Improving the Scalability of Semi-Markov Conditional Random Fields for Named Entity Recognition

Daisuke Okanohara† Yusuke Miyao† Yoshimasa Tsuruoka ‡ Jun'ichi Tsujii†‡§
†Department of Computer Science, University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan
‡School of Informatics, University of Manchester
POBox 88, Sackville St, MANCHESTER M60 1QD, UK
§SORST, Solution Oriented Research for Science and Technology
Honcho 4-1-8, Kawaguchi-shi, Saitama, Japan
{hillbig,yusuke,tsuruoka,tsujii}@is.s.u-tokyo.ac.jp

Abstract

This paper presents techniques to apply semi-CRFs to Named Entity Recognition tasks with a tractable computational cost. Our framework can handle an NER task that has long named entities and many labels which increase the computational cost. To reduce the computational cost, we propose two techniques: the first is the use of feature forests, which enables us to pack feature-equivalent states, and the second is the introduction of a filtering process which significantly reduces the number of candidate states. This framework allows us to use a rich set of features extracted from the chunk-based representation that can capture informative characteristics of entities. We also introduce a simple trick to transfer information about distant entities by embedding label information into non-entity labels. Experimental results show that our model achieves an F-score of 71.48% on the JNLPBA 2004 shared task without using any external resources or post-processing techniques.

1 Introduction

The rapid increase of information in the biomedical domain has emphasized the need for automated information extraction techniques. In this paper we focus on the Named Entity Recognition (NER) task, which is the first step in tackling more complex tasks such as relation extraction and knowledge mining.

Biomedical NER (Bio-NER) tasks are, in general, more difficult than ones in the news domain. For example, the best F-score in the shared task of Bio-NER in COLING 2004 JNLPBA (Kim et al., 2004) was 72.55% (Zhou and Su, 2004) [1], whereas the best performance at MUC-6, in which systems tried to identify general named entities such as person or organization names, was an accuracy of 95% (Sundheim, 1995).

Many of the previous studies of Bio-NER tasks have been based on machine learning techniques including Hidden Markov Models (HMMs) (Bikel et al., 1997), the dictionary HMM model (Kou et al., 2005) and Maximum Entropy Markov Models (MEMMs) (Finkel et al., 2004). Among these methods, conditional random fields (CRFs) (Lafferty et al., 2001) have achieved good results (Kim et al., 2005; Settles, 2004), presumably because they are free from the so-called label bias problem by using a global normalization.

Sarawagi and Cohen (2004) have recently introduced semi-Markov conditional random fields (semi-CRFs). They are defined on semi-Markov chains and attach labels to the subsequences of a sentence, rather than to the tokens[2]. The semi-Markov formulation allows one to easily construct entity-level features. Since the features can capture all the characteristics of a subsequence, we can use, for example, a dictionary feature which measures the similarity between a candidate segment and the closest element in the dictionary. Kou et al. (2005) have recently showed that semi-CRFs perform better than CRFs in the task of recognition of protein entities.

The main difficulty of applying semi-CRFs to Bio-NER lies in the computational cost at training

[1] Krauthammer (2004) reported that the inter-annotator agreement rate of human experts was 77.6% for bio-NLP, which suggests that the upper bound of the F-score in a Bio-NER task may be around 80%.

[2] Assuming that non-entity words are placed in unit-length segments.

Table 1: Length distribution of entities in the training set of the shared task in 2004 JNLPBA

Length	# entity	Ratio
1	21646	42.19
2	15442	30.10
3	7530	14.68
4	3505	6.83
5	1379	2.69
6	732	1.43
7	409	0.80
8	252	0.49
>8	406	0.79
total	51301	100.00

because the number of named entity classes tends to be large, and the training data typically contain many long entities, which makes it difficult to enumerate all the entity candidates in training. Table 1 shows the length distribution of entities in the training set of the shared task in 2004 JNLPBA. Formally, the computational cost of training semi-CRFs is $O(KLN)$, where L is the upper bound length of entities, N is the length of sentence and K is the size of label set. And that of training in first order semi-CRFs is $O(K^2LN)$. The increase of the cost is used to transfer non-adjacent entity information.

To improve the scalability of semi-CRFs, we propose two techniques: the first is to introduce a filtering process that significantly reduces the number of candidate entities by using a “lightweight” classifier, and the second is to use *feature forest* (Miyao and Tsujii, 2002), with which we pack the feature equivalent states. These enable us to construct semi-CRF models for the tasks where entity names may be long and many class-labels exist at the same time. We also present an extended version of semi-CRFs in which we can make use of information about a preceding named entity in defining features within the framework of first order semi-CRFs. Since the preceding entity is not necessarily adjacent to the current entity, we achieve this by embedding the information on preceding labels for named entities into the labels for non-named entities.

