Developing a Robust Part-of-Speech Tagger for Biomedical Text

Yoshimasa Tsuruoka[12], Yuka Tateishi[12], Jin-Dong Kim[12], Tomoko Ohta[12], John McNaught[35], Sophia Ananiadou[45], and Jun'ichi Tsujii[23]

[1] CREST, JST (Japan Science and Technology Agency)
Honcho 4-1-8, Kawaguchi-shi, Saitama 332-0012, JAPAN
{tsuruoka,yucca,jdkim,okap}@is.s.u-tokyo.ac.jp
[2] University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, JAPAN
tsujii@is.s.u-tokyo.ac.jp
[3] School of Informatics, University of Manchester
POBox 88, Sackville St, MANCHESTER M60 1QD, UK
[4] School of Computing, Science and Engineering, Salford University
Salford, Greater Manchester M5 4WT, UK
S.Ananiadou@salford.ac.uk
[5] The National Centre for Text Mining
POBox 88, Sackville St, MANCHESTER M60 1QD, UK

Abstract. This paper presents a part-of-speech tagger which is specifically tuned for biomedical text. We have built the tagger with maximum entropy modeling and a state-of-the-art tagging algorithm. The tagger was trained on a corpus containing newspaper articles and biomedical documents so that it would work well on various types of biomedical text. Experimental results on the Wall Street Journal corpus, the GENIA corpus, and the PennBioIE corpus revealed that adding training data from a different domain does not hurt the performance of a tagger, and our tagger exhibits very good precision (97% to 98%) on all these corpora. We also evaluated the robustness of the tagger using recent MEDLINE articles.

1 Introduction

Since a huge amount of biomedical knowledge is described in the literature, automatic information extraction from biomedical documents is increasingly important for many researchers in this domain.

For extracting information from text, many natural language processing (NLP) techniques can be employed. For example, a simple approach to extracting information about protein-protein interactions would involve scanning the text for particular verbs and neighboring noun phrases by applying some linguistic patterns on words and their part-of-speech (POS) tags. A more sophisticated way would be to use parsers to deeply analyze the syntactic and semantic relations among the entities in the sentences.

In order to carry out noise-free information extraction, the very basic step in natural language processing of POS tagging must be performed with high precision. The precision of POS tagging not only directly affects the performance of pattern-based approaches but also influences the accuracy of parsing which in general uses the POS tags on the words as part of the input [1, 2].

For documents like newspaper articles, there are a number of publicly available NLP tools including POS taggers, chunkers (shallow parsers), and syntactic parsers. However, the problem for researchers working on biomedical information extraction is that such tools do not necessarily work well on biomedical documents because the characteristics of biomedical text are considerably different from those of newspaper articles, which are often used as the training data for NLP tools [3, 4] . Table 1 lists some examples of tagging errors made by the TnT tagger [5], a popular HMM-based POS tagger, which is trained on the Wall Street Journal corpus, when it is applied to biomedical text.

Recently, two large biomedical corpora that are annotated with POS tags have become publicly available: the GENIA corpus [6] and the PennBioIE corpus [3]. In building these corpora, the developers used a POS tagger to reduce manual annotation effort and reported that they could achieve better performance than with a standard tagger by using an already annotated portion of their corpus for training the tagger. Their observation clearly suggests that we might be able to build a good POS tagger for biomedical documents if we use their corpora as the training data.

However, since each corpus consists of text extracted from a particular domain (e.g. transcription factors for the GENIA corpus) and does not cover the entire characteristics of biomedical text, there are still remaining issues to be addressed: (1) Which corpus should we use for training? (2) Should we use a single corpus or combine two corpora? (3) Does the combination of corpora from different domains have a bad effect on trained tagger performance? if so, how much?

The purpose of this paper is to clarify these issues and develop a reliable POS tagger that can be used as a fundamental tool for biomedical text mining. In this paper we evaluate the performance of a part-of-speech tagger by using different combinations of corpora as the training data, and show how the domain of the training corpus affects the tagging performance. We also investigate the robustness of the trained taggers using recent MEDLINE articles.

