Developing Basic NLP Tools for Biomedical Texts

Yoshimasa Tsuruoka[12] tsuruoka@is.s.u-tokyo.ac.jp
Jin-Dong Kim[2] jdkim@is.s.u-tokyo.ac.jp
Yuka Tateishi[12] yucca@is.s.u-tokyo.ac.jp
Tomoko Ohta[12] okap@is.s.u-tokyo.ac.jp
Jun'ichi Tsujii[21] tsujii@is.s.u-tokyo.ac.jp

[1] CREST, JST (Japan Science and Technology Agency), Honcho 4-1-8, Kawaguchi-shi, Saitama 332-0012, Japan
[2] Department of Computer Science, University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033, Japan

Abstract

This paper presents a part-of-speech tagger which are specifically tuned for biomedical texts. We have built the tagger using maximum entropy models with inequality constraints, which have high generalization capacity and can produce compact models. The tagger was trained on a corpus containing newspaper articles and biology texts so that it works well on various types of biomedical documents. Experimental results using the Penn Treebank corpus and the GENIA corpus show that our tagger performs very well on both corpora (96.9% precision on the Penn Treebank, 98.1% on the GENIA corpus). We also present a named-entity tagger which employs a new strategy to make use of the features that are not available in conventional strategies. The tagger exhibited an f-score of 70.7% on a standard evaluation set. The taggers presented in this paper are publicly available on our web-site.

Keywords: natural language processing, part-of-speech tagging, named entity tagging

1 Introduction

Since huge amount of biomedical knowledge is described in texts (e.g. MEDLINE abstracts), automatic information extraction from biomedical documents increasingly plays an important role for the researchers in the biomedical domain.

For extracting information from texts, many natural language processing (NLP) techniques can be employed. For example, a simple approach to extract information about protein-protein interaction is to write regular expressions about part-of-speech tags and noun-phrases. A more sophisticated approach would be to use parsers to precisely analyze syntactic and semantic structure of the sentences.

For the documents like newspaper articles, many NLP tools are publicly available, including part-of-speech taggers, chunkers, and parsers. However, they do not necessarily work well on biomedical documents because the characteristic of biomedical documents are quite different from that of newspaper articles, which are often used as the training data for general-purpose NLP tools.

In this paper, we first present a part-of-speech tagger which are specifically tuned for biomedical texts. Since the part-of-speech tags assigned to the words significantly affect the performance of the subsequent processings, a part-of-speech tagger must be as reliable as possible. In order to build a robust part-of-speech tagger, we adopt a maximum entropy model with inequality constraints [7] and use a training set that contains both newspaper articles and biology texts.

We also present a named-entity tagger which employs a new strategy to make use of the features that are not available in conventional strategies. The performance of the tagger is evaluated on a common evaluation data set.

Our aim is to make reliable NLP tools for biomedical documents publicly available and to promote deeper analyses. The taggers presented in this paper are downloadable from our web-site.

This paper is organized as follows. Section 2 describes the model of the proposed part-of-speech tagger. Section 3 describes the algorithm of the named entity tagger. Experimental results are provided in Section 4. Finally, Section 5 offers some concluding remarks.

2 Part-of-speech Tagging

Part-of-speech tagging is a basic processing for natural language processing. The task is to assign each word with its part-of-speech tag. Because deeper processings such as chunking and parsing generally trust the part-of-speech tags assigned to the words, the tags must be highly accurate.

In order to build a part-of-speech tagger, we need a corpus for training. The Penn Treebank corpus [12], which is a collection of newspaper articles, contains part-of-speech tags and can be used as training data. However, since the tagger must work well on biomedical documents, we need a corpus of such documents. We briefly review the GENIA part-of-speech corpus in the following section.

2.1 The GENIA Part-of-speech corpus

We have made a part-of-speech (POS) corpus, where POS information was annotated to the raw texts of the GENIA corpus. In the corpus, POS is assigned to each word in the text according its syntactic role. The principle is applied even to the words that are part of multi-word to terms. That is, each component of a multi-word term is assigned a POS according to the syntactic role of the word, not according to the role of the term. The annotation scheme for the POS corpus is based on that of Penn Treebank (PTB) corpus widely used in constructing general-purpose statistics -based NLP-systems. We modified the PTB scheme slightly in order to achieve consistent annotation: the use of NNP and NNPS tags is limited so that only the names of the months, the names of authors of the papers, journals, research institutes, and initials of patients and other people who contributed to experiments described in the paper, and all other nouns are tagged as common nouns, even when a person's name appears as a part of other names (e.g., Cushing's syndrome, Southern blotting).

