Two-Phase Biomedical NE Recognition based on SVMs

Ki-Joong Lee   Young-Sook Hwang   and Hae-Chang Rim
Department of Computer Science & Engineering
Korea University
1, 5-ka, Anam-dong, SEOUL, 136-701, KOREA
{kjlee, yshwang, rim}@nlp.korea.ac.kr

Abstract

Using SVMs for named entity recognition, we are often confronted with the multi-class problem. Larger as the number of classes is, more severe the multi-class problem is. Especially, one-vs-rest method is apt to drop the performance by generating severe unbalanced class distribution. In this study, to tackle the problem, we take a two-phase named entity recognition method based on SVMs and dictionary; at the first phase, we try to identify each entity by a SVM classifier and post-process the identified entities by a simple dictionary look-up; at the second phase, we try to classify the semantic class of the identified entity by SVMs. By dividing the task into two subtasks, i.e. the entity identification and the semantic classification, the unbalanced class distribution problem can be alleviated. Furthermore, we can select the features relevant to each task and take an alternative classification method according to the task. The experimental results on the GENIA corpus show that the proposed method is effective not only in the reduction of training cost but also in performance improvement: the identification performance is about $79.9(F_\beta = 1)$, the semantic classification accuracy is about $66.5(F_\beta = 1)$.

1 Introduction

Knowledge discovery in the rapidly growing area of biomedicine is very important. While most knowledge are provided in a vast amount of texts, it is impossible to grasp all of the huge amount of knowledge provided in the form of natural language. Recently, computational text analysis techniques based on NLP have received a spotlight in bioinformatics. Recognizing the named entities such as proteins, DNAs, RNAs, cells etc. has become one of the most fundamental tasks in the biomedical knowledge discovery.

Conceptually, named entity recognition consists of two tasks: identification, which finds the boundaries of a named entity in a text, and classification, which determines the semantic class of that named entity. Many machine learning approaches have been applied to biomedical named entity recognition(Nobata, 1999)(Hatzivalssiloglou, 2001)(Kazama, 2002). However, no work has achieved sufficient recognition accuracy. One reason is the lack of annotated corpora. This is somewhat appeased with announcement of the GENIA corpus v3.0(GENIA, 2003). Another reason is that it is difficult to recognize biomedical named entities by using general features compared with the named entities in newswire articles. In addition, since non-entity words are much more than entity words in biomedical documents, class distribution in the class representation combining a B/I/O tag with a semantic class C is so severely unbalanced that it costs too much time and huge resources, especially in SVMs training(Hsu, 2001).

Therefore, Kazama and his colleagues tackled the problems by tuning SVMs(Kazama, 2002). They splitted the class with unbalanced class distribution into several subclasses to reduce the training cost. In order to solve the data sparseness problem, they explored various features such as word cache features and HMM state features. According to their report, the word cache and HMM state features made a positive effect on the performance improvement. But, not separating the identification task from the semantic classification, they tried to classify the named entities in the integrated process.

By the way, the features for identifying the biomedical entity are different from those for semantically classifying the entity. For example, while orthographical characteristics and a part-of-speech tag sequence of an entity are strongly related to the identification, those are weakly related to the semantic classification. On the other hand, context words seem to provide useful clues to the semantic classification of a given entity. Therefore, we will separate the identification task from the semantic classification task. We try to select different features according to the task. This approach enables us to solve the unbalanced class distribution problem which often occurs in a single complicated approach. Besides, to improve the performance, we will post-process the results of SVM classifiers by utilizing the dictionary. That is, we adopt a simple dictionary lookup method to correct the errors by SVMs in the identification phase.

Through some experiments, we will show how separating the entity recognition task into two subtasks contributes to improving the performance of biomedical named entity recognition. And we will show the effect the hybrid approach of the SVMs and the dictionary-lookup.

2 Definition of Named Entity Classification Problem

We divide the named entity recognition into two subtasks, the identification task which finds the regions of the named entities in a text and the semantic classification which determines the semantic classes of them. Figure 1 illustrates the proposed method, which is called two-phase named entity recognition method.

