Two-Phase Biomedical Named Entity Recognition Using A Hybrid Method

Seonho Kim[1], Juntae Yoon[2], Kyung-Mi Park[1], and Hae-Chang Rim[1]

[1] Dept. of Computer Science and Engineering,
Korea University, Seoul, Korea
[2] NLP Lab. Daumsoft Inc. Seoul, Korea

Abstract. Biomedical named entity recognition (NER) is a difficult problem in biomedical information processing due to the widespread ambiguity of terms out of context and extensive lexical variations. This paper presents a two-phase biomedical NER consisting of term boundary detection and semantic labeling. By dividing the problem, we can adopt an effective model for each process. In our study, we use two exponential models, conditional random fields and maximum entropy, at each phase. Moreover, results by this machine learning based model are refined by rule-based postprocessing implemented using a finite state method. Experiments show it achieves the performance of F-score 71.19% on the JNLPBA 2004 shared task of identifying 5 classes of biomedical NEs.

1 Introduction

Due to dynamic progress in biomedical literature, a vast amount of new information and research results have been published and many of them are available in the electronic form - for example, like the PubMed MedLine database. Thus, automatic knowledge discovery and efficient information access are strongly demanded to curate domain databases, to find out relevant information, and to integrate/update new information across an increasingly large body of scientific articles. In particular, since most biomedical texts introduce specific notations, acronyms, and innovative names to represent new concepts, relations, processes, functions, locations, and events, automatic extraction of biomedical terminologies and mining of their diverse usage are major challenges in biomedical information processing system. In these processes, biomedical named entity recognition (NER) is the core step to access the higher level of information.

In fact, there has been a wide range of research on NER like the NER task on the standard newswire domain in the Message Understanding Conference (MUC-6). In this task, the best system reported 95% accuracy in identifying seven types of named entities (person, organization, location, time, date, money, and percent). While the performance in the standard domain turned out to be quite good as shown in the papers, that in the biomedical domain is not still satisfactory, which is mainly due to the following characteristics of biomedical terminologies: First, NEs have various naming conventions. For instance, some entities have descriptive and expanded forms such as "*activated B cell lines, 47 kDa sterol*

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 646–657, 2005.

regulatory element binding factor", whereas some entities appear in shortened or abbreviated forms like "*EGFR*" and "*EGF receptor*" representing epidermal growth factor receptor. Second, biomedical NEs have the widespread ambiguity out of context. For instance, "*IL-2*" can be doubly classified as "protein" and "DNA" according to its context. Third, biomedical NEs often comprise a nested structure, for example "*⟨DNA⟩⟨protein⟩TNF alpha⟨/protein⟩gene⟨/DNA⟩*". According to [13], 16.57% of biomedical terms in GENIA have cascaded constructions. In the case, recognition of the longest terms is the main target in general. However, in our evaluation task, when the embedded part of a term is regarded as the meaningful or important class in the context, the term is labeled only with the class of embedded one. Thus, identification of internal structures of NEs is helpful to recognize correct NEs. In addition, more than one NE often share the same head noun with a conjunction/disjunction or enumeration structure, for instance, "*IFN-gamma and GM-CSF mRNA*", "*CD33+, CD56+, CD16- acute leukemia*"or "*antigen- or cAMP-activated Th2 cell*". Last, there is a lot of inter-annotator disagreement. [7] reported that the inter-annotator agreement rate of human experts was just 77.6% when performing gene/protein/mRNA classification task manually.

Thus, a lot of term occurrences in real text would not be identified with simple dictionary look-up, despite the availability of many terminological databases, as claimed in [12]. That is one of the reasons why machine learning approaches are more dominant in biomedical NER than rule-based or dictionary-based approaches [5], even though existence of reliable training resources is very critical.

Accordingly, much work has been done on biomedical NER, based on machine learning techniques. [3] and [13] have used hidden Markov Model (HMM) for biomedical NER where state transitions are made by semantic trigger features. [4] and [11] have applied maximum entropy plus Markovian sequence based models such as maximum entropy markov model (MEMM) and conditional random fields (CRFs), which present a way for integrating different features such as internal word spellings and morphological clues within an NE string and contextual clues surrounding the string in the sentence.