2 POS Tagging Algorithm

As our POS tagging algorithm, we adopt a method based on a Cyclic Dependency Network proposed by Toutanova et al. [8], which is currently one of the best algorithms for English POS tagging. Unlike the popular Maximum Entropy Markov Modeling (MEMM) approach, this method can incorporate features about the tags on both sides of the classification target. Toutanova et al. achieved an accuracy of 97.24% on sections 22-24 in the Wall Street Journal corpus, using sections 0-18 for training. On the same sets for training and testing, Gimenez

Tagging Errors	Correct Tagging
... and membrane potential after mitogen binding. CC NN NN IN NN **JJ**	binding NN
... two factors, which bind to the same kappa B enhancers ... CD NNS WDT **NN** TO DT JJ NN NN NNS	bind VBP
... by analysing the Ag amino acid sequence. IN VBG DT **VBG** JJ NN NN	Ag NN
... to contain more T-cell determinants than ... TO VB **RBR JJ** NNS IN	more T-cell JJR NN
Stimulation of interferon beta gene transcription in vitro by NN IN JJ JJ NN NN **IN NN** IN	in vitro FW FW

Table 1. Examples of tagging errors made by an HMM-based tagger trained on the Wall Street Journal corpus. The tagset includes NN (Noun, singular or mass), JJ (Adjective), VBP (Verb, non-3rd ps. sing. present), VBG (Verb, gerund/present participle), JJR (Adjective, comparative), RBR (Adverb, comaparative), IN (Preposition/subordinating conjunction), and FW (Foreign word). For the complete information about the tagset, see [7].

and Marquez [9] achieved an accuracy of 97.05% with support vector machines and Collins [10] achieved an accuracy of 97.11% with a discriminative HMM model.

2.1 POS tagging with a Cyclic Dependency Network

We briefly describe the POS tagging algorithm based on a cyclic dependency network. For further details of the algorithm, see [8].

Given a sentence $\{w_1...w_n\}$, the task of POS tagging is to find the tag sequence that maximizes the following score:

$$score = \prod_{i=1}^{n} p(t_i|t_{i-2}t_{i-1}t_{i+1}t_{i+2}w_1...w_n) \quad (1)$$

where t_i is the POS tag of the ith position. The best tag sequence can be computed in polynomial time by dynamic programming.

A probabilistic classifier is employed for estimating the local probabilities $p(t_i|t_{i-2}t_{i-1}t_{i+1}t_{i+2}w_1...w_n)$, which give the probability distribution for the tags on each token.

The advantage of this modeling over the standard left-to-right decomposition is that we can incorporate the information about the tags on both sides of t_i, i.e. $(t_{i-2}t_{i-1})$ and $(t_{i+1}t_{i+2})$ in performing local classification.

2.2 Local probabilistic classifier

We use maximum entropy modeling with inequality constraints [11] as the local probabilistic classifier. This modeling has a comparable generalization capacity

Current word	w_i	& t_i
Previous word	w_{i-1}	& t_i
Next word	w_{i+1}	& t_i
Bigram features	w_{i-1}, w_i	& t_i
	w_i, w_{i+1}	& t_i
Previous tag	t_{i-1}	& t_i
Tag two back	t_{i-2}	& t_i
Next tag	t_{i+1}	& t_i
Tag two ahead	t_{i+2}	& t_i
Tag Bigrams	t_{i-2}, t_{i-1}	& t_i
	t_{i-1}, t_{i+1}	& t_i
	t_{i+1}, t_{i+2}	& t_i
Tag Trigrams	t_{i-2}, t_{i-1}, t_{i+1}	& t_i
	t_{i-1}, t_{i+1}, t_{i+2}	& t_i
Tag 4-grams	t_{i-2}, t_{i-1}, t_{i+1}, t_{i+2}	& t_i
Tag/Word	t_{i-1}, w_i	& t_i
combination	t_{i+1}, w_i	& t_i
	t_{i-1}, t_{i+1}, w_i	& t_i
Prefix features	prefixes of w_i (up to length 10)	& t_i
Suffix features	suffixes of w_i (up to length 10)	& t_i
Lexical features	whether w_i has a hyphen	& t_i
	whether w_i has a number	& t_i
	whether w_i has a capital letter	& t_i
	whether w_i is all capital	& t_i