2 CRFs and Semi-CRFs

CRFs are undirected graphical models that encode a conditional probability distribution using a given set of features. CRFs allow both discriminative training and bi-directional flow of probabilistic information along the sequence. In NER, we often use linear-chain CRFs, which define the conditional probability of a state sequence **y** = y_1, ..., y_n given the observed sequence **x** = x_1,...,x_n by:

$$p(\mathbf{y}|\mathbf{x}, \lambda) = \frac{1}{Z(\mathbf{x})} \exp(\Sigma_{i=1}^{n} \Sigma_j \lambda_j f_j(y_{i-1}, y_i, \mathbf{x}, i)), \tag{1}$$

where $f_j(y_{i-1}, y_i, \mathbf{x}, i)$ is a feature function and $Z(\mathbf{x})$ is the normalization factor over all the state sequences for the sequence **x**. The model parameters are a set of real-valued weights $\lambda = \{\lambda_j\}$, each of which represents the weight of a feature. All the feature functions are real-valued and can use adjacent label information.

Semi-CRFs are actually a restricted version of order-L CRFs in which all the labels in a chunk are the same. We follow the definitions in (Sarawagi and Cohen, 2004). Let **s** = $\langle s_1, ..., s_p \rangle$ denote a segmentation of x, where a segment s_j = $\langle t_j$, u_j, $y_j \rangle$ consists of a start position t_j, an end position u_j, and a label y_j. We assume that segments have a positive length bounded above by the pre-defined upper bound L ($t_j \leq u_j$, $u_j - t_j + 1 \leq L$) and completely cover the sequence **x** without overlapping, that is, **s** satisfies $t_1 = 1$, $u_p = |\mathbf{x}|$, and $t_{j+1} = u_j + 1$ for $j = 1, ..., p-1$. Semi-CRFs define a conditional probability of a state sequence **y** given an observed sequence **x** by:

$$p(\mathbf{y}|\mathbf{x}, \lambda) = \frac{1}{Z(\mathbf{x})} \exp(\Sigma_j \Sigma_i \lambda_i f_i(s_j)), \tag{2}$$

where $f_i(s_j) := f_i(y_{j-1}, y_j, \mathbf{x}, t_j, u_j)$ is a feature function and $Z(\mathbf{x})$ is the normalization factor as defined for CRFs. The inference problem for semi-CRFs can be solved by using a semi-Markov analog of the usual Viterbi algorithm. The computational cost for semi-CRFs is $O(KLN)$ where L is the upper bound length of entities, N is the length of sentence and K is the number of label set. If we use previous label information, the cost becomes $O(K^2LN)$.

3 Using Non-Local Information in Semi-CRFs

In conventional CRFs and semi-CRFs, one can only use the information on the adjacent previous label when defining the features on a certain state or entity. In NER tasks, however, information about a distant entity is often more useful than

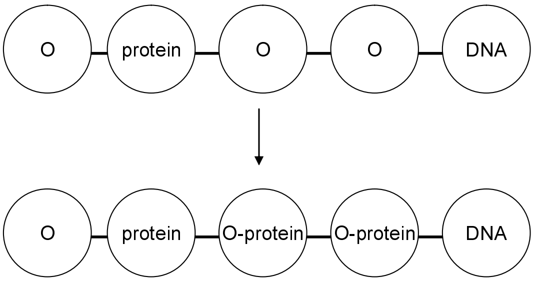

Figure 1: Modification of "**O**" (other labels) to transfer information on a preceding named entity.

information about the previous state (Finkel et al., 2005). For example, consider the sentence "*... including Sp1 and CP1.*" where the correct labels of "*Sp1*" and "*CP1*" are both "protein". It would be useful if the model could utilize the (non-adjacent) information about "*Sp1*" being "protein" to classify "*CP1*" as "protein". On the other hand, information about adjacent labels does not necessarily provide useful information because, in many cases, the previous label of a named entity is "O", which indicates a non-named entity. For 98.0% of the named entities in the training data of the shared task in the 2004 JNLPBA, the label of the preceding entity was "O".

In order to incorporate such non-local information into semi-CRFs, we take a simple approach. We divide the label of "O" into "O-protein" and "O" so that they convey the information on the preceding named entity. Figure 1 shows an example of this conversion, in which the two labels for the third and fourth states are converted from "O" to "O-protein". When we define the features for the fifth state, we can use the information on the preceding entity "protein" by looking at the fourth state. Since this modification changes only the label set, we can do this within the framework of semi-CRF models. This idea is originally proposed in (Peshkin and Pfeffer, 2003). However, they used a dynamic Bayesian network (DBNs) rather than a semi-CRF, and semi-CRFs are likely to have significantly better performance than DBNs.

In previous work, such non-local information has usually been employed at a post-processing stage. This is because the use of long distance dependency violates the locality of the model and prevents us from using dynamic programming techniques in training and inference. Skip-CRFs (Sutton and McCallum, 2004) are a direct implementation of long distance effects to the model. However, they need to determine the structure for propagating non-local information in advance. In a recent study by Finkel et al., (2005), non-local information is encoded using an independence model, and the inference is performed by Gibbs sampling, which enables us to use a state-of-the-art factored model and carry out training efficiently, but inference still incurs a considerable computational cost. Since our model handles limited type of non-local information, i.e. the label of the preceding entity, the model can be solved without approximation.

4 Reduction of Training/Inference Cost

The straightforward implementation of this modeling in semi-CRFs often results in a prohibitive computational cost.