These works took an one-phase based approach where boundary detection of named entities and semantic labeling come together. On the other hand, [9] proposed a two-phase model in which the biomedical named entity recognition process is divided into two processes of distinguishing biomedical named entities from general terms and labeling the named entities with semantic classes that they belong to. They use support vector machines (SVM) for each phase. However, the SVM does not provide an easy way for labeling Markov sequence data like B following O and I following B in named entities. Furthermore, since this system is tested on the GENIA corpus rather than JNLPBA 2004 shared task, we cannot confirm the effectiveness of this approach on the ground of experiments for common resources.

In this paper, we present a two-phase named entity recognition model: (1) boundary detection for NEs and (2) term classification by semantic labeling. The advantage of dividing the recognition process into two phase is that we can select separately a discriminative feature set for each subtask, and moreover can measure effectiveness of models at each phase. We use two exponential models for this work, namely conditional random fields for boundary detection having Markov sequence, and the maximum entropy model for semantic labeling. In addition, results from the machine learning based model are refined by a rule-based postprocessing, which is implemented using a finite state transducer (FST). The FST is constructed with the GENIA 3.02 corpus. We here focus on identification of five classes of NEs, i.e. "protein", "RNA", "DNA", "cell line", and "cell type" and experiments are conducted on the training and evaluation set provided by the shared task in COLING 2004 JNLPBA.

2 Training

2.1 Maximum Entropy and Conditional Random Fields

Before we describe the features used in our model, we briefly introduce the ME and CRF model which we make use of. In the ME framework, the conditional probability of predicting an outcome o given a history h is defined as follows:

$$p_\lambda(o|h) = \frac{1}{Z_\lambda(h)} exp\left(\sum_{i=1}^{k} \lambda_i f_i(h,o)\right) \tag{1}$$

where $f_i(h,o)$ is a binary-valued feature function, λ_i is the weighting parameter of $f_i(h,o)$, k is the number of features, and $Z_\lambda(h)$ is a normalization factor for $\Sigma_o p_\lambda(o|h)$=1. That is, the probability $p_\lambda(o|h)$ is calculated by the weighted sum of active features. Given an exponential model with k features and a set of training data, empirical distribution, weights of the k features are trained to maximize the model's log-likelihood:

$$L(p) = \sum_{o,h} \tilde{p}(h,o) log(o|h) \tag{2}$$

Although the maximum entropy model above provides a powerful tool for classification by integrating different features, it is not easy to model the Markov sequence data. In this case, the CRF is used for a task of assigning label sequences to a set of observation sequences. Based on the principle of maximum entropy, a CRF has a single exponential model for the joint probability of the entire sequence of labels given the observation sequence. The CRF is a special case of the linear chain that corresponds to conditionally trained finite-state machine and define conditional probability distributions of a particular label sequence **s** given observation sequence **o**

$$\begin{aligned} p_\lambda(\mathbf{s}|\mathbf{o}) &= \tfrac{1}{Z(\mathbf{o})} exp(\textstyle\sum_{j=1}^{k} \lambda_j F_j(\mathbf{s},\mathbf{o})) \\ F_j(\mathbf{s},\mathbf{o}) &= \textstyle\sum_{i=1}^{n} f_j(s_{i-1}, s_i, \mathbf{o}, i) \end{aligned} \tag{3}$$

where $\mathbf{s} = s_1 \ldots s_n$, and $\mathbf{o} = o_1 \ldots o_n$, $Z(\mathbf{o})$ is a normalization factor, and each feature is a transition function [8]. For example, we can think of the following feature function.

$$f_j(s_{i-1}, s_i, \mathbf{o}, i) = \begin{cases} 1 \text{ if } s_{i-1}\text{=B and } s_i\text{=I,} \\ \quad \text{and the observation word at position i is ``}\textit{gene}\text{''} \\ 0 \; \textit{otherwise} \end{cases} \tag{4}$$

Our CRFs for term boundary detection have a first-order Markov dependency between output tags. The label at position i, s_i is one of B, I and O. In contrast to the ME model, since B is the beginning of a term, the transition from O to I is not possible. CRFs constrain results to consider only reasonable paths. Thus, total 8 combinations are possible for $(s_{i-1}$,$s_i)$ and the most likely $\mathbf{s}$ can be found with the Viterbi algorithm. The weights are set to maximize the conditional log likelihood of labeled sequences in the training set using a quasi-Newton method called L-BFGS [2].