Table 2. Feature templates used in POS tagging experiments. Tags are parts-of-speech.

to that of Gaussian priors [12], which is a popular method for regularization in maximum entropy modeling. The advantage of this modeling is that most of the parameters become zero after training, resulting in a compact set of parameters. This advantage is especially useful in developing practical NLP tools because compact models require less computational cost and memory at run-time. This modeling has one meta-parameter called *width factor* for regularization. We tuned this parameter using the development data and set it to be 1.0.

For the features used in local classification, we adopted the feature set provided by [8] except for complex features like crude company name detection features because they are too specific to newspaper articles. Table 2 lists the feature templates used in our experiments.

2.3 Pruning

One problem of the tagging algorithm based on a cyclic dependency network is the computational cost for decoding (finding the best tag sequence) because the search space is very large.

To reduce the search space of Viterbi decoding in POS tagging, Ratnaparkhi [13] proposed to use a *Tag Dictionary* by which we consider only the tag-word

	# tokens	# sentences
WSJ for training	912,344	38,219
GENIA for training	450,492	18,508
PennBioIE for training	641,838	29,422
WSJ for testing	129,654	5,462
GENIA for testing	50,562	2,036
PennBioIE for testing	70,713	3,270

Table 3. Statistics of the corpora used in the experiments.

pairs that appear in the training sentences as the candidate tags. However, in our preliminary experiments, the use of a tag dictionary limits precision because the training set does not cover all the tag-word pairs which appear in unseen data. We thus take a different approach to reducing the computational cost for decoding.

We first generate the candidate tags on each word using the zero-th order probability $p(t_i|w_1...w_n)$ given by the local classifier trained without the information about the adjacent tags. If the probability of a candidate is lower than one hundredth of that of the tag with the highest probability, the candidate is not considered in the decoding. This pruning method gave considerable speed-up with little loss of tagging accuracy.

3 Experiments on Annotated Corpora

The first set of experiments was carried out on the three corpora that are annotated with POS tags.

3.1 Corpora

We used the following three corpora for training and testing.

- Wall Street Journal (WSJ) corpus
  The corpus is included in the Penn Treebank [7] and consists of 1 million words of 1989 Wall Street Journal material. Each word is annotated with part-of-speech tags. We split the corpus into the training and the test set, following a standard splitting criterion provided in [8]: Sections 0-18 for training, 19-21 for development, and 22-24 for testing. The development set was used for feature selection and parameter tuning.
- GENIA corpus (version 3.02) [6]
  The corpus consists of 2,000 MEDLINE abstracts that have the three MeSH keywords, "Human", "Blood", and "Transcription Factors". We constructed the training set using 90% of the corpus and the test set using the rest.
- PennBioIE corpus (Release 0.9) [3]
  The corpus contains the MEDLINE abstracts in two domains of biomedical knowledge: (1) inhibition of the cytochrome P450 family of enzymes (1100

	WSJ	GENIA	PennBioIE
WSJ	97.05	85.19	86.14
GENIA	78.57	98.49	86.59
PennBioIE	85.45	93.20	97.74
WSJ + GENIA	96.96	98.32	91.98
WSJ + PennBioIE	96.94	93.34	97.75
GENIA + PennBioIE	85.60	98.35	97.63
WSJ + GENIA + PennBioIE	96.89	98.20	97.68

Table 4. POS tagging accuracy on the test sets.

texts) (2) molecular genetics of cancer (1157 texts). We constructed the training data by merging the first 90% of the text from each domain. The rest was used as the test data.