The decision was made because the need for the distinction is rather small from the viewpoint of syntactic processing such as parsers while the distinction is costly for consistency due to the abundance of non-proper names that begin with a capital letter in biology texts. Prefixes and postfixes are tagged based on their syntactic role. For example, the token 'up-' in 'up- and downregulation' is assigned an RP (particle) tag because it originate from a particle (em regulate up), and an NNS (plural noun) tag is assigned to the token 's' in 'factor(s)'.

The POS corpus (2,000 annotated abstracts) is publicly available in three formats. One is a "PTB-like" format where there are one TOKEN/POS pair per line. Another is an XML format where tokens are represented in *w* elements and the POS is represented as the *c* attribute. Yet another is a "merged" format where the POS annotation is merged into the term corpus (Figure 1).

In the "merged" version, it is assumed that the *w* elements are inside the *cons* elements. However, sometimes a token was split by the <*cons*> tags, i.e.,a technical term represented by a *cons* element is inside a token represented by a *w* element. For example, in Figure 1 the token *IL-2-mediated* because of <*cons*> tags around *IL-2*. In such cases, we made each fragment one *w* element. The last fragment of the split token is assigned the original POS assigned to the whole token and all others are assigned * as the value of the *c* attribute, as shown in Figure 1 This phenomenon shows that tokenization is problematic in biology texts.

<abstract>
<sentence><cons lex="IL-2-mediated_T_cell_proliferation" sem="G#other_name"><cons lex="IL-2" sem="G#protein_molecule"><w c="*">IL-2</w></cons><w c="JJ">-mediated</w> <cons lex="T_cell" sem="G#cell_type"><w c="NN">T</w> <w c="NN">cell</w></cons> <w c="NN">proliferation</w></cons> <w c="VBZ">is</w> <w c="DT">a</w> <w c="JJ">critical</w> <w c="JJ">early</w> <w c="NN">event</w> ...</sentence>
...

Figure 1: POS information merged with the GENIA term corpus

Table 1: Contextual predicates used in the part-of-speech tagger

$t_{i-1} = X$
$t_{i-1}w_i = X$
$w_{i-1} = X$
$w_i = X$
$w_{i+1} = X$
$w_{i-1}w_i = X$
$w_iw_{i+1} = X$
the first letter of w_i is uppercase
X is suffix of w_i, $\|X\| \leq 5$

2.2 Maximum Entropy Markov Model

Our tagger adopts a first-order Markov model for part-of-speech tagging. The states of the model represent part-of-speech tags. Given a sentence $\{w_1...w_n\}$, a tag sequence candidate $\{t_1...t_n\}$ has conditional probability:

$$P(t_1...t_n|w_1...w_n) = \prod_{i=1}^{n} p(t_i|t_{i-1}w_1...w_{i-1}) \tag{1}$$

Transition probabilities are estimated using a maximum entropy model. The model can make use of the information of the preceding tags and all the words in the sentence. Table 1 shows the contextual features used in our tagger.

Maximum entropy models require the devices to alleviate the problem of overfitting. We adopt maximum entropy modeling with inequality constraints proposed by Kazama and Tsujii [7]. This model has high generalization capacity comparable to the use of Gaussian priors [3], which is the most popular method to avoid overfitting. This model has an advantage that the solution of parameter estimation becomes sparse, resulting in a compact set of parameters. This advantage is especially useful in terms of developing practical tools because compact models require less computational cost and memory in run-time.

2.3 Training

The GENIA corpus consists of the abstracts that have the three MeSH keywords, "Human", "Blood", and "Transcription Factors". So the corpus is a good training set for biology documents including many gene and protein names. However, the corpus is not sufficient to achieve high performance for various types of documents in MEDLINE abstracts such as medical documents. For that reason, we used not only the GENIA corpus but also the Penn Treebank corpus for training.