2.2 Features for Term Boundary Detection

Table 1 shows features for the step of finding the boundary of biomedical terms. Here, we give a supplementary description of a part of the features.

Table 1. Feature set for boundary detection (+:conjunction)

Model	Feature	Description
CRF, ME_{markov}	Word	$w_{i-1}, w_{i-2}, w_i, w_{i+1}, w_{i+2}$
CRF, ME_{markov}	Word Normalization	normalization forms of the 5 words
CRF, ME_{markov}	POS	$POS_{w_{i-1}}$, POS_{w_i}, $POS_{w_{i+1}}$
CRF, ME_{markov}	Word Construction form	WF_{w_i}
CRF, ME_{markov}	Word Characteristics	$WC_{w_{i-1}}, WC_{w_i}, WC_{w_{i+1}}$
CRF, ME_{markov}	Contextual Bigrams	$w_{i-1} + w_i$
		$w_i + w_{i+1}$
		$w_{i+1} + w_{i+2}$
CRF, ME_{markov}	Contextual Trigrams	$w_{i-1} + w_i + w_{i+1}$
CRF, ME_{markov}	Bigram POS	$POS_{w_{i-1}} + POS_{w_i}$
		$POS_{w_i} + POS_{w_{i+1}}$
CRF, ME_{markov}	Trigram POS	$POS_{w_{i-1}} + POS_{w_i} + POS_{w_{i+1}}$
CRF, ME_{markov}	Modifier	MODI(w_i)
CRF, ME_{markov}	Header	HEAD(w_i)
CRF, ME_{markov}	SUFFIX	SUFFIX(w_i)
CRF, ME_{markov}	Chunk Type	$CType_{w_i}$
CRF, ME_{markov}	Chunk Type + Pre POS	$CType_{w_i} + POS_{w_{i-1}}$
ME_{markov}	Pre label	$label_{w_{i-1}}$
ME_{markov}	Pre label + Cur Word	$label_{w_{i-1}} + w_i$

- **word and POS:** 5 words(target word(w_i), left two words, and right two words) and three POS($POS_{w_{i-1}}$, POS_{w_i}, $POS_{w_{i+1}}$) are considered.

- **word normalization:** This feature contributes to word normalization. We attempt to reduce a word to its stem or root form with a simple algorithm which has rules for words containing plural, hyphen, and alphanumeric letters. Specifically, the following patterns are considered.

  (1) "lymphocytes", "cells" → "lymphocyte", "cell"
  (2) "il-2", "il-2a", "il2a" → "il"
  (3) "5-lipoxygenase", "v-Abl" → "lipoxygenase", "abl"
  (4) "peri-kappa" or "t-cell" has two normalization forms of "peri" and "kappa" and "t" and "cell" respectively.
  (5) "Ca2+-independent" has two roots of "ca" and "independent".
  (6) The root of digits is "D".

- **informative suffix:** This feature appears if a target word has a salient suffix for boundary detection. The list of salient suffixes is obtained by relative entropy [10].

- **word construction form:** This feature indicates how a target word is orthographically constructed. Word shapes refer to a mapping of each word on equivalence classes that encodes with dashes, numerals, capitalizations, lower letters, symbols, and so on. All spellings are represented with combinations of the attributes[1]. For instance, the word construction form of "*IL-2*" would become "IDASH-ALPNUM".

- **word characteristics:** This feature appears if a word represents a DNA sequence of "A","C","G","T" or Greek letter such as beta or alpha, ordinal index such as I, II or unit such as BU/ml, micron/mL. It is encoded with "ACGT", "GREEK", "INDEX", "UNIT".

- **head/modifying information:** If a word prefers the rightmost position of terminologies, we regard it has the property of a head noun. On the other hand, if a word frequently occurs in other positions, we regard it has the property of a modifying noun. It can help to establish the beginning and ending point of multi-word entities. We automatically extract 4,382 head nouns and 7,072 modifying nouns from the training data as shown in Table 2.

- **chunk-type information:** This feature is also effective in determining the position of a word in NEs, "B", "I", "O" which means "begin chunk", "in chunk" and "others", respectively. We consider the chunk type of a target word and the conjunction of the current chunk type and the POS of the previous word to represent the structure of an NE.