The statistics are shown in Table 3. Training sets and test sets are mutually exclusive: no sentences in the training sets were included in the test sets.

3.2 POS tagging performance

	WSJ	GENIA	PennBioIE
WSJ	97.20	91.55	90.51
GENIA	85.27	98.55	92.21
PennBioIE	87.35	93.44	97.92
WSJ + GENIA	97.20	98.54	93.60
WSJ + PennBioIE	97.21	94.03	97.97
GENIA + PennBioIE	88.34	98.41	97.84
WSJ + GENIA + PennBioIE	97.20	98.35	97.87

Table 5. POS tagging accuracy on the test sets (without the distinction between proper nouns and nouns).

We evaluated the performance of POS tagging with the following seven different combinations of the corpora as the training data.

- WSJ
- GENIA
- PennBioIE
- WSJ + GENIA
- WSJ + PennBioIE
- GENIA + PennBioIE
- WSJ + GENIA + PennBioIE

Table 4 shows the accuracies on the test sets. The tagger trained on the WSJ corpus achieved an accuracy of 97.05% on the test set of the WSJ corpus. Since this test set is the same as that used in [8], the accuracies are directly comparable. Our accuracy is slightly lower than their accuracy (97.24%). This might look strange because our tagger employs the same tagging algorithm. The suspected reason is that they used features which are specifically tuned to the WSJ corpus such as company-name detection features. We did not use such features because our target is biomedical text. The feature set we used in this paper is almost identical to those in [10], and our tagger gives comparable performance to that achieved by Perceptron (97.11%) [10] and SVMs (97.05%) [9].

The tagger trained on the GENIA corpus achieved an accuracy of 98.49% on the test set of the GENIA corpus, which is slightly better than the above-mentioned performance on the WSJ corpus. This suggests that the texts in the GENIA corpus are less diverse than the WSJ corpus.

An interesting observation is that the performance on the PennBioIE corpus was improved from 86.59% to 91.98% by adding the WSJ corpus on top of the GENIA corpus. This indicates that even the text from a considerably different domain could contribute to the improvement of the tagger.

The most important observation in Table 4 is that the taggers trained on multiple corpora give good performance on all the test sets corresponding to the training corpora. In other words, adding text from a different domain did not deteriorate the precision of the tagger, which clearly indicates the robustness of our tagger.

In analyzing the tagging results, we found that the evaluation scheme was too strict. As pointed out in [4], the distinction between proper nouns and (normal) nouns is often ambiguous in the biomedical domain. The majority of the errors were caused by failure to make this distinction correctly, and the precisions shown in Table 4 are thus correspondingly depressed.

Since this distinction is often unnecessary from the natural language processing point of view, we also calculated the precisions achieved by ignoring the distinction between nouns and proper nouns. The results are shown in Table 5. The tagger trained on the WSJ corpus achieved accuracies of about 90% on the GENIA corpus and the PennBioIE corpus, which are considerably better than those given by the strict evaluation scheme.

The key observation revealed in Table 4 becomes much clearer: no loss of accuracy on the WSJ corpus was observed by adding the GENIA corpus and the PennBioIE corpus to the WSJ corpus.

4 Experiments on Recent MEDLINE Articles

In the previous section we evaluated the performance of our tagger on existing annotated corpora, and the tagger trained on the combination of all the three corpora exhibited very good performance. This suggests that the tagger is robust and would work well on other types of biomedical documents. Nevertheless, we

cannot rule out the possibility of over-fitting: The tagger might have shown good performance on the text in the particular domain from which the training data was constructed. To fully evaluate the robustness of the tagger, we need to use totally unseen text for the taggers.

In order to investigate the robustness of the tagger, we collected several recent abstracts of papers in three popular biomedical journals: Nucleic Acid Research (NAR), Nature Medicine (NMED), and Journal of Clinical Investigation (JCI). We randomly chose three abstracts from the latest issue of each journal, which are all published later than March 2005. The total number of tokens was 1,835.