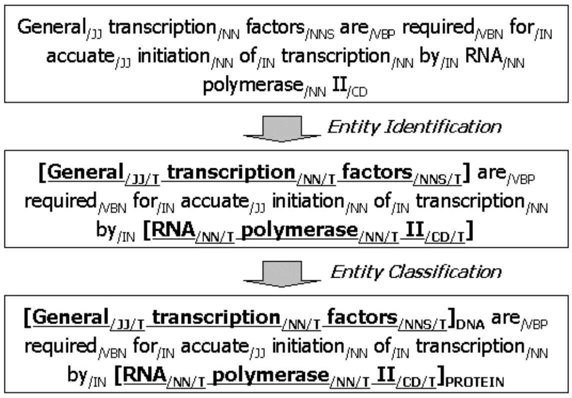

Figure 1: Examples of Biomedical Named Entity Recognition

The identification task is formulated as classification of each word into one of two classes, T *or* O that represent region information. The region information is encoded by using simple T/O representation: T means that current word is a part of a named entity, and O means that the word is not in a named entity. With the representation, we need only one binary SVM classifier of two classes, T, O.

The semantic classification task is to assign one of semantic classes to the identified entity. At the semantic classification phase, we need to classify only the identified entities into one of the N semantic classes because the entities were already identified. Non-entity words are ignored at this phase. The classes needed to be classified are just only the N semantic classes. Note that the number of total classes, $N + 1$ is remarkably small compared with the number, $2N + 1$ required in the complicated recognition approaches in which a class is represented by combining a region information B/I/O with a semantic class C. It can considerably reduce workload in the named entity recognition.

Especially when using SVMs, the number of classes is very critical to the training in the aspect of training time and required resources. Let L be the number of training samples and let N be the number of classes. Then one-vs-rest method takes $N \times O(L)$ in the training step. The complicated approach with the B/I/O notation requires $(2N + 1) \times O(L_{words})$ (L is number of total words in a training corpus). In contrast, the proposed approach requires $(N \times O(L_{entities})) + O(L_{words})$.

Here, $O(L_{words})$ stands for the number of words in a training corpus and $O(L_{entities})$ for the number of entities. It is a considerable reduction in the training cost. Ultimately, it affects the performance of the entity recognizer.

To achieve a high performance of the defined tasks, we use SVM(Joachims, 2002) as a machine learning approach which has showed the best performance in various NLP tasks. And we post-process the classification results of SVMs by utilizing a dictionary. Figure 2 outlines the proposed two-phase named entity recognition system. At each phase, each classifier with SVMs outputs the class of the best score. For classifying multi-classes based on a binary classifier SVM, we use the one-vs-rest classification method and the linear kernel in both tasks.

Furthermore, for correcting the errors by SVMs, the entity-word dictionary constructed from a training corpus is utilized in the identification phase. The dictionary is searched to check whether the boundary words of an identified entity were excluded or not because the boundary words of an entity might be excluded during the entity identification. If a boundary word was excluded, then we concatenate the left or the right side word adjacent to the identified entity. This post-processing may enhance the capability of the entity identifier.

3 Biomedical Named Entity Identification

The named entity identification is defined as the classification of each word to one of the classes that represent the region information. The region information is encoded by using simple T/O representation: T means that the current word is a part of a named entity, and O means that the current word is not in a named entity.

The above representation yields two classes of the task and we build just one binary SVM classifiers for them. By accepting the results of the SVM classifier, we determine the boundaries of an entity. To correct boundary errors, we post-process the identified entities with the entity-word dictionary.

3.1 Features for Entity Identification

An input x to a SVM classifier is a feature representation of a target word to be classified and its context. We use a bit-vector representation. The features of the designated word are composed of orthographical characteristic features, prefix, suffix, and lexical of the word.

Table 1 shows all of the 24 orthographical features. Each feature may be a discriminative feature appeared in biomedical named entites such as protein, DNA and RNA etc. Actually, the name of protein, DNA or RNA is composed by combining alpha-numeric string with several characters such as Greek or special symbols and so on.