We also tested an ME-based model for boundary detection. For this, we add two special features : previous state (label) and conjunction of previous label

[1] "IDASH" (inter dash), "EDASH" (end dash), "SDASH" (start dash), "CAP"(capitalization), "LOW"(lowercase), "MIX"(lowercase and capitalization letters), "NUM"(digit), "ALPNUM"(alpha-numeric), "SYM"(symbol), "PUNC"(punctuation),and "COMMA"(comma)

Table 2. Examples of Head/Modifying Nouns

Modifying Nouns	Head Nouns
nf-kappa	cytokines
nuclear	elements
activated	assays
normal	complexes
phorbol	macrophages
viral	molecules
inflammatory	pathways
murine	extracts
electrophoretic	glucocorticoids
acute	levels
intracellular	responses
epstein-barr	clones
cytoplasmic	motifs

and current word to consider **state transition**. That is, a previous label can be represented as a feature function in our model as follows:

$$f_i(h,o) = \begin{cases} 1\ if \text{ pre\_label+tw=B+gene,o=I} \\ 0\ otherwise \end{cases} \tag{5}$$

It means that the target word is likely to be inside a term (I), when the word is "gene" and the previous label is "B". In our model, the current label is deterministically assigned to the target word with considering the previous state with the highest probability.

2.3 Features for Semantic Labeling

Table 3 shows features for semantic labeling with respect to recognized NEs.

- **word contextual feature:** We make use of three kinds of internal and external contextual features: words within identified NEs, their word normalization forms, and words surrounding the NEs. In Table 3, NE_{w_0} denotes the rightmost word in an identified NE region. Moreover, the presence of specific head nouns acting as functional words takes precedence when determining the term class, even though many terms do not contain explicit term category information. For example, functional words, such as "*factor*", "*receptor*", and "*protein*" are very useful in determining *protein* class, and "*gene*", "*promoter*", and "*motif*" are clues for classifying *DNA* [5]. In general, such functional words are often the last word of an entity. This is the reason we consider the position where a word occurs in NEs along with the word. For inside context features, we use non-positional word features as well. As non-positional features, all words inside NEs are used.
- **internal bigrams and trigrams:** We consider the rightmost bigrams/ trigrams inside identified NEs and the normalized bigrams/trigrams.

Table 3. Feature Set for Semantic Classification

Feature	description
Word Features (positional)	$NE_{w_{others}}, NE_{w_{-3}}, NE_{w_{-2}}, NE_{w_{-1}}, NE_{w_0}$
Word Features (non-positional)	All_{NE_w}
Word Normalization (positional)	$WF_{NE_{w_{-3}}}, WF_{NE_{w_{-2}}}, WF_{NE_{w_{-1}}}, WF_{NE_{w_0}}$
Left Context(Words Surrounding an NE)	LCW_{-2}, LCW_{-1}
Right Context	RCW_{+1}, RCW_{+2}
Internal Bigrams	$NE_{w_{-1}} + NE_{w_0}$
Internal Trigrams	$NE_{w_{-2}} + NE_{w_{-1}} + NE_{w_0}$
Normalized Internal Bigrams	$WF_{NE_{w_{-1}}} + WF_{NE_{w_0}}$
Normalized Internal Trigrams	$NE_{w_{-2}} + NE_{w_{-1}} + NE_{w_0}$
IDASH-word related Bigrams/Trigrams	
Keyword	KEYWORD(NE_i)

- **IDASH-word related bigrams/trigrams:** This feature appears if NE_{w_0} or $NE_{w_{-1}}$ contains dash characters. In this case, the bigram/trigram are additionally formed by removing all dashes from the spelling. It is useful to deal with lexical variants.
- **keywords:** This feature appears if the identified NE is informative keyword with respect to a specific class. The keywords set comprises terms obtained by the relative entropy between general and biomedical domain corpora.

3 Rule-Based Postprocessing

A rule-based method can be used to correct errors by NER based on machine learning. For example, the CRFs tag "*IL-2 receptor expression*" as "B I I", since the NEs ended with "*receptor expression*" in training data almost belong to "*other_name*" class even if the NEs ended with "*receptor*" belong to "*protein*" class. It should be actually tagged as "B I O". That kind of errors is caused mainly by the cascaded phenomenon in biomedical names. Since our system considers all NEs belonging to other classes in the recognition phase, it tends to recognize the longest ones. That is, in the term classification phase, such NEs are classified as "other" class and are ignored. Thus, the system losts embedded NEs although the training and evaluation set in fact tends to consider only the embedded NE when the embedded one is more meaningful or important.