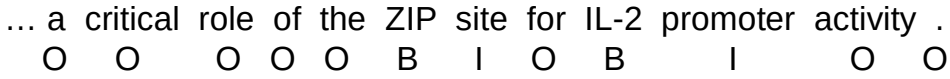

Figure 2: The BIO framework for named entity recognition

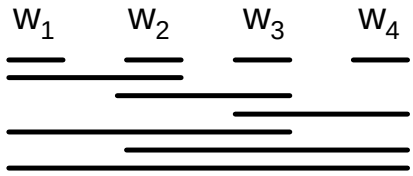

Figure 3: The sub-word-sequences for a four-word sentence

3 Named-entity Tagging

To be able to extract information about proteins from a text, one has to first recognize their names in it. This kind of problem has been extensively studied in the field of natural language processing as the named-entity recognition task. The most popular approach is to train the recognizer on an annotated corpus by using a machine learning algorithm, such as Hidden Markov Models, support vector machines (SVMs) [17], and maximum-entropy models [1]. The task of the classifier in the machine learning framework is to determine the text regions corresponding to protein names.

Ohta et al. provided the GENIA corpus [13] with named-entity tags, which could be used as a gold-standard for evaluating and training named-entity recognition algorithms. The corpus has fostered research on machine learning techniques for recognizing biological entities in texts [15, 9, 6].

3.1 Method

The most popular approach for machine-learning based named entity recognition is to assign a tag to each word. Figure 2 shows an example of such approach, where the tagger assigns 'B' to the beginning of a term, 'I' to the inside of a term, and 'O' to the other words.

In the BIO framework, you do not have the information of the last word of a term when looking at the beginning of the term. Therefore, for example, you cannot make use of a feature concerning the whole word-sequence of the term.

In this paper, we take a different approach. We consider all the sub word-sequences of a sentence as the candidates of named entities, and classify each candidate sub word-sequence by machine learning (Figure 3). This approach enables us to use various types of features that cannot be incorporated in the BIO framework.

However, if we take this approach in a naive way, we face a serious problem of computational cost. Because the number of sub word-sequence is $O(n^2)$ of the length of a sentence, the number of candidates becomes prohibitively large if the sentence is long.

In a preliminary experiment, we tried to train the classifier using all the sub word-sequences in the training data. However, it turned out that it was impossible to train because the training required too much memory and time.

To reduce the burden of the classifier, we propose a two-phase approach. In the first phase, we select named-entity candidates from all the sub word-sequences with a simple statistical method that requires far less computational cost than machine learning. In the second phase, a machine learning algorithm is employed to choose named-entities from the selected candidates.

Table 2: Contextual predicates used in the named-entity tagger

$w_{b-2} = X$
$w_{b-1} = X$
$w_{e+1} = X$
$w_{e+2} = X$
$w_{b-2}w_{b-1} = X$
$w_{b-1}w_{e+1} = X$
$w_{e+1}w_{e+2} = X$
$w_b = X$
$w_e = X$
$w_i = X, i \geq b, i \leq e$
the first and the last letter of w_i are uppercase
X is suffix of w_e, $\|X\| \leq 5$

Table 3: Training sets for the experiments of part-of-speech tagging

	Penn Treebank	GENIA corpus
Training Set A	39,832	0
Training Set B	0	18,508
Training Set C	39,832	18,508

The problem is how to select candidates in the first phase. The goal of the first phase is to efficiently filter out the word-sequences that cannot be a named entity. Intuitively, one can rule out the word-sequences that include common English words such as "We", "show", and "are". In order to automatically perform such filtering, we first calculate for every word the probability that the word becomes a part of a named-entity. Then we discard all the candidates including the words, the probability of which is lower than the predefined threshold.

We use a maximum entropy classifier in the second phase. Some representative features used in the model are shown in Table 2, where b and e represent the starting and ending position of the candidate term respectively.

4 Experiment

4.1 Part-of-speech tagging

We prepared three sets of sentences for training. Table 3 shows the number of sentences contained in the training sets. Training set A contains all the sentences in Sections 2 to 21 in the Penn Treebank corpus. Training set B was constructed from the GENIA corpus by random selection. Training set C was made by merging set A and B.

Training sets and test sets are mutually exclusive: no sentences in the training sets were included in the test sets. The test set for the Penn Treebank was constructed from Section 23, which is often used as an evaluation set. The test set for the GENIA corpus consists of 2,036 sentences.