Table 1: Orthographical characteristic features of the designated word

Orthographic Feature	examples
DIGITS	1 , 39
SINGLE_CAP	A , M
COMMA	,
PERIOD	.
HYPHON	-
SLASH	/
QUESTION_MARK	?
OPEN_SQUARE	[
CLOSE_SQUARE	]
OPEN_PAREN	(
CLOSE_PAREN	)
COLON	:
SEMICOLON	;
PERCENT	%
APOSTROPHE	’
ETC_SYMBOL	+, *, etc.
TWO_CAPS	alphaCD28
ALL_UPPER	AIDS
INCLUDE_CAPS	c-Jun
GREEK_LETTER	NF-kappa
ALPHA_NUMERIC	p65
ALL_LOWER	motif
CAPS_DIGIT	CD40
INIT_CAP	Rel

And the suffix/prefix, the designated word and the context word features are as follows:

$$w_i = \begin{cases} 1 & \text{if the word is the } i^{th} \text{ word} \\ & \text{in the vocabulary V} \\ 0 & \text{otherwise} \end{cases}$$

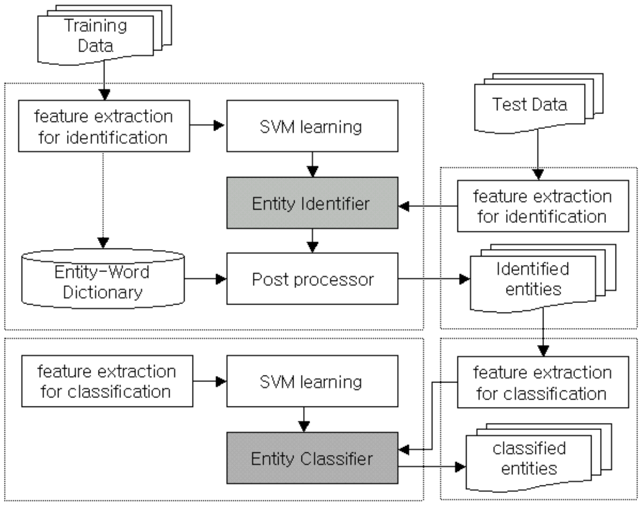

Figure 2: System Configuration of Two Phase Biomedical NE Recognition System

$$pos_i = \begin{cases} 1 & \text{if the word is assigned the } i^{th} \\ & \text{POS tag in the POS tag list} \\ 0 & \text{otherwise} \end{cases}$$

$$suf_i = \begin{cases} 1 & \text{if the word contains the} \\ & i^{th} \text{ suffix in the suffix list} \\ 0 & \text{otherwise} \end{cases}$$

$$pre_i = \begin{cases} 1 & \text{if the word contains the} \\ & i^{th} \text{ prefix in the prefix list} \\ 0 & \text{otherwise} \end{cases}$$

$$w_{ki} = \begin{cases} 1 & \text{if a word at k is the } i^{th} \text{ word} \\ & \text{in the vocabulary V} \\ 0 & \text{otherwise} \end{cases}$$

$$pos_{ki} = \begin{cases} 1 & \text{if a word at k is assigned the} \\ & i^{th} \text{ POS tag in the POS tag list} \\ 0 & \text{otherwise} \end{cases}$$

In the definition, k is the relative word position from the target word. A negative value represents a preceeding word and a positive value represents a following word. Among them, the part-of-speech tag sequence of the word and the context words is a kind of a syntactic rule to compose an entity. And lexical information is a sort of filter to identify an entity which is as possible as semantically cohesive.

3.2 Post-Processing by Dictionary Look-Up

After classifying the given instances, we do post-processing of the identified entities. During the post-processing, we scan the identified entities and examine the adjacent words to those. If the part-of-speech of an adjacent word belongs to one of the group, adjective, noun, or cardinal, then we look up the dictionary to check whether the word is in it or not. If it exists in the dictionary, we include the word into the entity region. The dictionary is constructed of words consisting of the named entities in a training corpora and stopwords are ignored.