This error correction is conducted by the rule-based method, i.e. **If** *condition* **THEN** *action*. For example, the rule '**IF** w_{i-2}=IL-2, w_{i-1}=receptor and w_i=expression **THEN** replace the tag of w_i with O' can be applied for the above case. We use a finite state transducer for this rule-based transformation, which is easy to understand with given lexical rules, and very efficient. Rules used for the FST are acquired from the GENIA corpus. We first retrieved all NEs including embedded NEs and longest NEs from GENIA 3.02 corpus and change

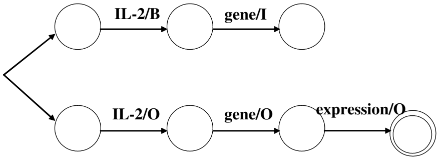

Fig. 1. Non-Deterministic FST

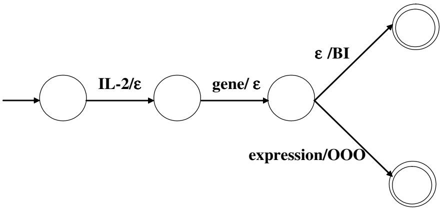

Fig. 2. Deterministic FST

the outputs of all other classes except the target 5 classes to O. That is, the input of FST is a sequence of words in a sentence and the output is categories corresponding to the words.

Then, we removed the rules in conflict with NE information from the training corpus. These rules are non-deterministic (Figure 1), and we can change it to the deterministic FST (Figure 2) since the lengths of NEs are finite. The deterministic FST is made by defining the final output function for the deterministic behavior of the transducer, delaying the output. The deterministic FST is defined as follows: $(\Sigma_1, \Sigma_2, Q, i, F, \otimes, *, \rho)$, where Σ_1 is a finite input alphabet; Σ_2 is a finite output alphabet; Q is a finite set of states or vertices; $i \in Q$ is the initial state; $F \subseteq Q$ is the set of final states; $\otimes$ is the deterministic state transition function that maps $Q \times \Sigma_1$ on Q; $*$ is the deterministic emission function that maps $Q \times \Sigma_1$ on Σ_2^* and $\rho : F \rightarrow \Sigma_2^*$ is the final output function for the deterministic behavior of the transducer.

4 Evaluation

4.1 Experimental Environments

In the shared task, only biomedical named entities which belong to 5 specific classes are annotated in the given training data. That is, terms belonging to other classes in GENIA are excluded from the recognition target. However, we consider all NEs in the boundary detection step since we separate the NER task into two phases. Thus, in order to utilize other class terms, we additionally annotated “O” class words in the training data where they corresponds to other classes such as *other_organic_compound*, *lipid*, and *multi_cell* in GENIA 3.02p version corpus. During the annotation, we only consider the longest NEs on

Table 4. Number of training examples

RNA	DNA	cell_line	cell_type	protein	other
472	5,370	2,236	2,084	16,042	11,475

GENIA. As a consequence, we find all biomedical named entities in text at the term detection phase. Then, biomedical NEs classified as *other* class are changed to *O* at the semantic labeling phase. The total words that belong to *other* class turned out to be 25,987. Table 4 shows the number of NEs with respect to each class on the training data. In our experiments, a quasi-Newton method called the L-BFGS with Gaussian Prior smoothing is applied for parameter estimation [2].

4.2 Experimental Results

Table 5 shows the overall performance on the evaluation data. Our system achieves an F-score of 71.19%. As shown in the table, the performance of NER for cell_line class was not good, because its boundary recognition is not so good as other classes. Also, Table 6 shows the results of semantic classification. In particular, the system often confuses *protein* with *DNA*, and *cell_line* with *cell_type*. Among the correctly identified 7,093 terms, 790 terms were misclassified.

Table 7 shows the performance of each phase. Our system obtains 76.88% F-score in the boundary detection task and, using 100% correctly recognized terms from annotated test data, 90.54% F-score in the semantic classification task. Currently, since we cannot directly assess the accuracy of the term detection process on the evaluation set because of *other* class words, the 75% of the training data were used for training and the rest for testing.