Table 4 shows the performance of the taggers trained on different sets of data. The tagger trained on set A achieved 97.0% on the Penn Treebank, which is very high. However, the tagger exhibits significantly lower performance on the GENIA corpus. On the other hand, the tagger trained on the GENIA corpus performs quite well on the GENIA corpus, but the performance on the Penn Treebank

Table 4: Part-of-speech tagging performance

	Penn Treebank	GENIA corpus
Our tagger trained on Set A	97.0%	84.3%
Our tagger trained on Set B	75.2%	98.1%
Our tagger trained on Set C	96.9%	98.1%

Table 5: Comparing with the TnT tagger

	Penn Treebank	GENIA corpus
TnT tagger with the precompiled PTB model	97.4%	84.4%
TnT tagger trained on Set A	96.7%	84.3%
TnT tagger trained on Set B	80.1%	97.9%
TnT tagger trained on Set C	96.5%	97.5%

is disastrous.

Note that the tagger trained on set C works surprisingly well on both corpora. The degradation of the performance from the taggers specifically trained for each corpus is negligible. This result indicates the robustness of the tagger, and it is expected to work well on other documents in the biomedical domain.

For comparison, we performed experiments using the TnT tagger [2], which is one of the state-of-the-art taggers publicly available. Table 5 shows the performance of the TnT tagger. The tagger has a pre-compiled model that was constructed from the Penn Treebank corpus, and the performance is given in the first row. The results indicate that the general-purpose tagger performs quite poorly on the biology texts. Note that our tagger trained on Set C performs significantly better than the TnT tagger trained on the same set.

The pre-compiled model showed a slightly higher precision on the Penn Treebank than our tagger. However, the reason is that the precompiled model was created by using the sentences in the test set, so it is natural that the tagger showed good performance on the data.

As stated in 2.1, the annotation policy of the of the GENIA corpus is slightly different from that of the Penn Treebank corpus. The GENIA corpus does not distinguish NNP from NN to keep annotation consistent. This disparity of annotation policy can make the result of the TnT tagger look worse than it really is. In order to clarify the effect of the policy difference, we have conducted experiments by not distinguishing NNP from NN. The results are shown in Table 6. The performance of the TnT tagger is 90.0%, which is still much lower that of our tagger trained on set C. This result confirms the advantage of our tagger over the TnT tagger.

Table 6: Comparing with the TnT tagger (NNP = NN, NNPS = NNS)

	Penn Treebank	GENIA corpus
TnT tagger with the precompiled PTB model	97.5%	90.0%

Table 7: Comparing with Other Models

	Recall	Precision	F-score
Support Vector Machines & HMM [18]	76.0%	69.4%	72.6%
Our model	72.8%	68.8%	70.7%
Maximum Entropy Markov Model [5]	71.6%	68.6%	70.1%
Conditional Random Field [14]	70.3%	69.3%	69.8%

Table 8: Performance on Each Category

	Recall	Precision	F-score
protein	76.8%	69.5%	73.0%
cell line	54.4%	54.6%	54.5%
DNA	65.0%	68.1%	66.5%
cell type	65.6%	78.2%	71.3%
RNA	68.6%	66.9%	67.8%

4.2 Named entity tagging

Until recently, it was difficult to compare the performance of named entity taggers because they use different corpora for evaluation. Kim et al. [8] provided a training and testing corpora for the shared task in the COLING workshop, which can be used for the standard evaluation set for named-entity taggers. In this paper, we used the data for evaluating our tagger.

Table 7 shows the performance of our model and the best three models reported in [8]. The performance of our tagger is not as good as the best model proposed by [18]. One reason suspected is that they use a kind of dictionary and it might have boosted the performance of their model. Although the performance of our model is not the best, the results that our model achieved better performance than a Maximum Entropy Markov Model and a Conditional Random Field, which are state-of-the-art techniques for named-entity recognition, are promising.

4.3 Related Work on Named-Entity tagging

Kazama et al. [6] reported an F-measure of 56.5% on the GENIA corpus version 1.1 using SVMs. Collier et al. [4] reported an F-measure of 75.9% on 100 MEDLINE abstracts using a Hidden Markov Model. Tanabe and Wilbur [16] achieved 85.7% precision and 66.7% recall using a combination of statistical and knowledge-based strategies. They used a transformation-based part-of-speech tagger to recognize single word protein names, and hand-crafted rules to filter out false positives and recover false negatives. Since the evaluation corpora used in these experiments were different from the corpus used in this paper, the results are not directly comparable.