Figure 3 illustrates the post-processing algorithm. In Figure 3, the word *cell* adjacent to the left of the identified entity *cycle-dependent transcription*, has the part-of-speech NN and exists in the dictionary. The word *factor* adjacent to the right of the entity has the part-of-speech NN. It exists in the dictionary, too. Therefore, we include the words *cell* and *factor* into the entity region and change the position tags of the words in the entity.

By taking the post-processing method, we can correct the errors by a SVM classifier. It also gives us a great effect of overcoming the low coverage problem of the small-sized entity dictionary.

4 Semantic Classification of Biomedical Named Entity

The objects of the semantic tagging are the entities identified in the identification phase. Each entity is assigned to a proper semantic class by voting the SVM classifiers.

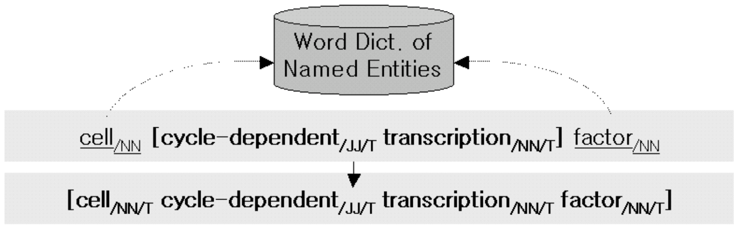

Figure 3: An example of the post-processing of an entity identification

4.1 Features for Semantic Classification

For semantically tagging an entity, an input x to a SVM classifier is represented by a feature vector. The vector is composed of following features:

$$fw_i = \begin{cases} 1 & \text{if a given entity contains one} \\ & \text{of the functional words} \\ 0 & \text{otherwise} \end{cases}$$

$$inw_i = \begin{cases} 1 & \text{if one of the words in the} \\ & \text{entity is in the inside word list} \\ 0 & \text{otherwise} \end{cases}$$

$$lcw_i = \begin{cases} 1 & \text{if noun or verb word in the} \\ & \text{left context is the } i^{th} \text{ word} \\ & \text{in the left context word list} \\ 0 & \text{otherwise} \end{cases}$$

$$rcw_i = \begin{cases} 1 & \text{if noun or verb word in the} \\ & \text{right context is the } i^{th} \text{ word} \\ & \text{in the right context word list} \\ 0 & \text{otherwise} \end{cases}$$

Of the above features, fw_i checks whether the entity contains one of functional words. The functional words are similar to the feature terms used by (Fukuda, 1998). For example, the functional words such as factor, receptor and protein are very helpful to classifying named entities into protein and the functional words such as gene, promoter and motif are very useful for classifying DNA.

In case of the context features of a given entity, we divide them into two kinds of context features, inside context features and outside context features. As inside context features, we take at most three words from the backend of the entity [1]. We make a list of the inside context words by collecting words in the range of the inside context. If one of the three words is the i^{th} word in the inside context word list, we set the inw_i bit to 1. The outside context features are grouped in the left ones and the right ones. For the left and the right context features, we restrict them to noun or verb words in a sentence, whose position is not specified. This grouping make an effect of alleviating the data sparseness problem when using a word as a feature.

For example, given a sentence with the entity, *RNA polymerase II* as follows:

> General transcription factor are required for accurate initiation of transcription by $\underline{\text{RNA polymerase II}}_{PROTEIN}$.

The nouns *transcription, factor, initiation* and the verbs *are, required* are selected as left context features, and the words *RNA, polymerase, II* are selected as inside context features. The bit field corresponding to each of the selected word is set to 1. In this case, there is no right context features. And since the entity contains the functional word *RNA*, the bit field of *RNA* is set to 1.

For classifying a given entity, we build SVM classifiers as many as the number of semantic classes. We take linear kernel and one-vs-rest classification method.