Table 5. Overall performance on the evaluation data

	Fully Correct			Left Correct	Right Correct
Class	**Recall**	**Precision**	**F-score**	**F-score**	**F-score**
protein	76.30	69.71	72.85	77.60	79.15
DNA	67.80	64.91	66.33	68.36	74.57
RNA	73.73	63.04	67.97	71.09	74.22
cell_line	57.40	54.88	56.11	59.04	65.69
cell_type	70.12	77.64	73.69	74.89	81.51
overall	**72.77**	**69.68**	**71.19**	**74.75**	**78.23**

Table 6. Confusion matrix over evaluation data

gold/sys	protein	DNA	RNA	cell_line	cell_type	other
protein	0	72	3	1	4	267
DNA	97	0	0	0	0	49
RNA	11	0	0	0	0	0
cell_line	10	1	0	0	63	37
cell_type	21	0	0	92	0	57

Table 7. Performance of term detection and semantic classification

	Recall	**Precision**	**F-score**
term detection (ME_{Markov})	74.03	75.31	74.67
term detection (CRF)	76.14	77.64	76.88
semantic classification	87.50	93.81	90.54
overall NER	72.77	69.68	71.19

Table 8. Performance of NE recognition methods (one-phase vs. two-phase)

method	**Recall**	**Precision**	**F-score**
one-phase	64.23	63.13	63.68
two-phase(*baseline*2) (only 5 classes)	66.24	64.54	65.38
two-phase(*baseline*2) **(5 classes+other class)**	68.51	67.58	68.04

Also, we compared our model with the one-phase model. The detailed results are presented in Table 8. Both of them have pros and cons. The best-reported system presented by [13] uses one-phase strategy. In our evaluation, the two-phase method shows a better result than the one-phase method, although direct comparison is not possible since we tested with a maximum entropy based exponential models in all cases. The features for one-phase method are identical with the recognition features except that the local context of a word is extended as previous 4 words and next 4 words. In addition, we investigate whether the consideration of "other" class words is helpful in the recognition performance. Table 8 shows explicit annotations of other NE classes much improve the performance of existing entity types.

In the next experiment, we test how individual methods have an effect on the performance in the term detection step. Table 9 shows the results obtained by combining different methods in the NER process. At the semantic labeling phase, all methods employed the ME model using the features described in 2.3. Baseline1 is the two-phase ME model which restrict the inspection of NE candidates to the NPs which include at least one biomedical salient word. Baseline2 is the two-phase ME model considering all words. In order to retrieve domain salient words, we utilized a relative frequency ratio of word distribution in the domain corpus and that in the general corpus [10]. We used the Penn II raw corpus as out-of-domain corpus. Both models do not use the features related to previous labels. As a result, usage of salient words decrease the performance and it only speeds up the training process. Baseline2+FST indicates boundary extension/contraction using FST are applied as postprocessing step in baseline2 recognition. In addition, we compared use of CRFs and ME with Markov process features. For this, we added features of previous labels to the feature set for ME. Baseline2+ME_{Markov} is the two-phase ME model considering all features including previous label related features. Baseline2+CRF is a model exploiting CRFs and baseline2+CRF+FST is a model using CRF and FST as postprocessing. As shown in Table 9, the CRFs based

Table 9. F-score for different methods

Method	Recall	Precision	F-score
$baseline1(salientNP)$	66.21	66.34	66.27
$baseline2(all)$	68.51	67.58	68.04
$baseline2 + FST$	68.89	68.53	68.71
$baseline2 + ME_{Markov}$	70.30	67.65	68.95
$baseline2 + ME_{Markov} + FST$	70.61	68.40	69.49
$baseline2 + CRF$	72.44	68.77	70.56
$baseline2 + CRF + FST$	72.77	69.68	71.19

Table 10. Comparisons with other systems

System	Precision	Recall	F-score
Zhou et. al (2004)	69.42	75.99	72.55
Our system	72.77	69.68	71.19
Finkel et. al (2004)	71.62	68.56	70.06
Settles (2004)	70.0	69.0	69.5

model outperforms the ME based model. Our system reached F-score 71.19% on the $baseline2 + CRF + FST$ model.

