing speed for tagging and sequential labelling of the 4,259 sentences of the test set data took 0.3 sec and 7.3 sec, respectively, which means that in total it took 7.6 sec. for recognizing protein names in the plain text of 4,259 sentences.

3.2 Dictionary enrichment

The advantage of the dictionary-based statistical approach is that it is versatile, as the user can easily improve its performance with no retraining. We assume the following situation as the ideal case: suppose that a user needs to analyze a large amount of text with protein names. The user wants to know

the maximum performance achievable for identifying protein names with our dictionary-based statistical recognizer which can be achieved by adding more protein names to the current dictionary. Note that protein names should be identified in context. That is, recall of the NER results with the ideal dictionary is not 100%. Some protein names in the ideal dictionary are dropped during statistical tagging or labelling.

Table 2 shows the scores after each step of dictionary enrichment. The first block (Tagging) shows the tagging performance after adding protein names appearing in the *test set* to the dictionary. The second block (Labelling) shows the performance of the sequence labelling of the output of the first step. Note that tagging and the sequence labelling models are not retrained using the test set.

3.3 Discussion

It is not possible in reality to train the recognizer on target data, *i.e.*, the test set, but it would be possible for users to add discovered protein names to the dictionary so that they could improve the overall performance of the recognizer without retraining.

Rule-based and procedural approaches are taken in (Fukuda et al., 1998; Franzen et al., 2002). Machine learning-based approaches are taken in (Collier et al., 2000; Lee et al., 2003; Kazama et al., 2002; Tanabe and Wilbur, 2002; Yamamoto et al., 2003; Tsuruoka, 2006; Okanohara et al., 2006). Machine learning algorithms used in these studies are Naive Bayes, C4.5, Maximum Entropy Models, Support Vector Machines, and Conditional Random Fields. Most of these studies applied machine learning techniques to *tokenized* sentences.

Table 3 shows the scores reported by other systems. Tsai et al. (Tsai et al., 2006) and Zhou and Su (Zhou and Su, 2004) combined machine learning techniques and hand-crafted rules. Tsai et al. (Tsai et al., 2006) applied CRFs to the JNLPBA-2004 data. After applying pattern-based post-processing, they achieved the best F-score (75.12) among those reported so far. Kim and Yoon(Kim and Yoon, 2007) also applied heuristic post-processing. Zhou and Su (Zhou and Su, 2004) achieved an F-score of 73.77.

Purely machine learning-based approaches have been investigated by several researchers. The GENIA Tagger (Tsuruoka, 2006) is trained on the JNLPBA-2004 Corpus. Okanohara et al. (Okanohara et al., 2006) employed semi-Markov CRFs whose performance was evaluated against the JNLPBA-2004 data set. Yamamoto et al. (Yamamoto et al., 2003) used SVMs for character-based protein name recognition and sequential labelling. Their protein name extraction performance was 69%. This paper extends the machine learning approach with a curated dictionary and CRFs and achieved high F-score 73.78, which is the top score among the heuristics-free NER systems. Table 4 shows typical recognition errors found in the recognition results that achieved F-score 73.78. In some cases, protein name boundaries of the JNLPBA-2004 data set are not consistent. It is also one of the reasons for the recognition errors that the data set contains general protein names, such as domain, family, and binding site names as well as anaphoric expressions, which are usually not covered by protein name repositories. Therefore, our impression on the performance is that an F-score of 73.78 is sufficiently high.

Furthermore, thanks to the dictionary-based approach, it has been shown that the upper bound performance using ideal dictionary enrichment, without any retraining of the models, has an F-score of 78.72.

4 Conclusions

This paper has demonstrated how to utilize known named entities to achieve better performance in statistical named entity recognition. We took a two-step approach where sentences are first tokenized and tagged based on a biomedical dictionary that consists of general English words and about 1.3 million protein names. Then, a statistical sequence labelling step predicted protein names that are not listed in the dictionary and, at the same time, reduced false negatives in the POS/PROTEIN tagging results. The significant benefit of this approach is that a user, not a system developer, can easily enhance the performance by augmenting the dictionary. This paper demonstrated that the state-of-the-art F-score 73.78 on the standard JNLPBA-2004 data set was achieved by our approach. Furthermore, thanks to the dictionary-based NER approach, the upper bound performance using ideal dictionary enrichment, without any retraining of the models, yielded F-score 78.72.

5 Acknowledgments

This research is partly supported by EC IST project FP6-028099 (BOOTStrep), whose Manchester team is hosted by the JISC/BBSRC/EPSRC sponsored National Centre for Text Mining.

Table 3: Conventional results for protein name recognition

Authors	R	P	F
Tsai et al.(Tsai et al., 2006)	71.31	79.36	75.12
Our system	**79.85**	**68.58**	**73.78**
Zhou and Su(Zhou and Su, 2004)	69.01	79.24	73.77
Kim and Yoon(Kim and Yoon, 2007)	75.82	71.02	73.34
Okanohara et al.(Okanohara et al., 2006)	77.74	68.92	73.07
Tsuruoka(Tsuruoka, 2006)	81.41	65.82	72.79
Finkel et al.(Finkel et al., 2004)	77.40	68.48	72.67
Settles(Settles, 2004)	76.1	68.2	72.0
Song et al.(Song et al., 2004)	65.50	73.04	69.07
Rössler(Rössler, 2004)	72.9	62.0	67.0
Park et al.(Park et al., 2004)	69.71	59.37	64.12

References

J. Aoe, An Efficient Digital Search Algorithm by Using a Double-Array Structure, *IEEE Transactions on Software Engineering*, 15(9):1066–1077, 1989.