5 Experiments

5.1 Experimental Environments

Experiments have been conducted on the GENIA corpus(v3.0p)(GENIA, 2003), which consists of 2000 MEDLINE abstracts annotated with Penn Treebank (PTB) POS tags. There exist 36 distinct semantic classes in the corpus. However, we used 22 semantic classes which are all but protein, DNA

[1]The average length of entities is about 2.2 in GENIA corpus.

and RNA's subclasses on the GENIA ontology [2]. The corpus was transformed into a B/I/O annotated corpus to represent entity boundaries and a semantic class.

We divided 2000 abstracts into 10 collections for 10-fold cross validation. Each collection contains not only abstracts but also paper titles. The vocabularies for lexical features and prefix/suffix lists were constructed by taking the most frequent 10,000 words from the training part only.

Also, we made another experimental environment to compare with the previous work by (Kazama, 2002). From the GENIA corpus, 590 abstracts(4,808 sentences; 20,203 entities; 128,463 words) were taken as a training part and 80 abstracts(761 sentences; 3,327 entities; 19,622 words) were selected as a test part. Because we couldn't make the experimental environment such as the same as that of Kazama's, we tried to make a comparable environment.

We implemented our method using the SVM-light package(Joachims, 2002). Though various learning parameters can significantly affect the performance of the resulting classifiers, we used the SVM system with linear kernel and default options.

The performance was evaluated by precision, recall and $F_{\beta=1}$. The overall $F_{\beta=1}$ for two models and ten collections, were calculated using 10-fold cross validation on total test collection.

5.2 Effect of Training Data Size

In this experiment, varying the size of training set, we observed the change of $F_{\beta=1}$ in the entity identification and the semantic classification. We fixed the test data with 200 abstracts(1,921 sentences; 50,568 words). Figure 4 shows that the performance was improved by increasing the training set size. As the performance of the identification increases, the gap between the performance of the identification and that of the semantic classification is gradually decreased.

5.3 Computational Efficiency

When using one-vs-rest method, the number of negative samples is very critical to the training in

[2] That is, All of the protein's subclass such as protein_molecule, protein_family_or_group were regarded as protein.

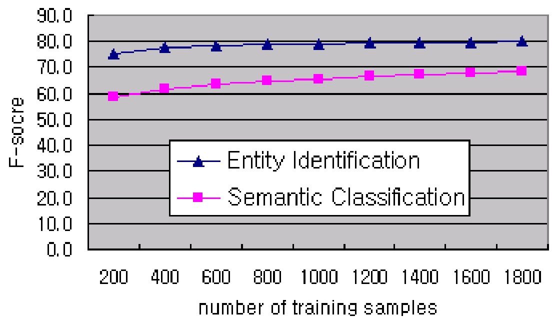

Figure 4: Perfomance shift according to the increase of training data size w/o post-processing

the aspect of training time and required resources. The SVM classifier for entity identifiation determines whether each word is included in an entity or not. Figure 5 shows there are much more negative samples than positive samples in the identification phase. Once entities are identified, non-entity words are not considered in next semantic classification phase. Therefore, the proposed method can effectively remove the unnecessary samples. It enables us effectively save the training costs.

Furthermore, the proposed method could effectively decrease the degree of the unbalance among classes by simplifying the classes. Figure 6 shows how much the proposed method can alleviate the unbalanced class distribution problem compared with 1-phase complicated classification model. However, even though the unbalanced class distribution problem could be alleviated in the identification phase, we are still suffering from the problem in the semantic classification as long as we take the one-vs-rest method. It indicates that we need to take another classification method such as a pairwise method in the semantic classification(Krebel, 1999).

5.4 Discriminative Feature Selection

We subsequently examined several alternatives for the feature sets described in section 3.1 and section 4.1.