Table 10 shows the comparison with top-ranked systems in JNLPBA 2004 shared task. The top-ranked systems made use of external knowledge from gazetteers and abbreviation handling routines, which were reported to be effective. Zhou et. al reported the usage of gazetteers and abbreviation handling improves the performance of the NER system by 4.8% in F-score [13]. Finkel et. al made use of a number of external resources, including gazetteers, web-querying, use of the surrounding abstract, abbreviation handling, and frequency counts from BNC corpus [4]. Settles utilized semantic domain knowledge of 17 kinds of lexicons [11]. Although the performance of our system is a bit lower than the best system, the results are very promising since most systems use external gazetteers, and abbreviation and conjunction/disjunction handling scheme. This suggests areas for further work.

5 Conclusion and Discussion

We presented a two-phase biomedical NE recognition model, term boundary detection and semantic labeling. We proposed two exponential models for each phase. That is, CRFs are used for term detection phase including Markov process and ME is used for semantic labeling. The benefit of dividing the whole process into two processes is that, by separating the processes with different characteristics, we can select separately the discriminative feature set for each subtask, and moreover measure effectiveness of models at each phase. Furthermore, we use the rule-based method as postprocessing to refine the result. The rules are extracted from the GENIA corpus, which is represented by the deterministic FST. The rule-based approach is effective to correct errors by cascading structures of biomedical NEs. The experimental results are quite promising. The system achieved 71.19% F-score without Gazetteers or abbreviation handling process. The performance could be improved by utilizing lexical database and testing various classification models.

Acknowledgements

This work was supported by Korea Research Foundation Grant, KRF-2004-037-D00017.

References

1. Thorten Brants. TnT A Statistical Part-of-Speech Tagger. In *Proceedings of the 6th Applied Natural Language Processing.*; 2000.
2. Stanley F. Chen and Ronald Rosenfeld. A Gaussian prior for smoothing maximum entropy models. *Technical Report CMUCS-99-108, Carnegie Mellon University.*
3. Nigel Collier, Chikashi Nobata and Jun-ichi Tsujii. Extracting the Names of Genes and Gene Products with a Hidden Markov Model. In *Proceedings of COLING 2000*; 201-207.
4. Jenny Finkel, Shipra Dingare, and Huy Nguyen. Exploiting Context for Biomedical Entity Recognition From Syntax to thw Web. In *Proceedings of JNLPBA/BioNLP 2004*; 88-91.
5. K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: identifying protein names from biological papers. In *Proceedins of the Pacific Symposium on Biocomputing 98*; 707-718.
6. Junichi Kazama, Takaki Makino, Yoshihiro Ohta and Junichi Tsujii. Tuning Support Vector Machines for Biomedical Named Entity Recognition, *Proceedings of the ACL Workshop on Natural Language Processing in the Biomedical Domain* 2002; 1-8.
7. Michael Krauthammer and Goran Nenadic. Term Identification in the Biomedical literature. Journal of Biomedical Informatics. 2004; 37(6):512-526.
8. John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML-01*; 282-289.
9. Ki-Joong Lee, Young-Sook Hwang, Seonho Kim, Hae-Chang Rim. Biomedical named entity recognition using two-phase model based on SVMs. *Journal of Biomedical Informatics* 2004; 37(6):436-447.
10. Kyung-Mi Park, Seonho Kim, Ki-Joong Lee, Do-Gil Lee, and Hae-Chang Rim. Incorportating Lexical Knowledge into Biomedical NE Recognition. In *Proceedings of Natural Language Processing in Biomedicine and its Applications Post-COLING Workshop* 2004; 76-79.
11. Burr Settles. Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets. In *Proceedings of JNLPBA/BioNLP* 2004; 104-107.
12. Olivia Tuason, Lifeng Chen, Hongfang Liu, Judith A. Blake, Carol Friedman. Biological Nomenclatures: A Source of Lexical Knowledge and Ambiguity. In *Pacific Symposium on Biocomputing* 2004; 238-249.
13. GuoDong Zhou, Jie Zhang, Jian Su, Chew-Lim Tan. Exploring Deep Knowledge Resources in Biomedical Name Recognition. In *Proceedings of JNLPBA/BioNLP* 2004; 99-102.