Lee et al. [11] reported an F-measure of 69.2% on the GENIA corpus version 3.0 using SVMs. Shen et al. achieved an F-measure of 70.8% on the same corpus by incorporating various features into a Hidden Markov Model. Since the difference between the GENIA corpora versions 3.0 and 3.02, which we used in this paper, is small, their results suggest that their methods worked better than ours regarding recognition. However, their approaches do not provide ID information on recognized terms.

Krauthammer et al. [10] proposed a dictionary-based method of gene/protein name recognition. They used BLAST for approximate string matching by mapping sequences of text characters into sequences of nucleotides that could be processed by BLAST. They achieved a recall of 78.8% and a precision of 71.1% evaluated with a partial match criterion, which was not as stringent as our criterion.

5 Conclusion

We have developed a part-of-speech tagger and a named-entity recognizer which are specifically tuned for biomedical documents.

In the experiments using the Penn Treebank corpus and the GENIA corpus, the part-of-speech tagger exhibited state-of-the-art performance on both corpora. The tagger will serve as an ideal preprocessor for a deeper analysis including chunking and parsing.

The named entity recognizer can identify the terms of protein, DNA, RNA, cell-line, and cell-type with an f-score of 70.7%, which is currently the second best performance on a common evaluation set for biological named-entity recognition.

The tools presented in this paper is publicly available on
http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/genia/tagger/

References

[1] Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] Thorsten Brants. TnT – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference (ANLP)*, 2000.

[3] Stanley F. Chen and Ronald Rosenfeld. A gaussian prior for smoothing maximum entropy models. *Technical Report CMUCS -99-108, Carnegie Mellon University*, 1999.

[4] Nigel Collier, Chikashi Nobata, and Junichi Tsujii. Automatic acquisition and classification of molecular biology terminology using a tagged corpus. *Journal of Terminology*, 7(2):239–258, 2001.

[5] Jenny Finkel, Huy Nguyen, Shipra Dingare, Malvina Nissimand Christopher Manning, and Gail Sinclair. Exploiting context for biomedical entity recognition: From syntax to the web. In *Proceedings of the COLING 2004 Workshop on Natural Language Processing in Biomedicine and its Applications*, 2004.

[6] Jun'ichi Kazama, Takaki Makino, Yoshihiro Ohta, and Jun'ichi Tsujii. Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain*, pages 1–8, 2002.

[7] Jun'ichi Kazama and Jun'ichi Tsujii. Evaluation and extension of maximum entropy models with inequality constraints. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, pages 137–144.

[8] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateishi, and Nigel Collier. Introduction to the bio-entity task at jnlpba. In *Proceedings of the COLING 2004 Workshop on Natural Language Processing in Biomedicine and its Applications*, 2004.

[9] Jin Dong Kim and Jun'ichi Tsujii. Corpus-based approach to biological entity recognition. In *Text Data Mining SIG (ISMB2002)*, 2002.

[10] Michael Krauthammer, Andrey Rzhetsky, Pavel Morozov, and Carol Friedman. Using BLAST for identifying gene and protein names in journal articles. *Gene*, 259:245–252, 2000.

[11] Ki-Joong Lee, Young-Sook Hwang, and Hae-Chang Rim. Two-phase biomedical NE recognition based on SVMs. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pages 33–40, 2003.

[12] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1994.

[13] Tomoko Ohta, Yuka Tateisi, Jin-Dong Kim, and Jun'ichi Tsujii. Genia corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of the Human Language Technology Conference (HLT 2002)*, 2002.

[14] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the COLING 2004 Workshop on Natural Language Processing in Biomedicine and its Applications*, 2004.

[15] K. Takeuchi and N. Collier. Use of support vector machines in extended named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, pages 119–125, 2002.

[16] Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in biomedical text. *BIOINFORMATICS*, 18(8):1124–1132, 2002.

[17] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[18] Zhou. Recognizing names in biomedical texts using hidden markov and svm plus sigmoid. In *Proceedings of the COLING 2004 Workshop on Natural Language Processing in Biomedicine and its Applications*, 2004.