The column (A) in Table 2 shows the identification cases. The base feature set consisted of only the designated word and the context words in the range from the left 2 to the right 2. Several alternatives for feature sets were constructed by adding a different combination of features to the base feature set. From

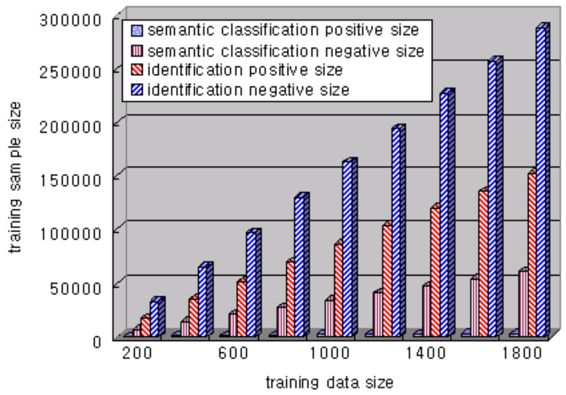

Figure 5: training size vs. positive and negative sample size in identification phase and semantic classification phase

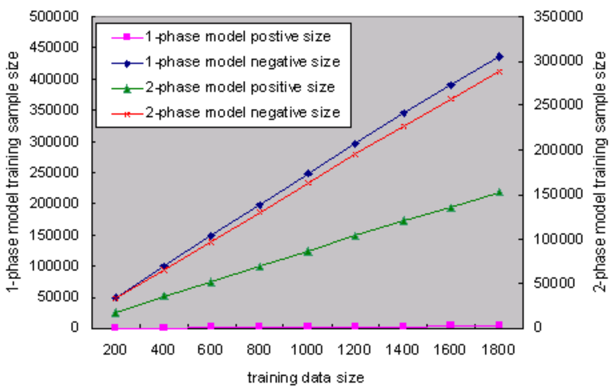

Figure 6: 2-phase model vs. 1-phase model : change of the negative and the positive sample size according to the training data size

(A)		(B)	
FeatSet	F-score	FeatSet	F-score
base	74.6	base(inw)	65.8
pos	77.4 (+2.8)	fw	67.9 (+2.1)
pre	75.0 (+0.4)	lcw	67.9 (+2.1)
suf	75.2 (+0.6)	rcw	67.0 (+1.2)
pre+suf	75.6 (+1.0)	lcw+rcw	66.4 (+0.6)
pos+pre	77.9 (+3.3)	**fw+lcw**	**68.1(+2.3)**
pos+suf	77.9 (+3.3)	fw+rcw	67.1 (+1.3)
all	**77.9 (+3.3)**	all	66.9 (+1.1)

Table 2: Effect of each feature set(training with 900 abstracts, test with 100 abstracts): (A) identification phase, (B) semantic classification phase

Table 2, we can see that part-of-speech information certainly improves the identification accuracy(about +2.8). Prefix and suffix features made a positive effect, but only modestly(about +1.2 on average).

The column (B) in Table 2 shows semantic classification cases with the identification phase of the best performance. We took the feature set composed of the inside words of an entity as a base feature set. And we made several alternatives by adding another features. The experimental results show that functional words and left context features are useful, but right context features are not. Furthermore, part-of-speech information was not effective in the semantic classification while it was useful for the entity identification. That is, when we took the part-of-speech tags of inside context words instead of the inside context words, the performance of the semantic classification was very low($F_{\beta=1.0}$ was 25.1).

5.5 Effect of PostProcessing by Dictionary Lookup

Our two-phase model has the problem that identification errors are propagated to the semantic classification. For this reason, it is necessary to ensure a high accuracy of the boundary identification by adopting a method such as post processing of the identified entities. Table 3 shows that the post processing by dictionary lookup is effective to improving the performance of not only the boundary identification accurary(79.2 vs. 79.9) but also the semantic classification accuracy(66.1 vs. 66.5).

When comparing with the (Kazama, 2002) even though the environments is not the same, the proposed two-phase model showed much better performance in both the entity identification (73.6 vs. 81.4) and the entity classification (54.4 vs. 68.0). One of the reason of the performance improvement is that we could take discriminative features for each subtask by separating the task into two subtasks.