L.E. Baum and T. Petrie, Statistical inference for probabilistic functions of finite state Markov chains, *The Annals of Mathematical Statistics*, 37:1554–1563, 1966.

J. Chang, H. Schutze, R. Altman, GAPSCORE: Finding Gene and Protein names one Word at a Time, *Bioinformatics*, Vol. 20, pp. 216-225, 2004.

N. Collier, C. Nobata, J. Tsujii, Extracting the Names of Genes and Gene Products with a Hidden Markov Model, *Proc. of the 18th International Conference on Computational Linguistics (COLING'2000)*, Saarbrucken, 2000.

Jenny Finkel, Shipra Dingare, Huy Nguyen, Malvina Nissim, Gail Sinclair and Christopher Manning, Exploiting Context for Biomedical Entity Recognition: From Syntax to the Web, *Proc. of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, pp. 88–91, 2004.

K. Franzen, G. Eriksson, F. Olsson, L. Asker, P. Liden, and J. Koster, Protein Names and How to Find Them, *Int. J. Med. Inf.*, Vol. 67, pp. 49–61, 2002.

K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi, Toward information extraction: identifying protein names from biological papers, *PSB*, pp. 705-716, 1998.

J. Kazama, T. Makino, Y. Ohta, J. Tsujii, Tuning Support Vector Machines for Biomedical Named Entity Recognition, *Proc. of ACL-2002 Workshop on Natural Language Processing in the Biomedical Domain*, pp. 1–8, 2002.

J.-D. Kim, T. Ohta, Y. Tateisi, J. Tsujii: GENIA corpus - semantically annotated corpus for bio-textmining, *Bioinformatics* 2003, 19:i180-i182.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, Introduction to the Bio-Entity Recognition Task at JNLPBA, *Proc. of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, pp. 70–75, 2004.

S. Kim, J. Yoon: Experimental Study on a Two Phase Method for Biomedical Named Entity Recognition, *IEICE Transactions on Informaion and Systems* 2007, E90-D(7):1103–1120.

Taku Kudo and Kaoru Yamamoto and Yuuji Matsumoto, Applying Conditional Random Fields to Japanese Morphological Analysis, *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*, pp. 230–237, 2004.

J. Lafferty, A. McCallum, and F. Pereira, Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, *Proc. of ICML-2001*, pp.282–289, 2001

K. J. Lee, Y. S. Hwang and H. C. Rim (2003), Two-Phase Biomedical NE Recognition based on SVMs, *Proc. of ACL 2003 Workshop on Natural Language Processing in Biomedicine*, Sapporo, 2003.

McCallum A, Freitag D, Pereira F.: Maximum entropy Markov models for information extraction and segmentation, *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000:591-598.

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

Kyung-Mi Park, Seon-Ho Kim, Do-Gil Lee and Hae-Chang Rim. Boosting Lexical Knowledge for Biomedical Named Entity Recognition, *Proc. of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, pp. 76-79, 2004.

Marc Rössler, Adapting an NER-System for German to the Biomedical Domain, *Proc. of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, pp. 92–95, 2004.

Burr Settles, Biomedical Named Entity Recognition Using Conditional Random Fields and Novel Feature

Table 4: Error Analysis

	False positives		
	Cause	Correct extraction	Identified term
1	dictionary	-	protein, binding sites
2	prefix word	trans-acting factor	common trans-acting factor
3	unknown word	-	ATTTGCAT
4	sequential labelling error	-	additional proteins
5	test set error	-	Estradiol receptors
	False negatives		
	Cause	Correct extraction	Identified term
1	anaphoric	(*the*) receptor, (*the*) binding sites	-
2	coordination (and, or)	transcription factors NF-kappa B and AP-1	transcription factors NF-kappa B
3	prefix word	activation protein-1	protein-1
		catfish STAT	STAT
4	postfix word	nuclear factor kappa B complex	nuclear factor kappa B
5	plural	protein tyrosine kinase(s)	protein tyrosine kinase
6	family name, biding site,	T3 binding sites	-
	and domain	residues 639-656	-
7	sequential labelling error	PCNA	-
		Chloramphenicol acetyltransferase	-
8	test set error	superfamily member	-

Sets, *Proc. of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, pp. 104–1007, 2004.

Yu Song, Eunju Kim, Gary Geunbae Lee and Byoungkee Yi, POSBIOTM-NER in the shared task of BioNLP/NLPBA 2004, *Proc. of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, pp. 100-103, 2004.

L. Tanabe and W. J. Wilbur, Tagging Gene and Protein Names in Biomedical Text, *Bioinformatics*, 18(8), pp. 1124–1132, 2002.

E.F. Tjong Kim Sang and J. Veenstra, Representing Text Chunks,*EACL-99*, pp. 173-179, 1999.

Richard Tzong-Han Tsai, W.-C. Chou, S.-H. Wu, T.-Y. Sung, J. Hsiang, and W.-L. Hsu, Integrating Linguistic Knowledge into a Conditional Random Field Framework to Identify Biomedical Named Entities, *Expert Systems with Applications*, 30 (1), 2006.

Yoshimasa Tsuruoka, GENIA Tagger 3.0, `http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/`, 2006.

K. Yamamoto, T. Kudo, A. Konagaya and Y. Matsumoto, Protein Name Tagging for Biomedical Annotation in Text, in Proc. of ACL-2003 Workshop on Natural Language Processing in Biomedicine, Sapporo, 2003.

Guofeng Zhou and Jian Su, Exploring Deep Knowledge Resources in Biomedical Name Recognition, *Proceedings of the Joint Workshop on Natural Language Processing of Biomedicine and its Applications (JNLPBA-2004)*, pp. 96-99, 2004.