Because the purpose is to evaluate the relative performance of the taggers, we focused only on the tokens where the taggers showed discrepancies. Of all the 1,835 tokens in the text, 330 tokens are tagged differently. We manually annotated the tokens with correct POS tags and evaluated the accuracies of the taggers.

	NAR	NMED	JCI	Total (Accuracy)
WSJ	43	19	35	97 (26.6%)
GENIA	121	74	132	327 (89.8%)
PennBioIE	124	65	118	307 (84.3%)
WSJ + GENIA	106	73	129	308 (84.6%)
WSJ + PennBioIE	127	69	117	313 (86.0%)
GENIA + PennBioIE	123	75	134	332 (91.2%)
WSJ + GENIA + PennBioIE	128	72	131	331 (90.9%)

Table 6. Relative performance on recent MEDLINE articles.

	NAR	NMED	JCI	Total (Accuracy)
WSJ	109	47	102	258 (70.9%)
GENIA	121	74	132	327 (89.8%)
PennBioIE	129	65	122	316 (86.8%)
WSJ + GENIA	125	74	135	334 (91.8%)
WSJ + PennBioIE	133	71	133	337 (92.6%)
GENIA + PennBioIE	128	75	135	338 (92.9%)
WSJ + GENIA + PennBioIE	133	74	139	346 (95.1%)

Table 7. Relative performance on recent MEDLINE articles (without the distinction between proper nouns and nouns).

The results are shown in Table 6 and 7. The tables show the numbers of correct tags given by individual taggers. Again, the tagger trained on the combined corpus performed best, which confirms the robustness of the tagger.

4.1 Error analysis

Our experimental results revealed that the tagger trained on texts from all three corpora gives the best performance. We investigated what types of errors are still remaining.

Some are errors that could be corrected with parsing. For example, in the sentence

> "These amplicons consist of a long inverted repeat with telomeric repeats at *both* ends and contain either the two different targeting cassettes used to inactivate JBP1 , or one cassette and one JBP1 gene ."

where *both* is incorrectly recognized as part of a *both – and* construction and labeled *CC*, the word can be assigned a proper POS if the coordination is correctly analyzed and 'ends' and 'contain' cannot be coordinated.

In the sentences

> "Both RNase E and RNase III *control* the stability of sodB mRNA upon translational inhibition by the small regulatory RNA RyhB ."

and

> "Using neutralizing antibodies and lactadherin-deficient animals , we show that lactadherin interacts with alphavbeta3 and alphavbeta5 integrins and *alters* both VEGF-dependent Akt phosphorylation and neovascularization .",

where *control* and *alters* are wrongly tagged as nouns, parsing will predict that they should be verbs (*VBN* and *VBZ* respectively) because a sentence needs a main verb.

There was one error, the correction of which would need deeper analysis. In the sentence

> "In the absence of VEGF , lactadherin administration induced alphavbeta3- and alphavbeta5-dependent Akt phosphorylation in endothelial cells in vitro and strongly *improved* postischemic neovascularization in vivo ."

even syntactic parsing cannot determine whether *improved* is a past form or past participle of a verb. Sentences like this one suggest that it may be dangerous to assign a single POS to a word before deeper syntactic and semantic analysis. Our future work should encompass allowing the tagger to output multiple candidate tags for each word and investigating the cost in parsing that would stem from this ambiguity.

The remaining errors have more lexical nature involving words that have several possible POSs but one is preferred over the others in the context of biomedical research abstracts. For example, in

> "Each long repeat within the linear amplicons corresponds to sequences covering the JBP1 locus , starting at the telomeres upstream of JBP1 and ending in a approximately 220 bp sequence repeated in an inverted (palindromic) orientation *downstream* of the JBP1 locus .",

the word 'downstream' is incorrectly labeled as a noun (*NN*), but in biomedical literature the word is more frequently used as adjective and that is true with this sentence. The error is expected to be eliminated if we can add more annotated biomedical texts to the training data. A similar result can be expected for the word 'set' (past participle incorrectly labeled as *NN*) in

> "In experiments with Leishmania tarentolae *set* up to disrupt the gene encoding the J-binding protein 1 (JBP1) , a protein binding to the unusual base beta-D-glucosyl-hydroxymethyluracil (J) of Leishmania , we obtained JBP1 mutants containing linear DNA elements (amplicons) of approximately 100 kb ."

and 'bleeding' (a nominal modifier incorrectly labeled as a verb taking object) in

> "In vivo , these inhibitors eliminate occlusive thrombus formation but do not prolong *bleeding* time .".