6 Conclusion

In this paper, we proposed a new method of two-phase biomedical named entity recognition based on SVMs and dictionary-lookup. At the first phase, we tried to identify each entity with one SVM classifier and to post-process with a simple dictionary look-up for correcting the errors by the SVM. At the second

Table 3: Performance comparison with or w/o post-processing($F_{\beta=1}$): (A)10-fold cross validation(1800 abstracts, test with 200 abstracts), (B)training with 590 abstracts, test with 80 abstracts

	A			B			(Kazama, 2002)	
	No. of Inst	W/O PostProc	with PostProc	No. of Inst	W/O PostProc	with PostProc	No. of Inst	
Identification		76.2/82.4/79.2	76.8/83.1/**79.9**		78.4/80.8/79.6	80.2/82.6/**81.4**		75.9/71.4/73.6
Classification		63.6/68.8/66.1	64.0/69.2/**66.5**		65.8/67.9/66.8	67.0/69.0/**68.0**		56.2/52.8/54.4
protein	25,276	60.9/79.8/69.1	61.7/78.8/69.2	1,056	61.3/81.3/69.9	62.8/80.7/70.6	709	49.2/66.4/56.5
DNA	8,858	65.1/63.9/64.5	65.0/63.8/64.4	474	71.4/61.0/65.8	72.1/61.6/66.4	460	49.6/37.0/42.3
RNA	683	72.2/71.7/72.0	73.8/72.5/73.1	36	74.4/88.9/81.0	75.6/86.1/80.5		
cell line	3,783	71.6/54.2/61.7	72.3/72.3/72.3	201	73.2/44.8/55.6	73.2/44.8/55.6	121	60.2/46.3/52.3
cell type	6,423	67.2/77.5/72.0	67.5/67.5/67.5	252	64.9/82.1/72.5	65.4/81.7/72.7	199	70.0/75.4/72.6

phase, we tried to classify the identified entity into its semantic class by voting the SVMs. By dividing the task into two subtasks, the identification and the semantic classification task, we could select more relevant features for each task and take an alternative classification method according to the task. This is resulted into the mitigation effect of the unbalanced class distribution problem but also improvement of the performance of the overall tasks.

References

N. Collier, C. Nobata, and J. Tsujii 2000. Extracting the Names of Genes and Gene Products with a Hidden Markov Model. In *Proc. of Coling2000*, pages 201-207.

K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. 1998. Information extraction: identifying protein nmes from biological papers. In *Proc. of the Pacific Symposium on Biocomputing '98(PSB'98).*

GENIA Corpus 3.0p. 2003. available at http://www-tsujii.is.s.u-tokyo.ac.jp/ genia/topics/Corpus/3.0/GENIA3.0p.intro.html

V. Hatzivalssiloglou, P. A. Duboue, and A. Rzhetsky. 2001. Disambiguating proteins, genes, and RNA in text: a machine learning approach. *Bioinformatics. 17 Supple 1.*

C. Hsu and C. Lin. 2001. A comparison on methods for multi-class support vector machines. Technical report, National Taiwan University, Taiwan.

T. Joachims. 1998. Making Large-Scale SVM Learning Practical. LS8-Report, 24, Universitat Dortmund, LS VIII-Report.

T. Joachims. 2000. Estimating the generalization performance of a SVM efficiently. In *Proc. of the Seventeenth International Conference on Machine Learning. Morgan Kaufmann*, pages 431-438.

SVM Light. 2002. available at http://svmlight.joachims.org/

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

U. H.-G Krebel 1999. Pairwise Classification and Support Vector machines. In *B. Scholkopf, C.J.C. Burges, Advances in Kernel Methods: Support Vector Learning*, pp. 255-268, The MIT Press, Cambridge, MA.

C. Nobata, N. Collier, and J. Tsujii. 1999. Automatic term identification and classification in biology texts. In *Proc. of the 5th NLPRS*, pages 369-374.

B.J. Stapley, L.A. Kelley, and M.J.E. Sternberg. 2002. Predicting the Sub-Cellular Location of Proteins from Text Using Support Vector Machines. In *Proc. of Pacific Symposium on Biocomputing 7*, pages 374-385.

Vladimir Vapnik. 1998. Statistical Learning Theory Wiley, New York.