In the sentence

> "Mutations in these genes may increase smooth swimming of the bacteria , potentially allowing *more* effective interactions with and invasion of host cells to occur ."

the word 'more' would be correctly labeled as an adverb *RBR* if it is known that a word 'more' is rarely used as 'greater in number' in academic texts. In

> "These amplicons consist of a *long* inverted repeat with telomeric repeats at both ends and contain either the two different targeting cassettes used to inactivate JBP1 , or one cassette and one JBP1 gene .",

the word 'long' would be assigned the correct POS (adjective) if the word 'inverted' is not usually modified by an adverb 'long' (meaning 'for a long time').

5 Conclusion

This paper presented a part-of-speech tagger which is specifically suitable for processing biomedical text.

We have built the tagger based on a cyclic dependency network with maximum entropy modeling with inequality constraints, and evaluated the tagger on three corpora: the WSJ corpus, the GENIA corpus and the PennBioIE corpus.

Experimental results revealed that adding training data from a different domain does not hurt the performance of our POS taggers, and the tagger trained on the combined set of all three corpora offers very good performance (97% to 98% precision). We confirmed the robustness of the tagger by testing it further on several recent MEDLINE abstracts.

Acknowledgments

The UK National Centre for Text Mining is funded by the Joint Information Systems Committee, the Biotechnology and Biological Sciences Research Council, and the Engineering and Physical Sciences Research Council.

References

1. Kudo, T., Matsumoto, Y.: Chunking with support vector machines. In: Proceedings of NAACL 2001. (2001) 192–199
2. Bikel, D.M.: Intricacies of collins' parsing model. Computational Linguistics **30** (2004) 479–511
3. Kulick, S., Bies, A., Libeman, M., Mandel, M., McDonald, R., Palmer, M., Schein, A., Ungar, L.: Integrated annotation for biomedical information extraction. In: Proceedings of HLT/NAACL-2004. (2004)
4. Tateisi, Y., Tsujii, J.: Part-of-speech annotation of biology research abstracts. In: Proceedings of 4th International Conference on Language Resource and Evaluation (LREC2004). (2004) 1267–1270
5. Brants, T.: TnT – a statistical part-of-speech tagger. In: Proceedings of the 6th Applied NLP Conference (ANLP). (2000)
6. Ohta, T., Tateisi, Y., Kim, J.D., Tsujii, J.: Genia corpus: an annotated research abstract corpus in molecular biology domain. In: Proceedings of the Human Language Technology Conference (HLT 2002). (2002)
7. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: The penn treebank. Computational Linguistics **19** (1994) 313–330
8. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of HLT-NAACL 2003. (2003) 252–259
9. Gimenez, J., Marquez, L.: Fast and accurate part-of-speech tagging: The SVM approach revisited. In: Proceedings of RANLP 2003. (2003) 158–165
10. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: Proceedings of EMNLP 2002. (2002) 1–8
11. Kazama, J., Tsujii, J.: Evaluation and extension of maximum entropy models with inequality constraints. In: Proceedings of EMNLP 2003. (2003)
12. Chen, S.F., Rosenfeld, R.: A gaussian prior for smoothing maximum entropy models. Technical Report CMUCS -99-108, Carnegie Mellon University (1999)
13. Ratnaparkhi, A.: A maximum entropy model for part-of-speech tagging. In: Proceedings of EMNLP 1997. (1997)