In biomedical documents, there are quite a few entity names which consist of many words (names of 8 words in length are not rare). This makes it difficult for us to use semi-CRFs for biomedical NER, because we have to set L to be eight or larger, where L is the upper bound of the length of possible chunks in semi-CRFs. Moreover, in order to take into account the dependency between named entities of different classes appearing in a sentence, we need to incorporate multiple labels into a single probabilistic model. For example, in the shared task in COLING 2004 JNLPBA (Kim et al., 2004) the number of labels is six ("protein", "DNA", "RNA", "cell_line", "cell_type" and "other"). This also increases the computational cost of a semi-CRF model.

To reduce the computational cost, we propose two methods (see Figure 2). The first is employing a filtering process using a lightweight classifier to remove unnecessary state candidates beforehand (Figure 2 (2)), and the second is the using the *feature forest model* (Miyao and Tsujii, 2002) (Figure 2 (3)), which employs dynamic programming at training "*as much as possible*".

4.1 Filtering with a naive Bayes classifier

We introduce a filtering process to remove low probability candidate states. This is the first step of our NER system. After this filtering step, we construct semi-CRFs on the remaining candidate states using a feature forest. Therefore the aim of this filtering is to reduce the number of candidate states, without removing correct entities. This idea

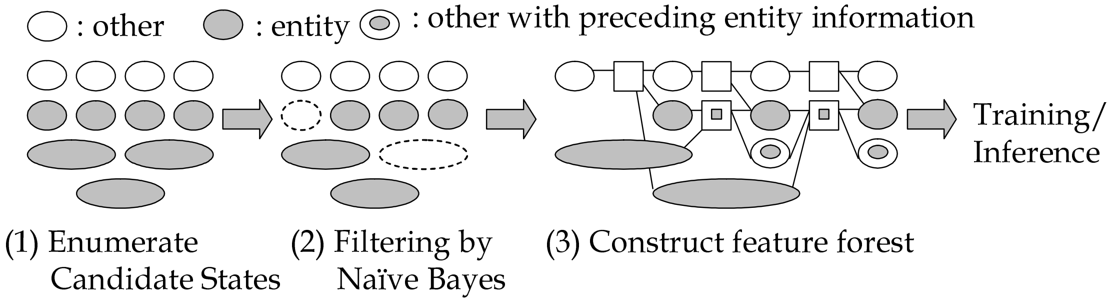

Figure 2: The framework of our system. We first enumerate all possible candidate states, and then filter out low probability states by using a light-weight classifier, and represent them by using feature forest.

Table 2: Features used in the naive Bayes Classifier for the entity candidate: w_s, w_{s+1}, ..., w_e. sp_i is the result of shallow parsing at w_i.

Feature Name	Example of Features
Start/End Word	w_s, w_e
Inside Word	w_s, w_{s+1}, ... , w_e
Context Word	w_{s-1}, w_{e+1}
Start/End SP	sp_s, sp_e
Inside SP	sp_s, sp_{s+1}, ..., sp_e
Context SP	sp_{s-1}, sp_{e+1}

is similar to the method proposed by Tsuruoka and Tsujii (2005) for chunk parsing, in which implausible phrase candidates are removed beforehand.

We construct a binary naive Bayes classifier using the same training data as those for semi-CRFs. In training and inference, we enumerate all possible chunks (the max length of a chunk is L as for semi-CRFs) and then classify those into "entity" or "other". Table 2 lists the features used in the naive Bayes classifier. This process can be performed independently of semi-CRFs

Since the purpose of the filtering is to reduce the computational cost, rather than to achieve a good F-score by itself, we chose the threshold probability of filtering so that the recall of filtering results would be near 100 %.

4.2 Feature Forest

In estimating semi-CRFs, we can use an efficient dynamic programming algorithm, which is similar to the forward-backward algorithm (Sarawagi and Cohen, 2004). The proposal here is a more general framework for estimating sequential conditional random fields.

This framework is based on *the feature forest model*, which was originally proposed for disambiguation models for parsing (Miyao and Tsujii, 2002). A feature forest model is a maximum entropy model defined over *feature forests*, which are abstract representations of an exponential number of sequence/tree structures. A feature forest is an "and/or" graph: in Figure 3, circles represent

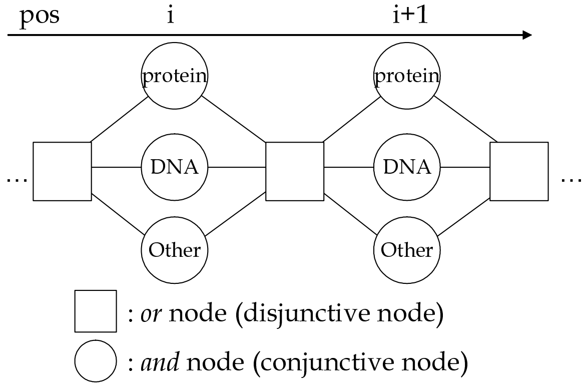

Figure 3: Example of feature forest representation of linear chain CRFs. Feature functions are assigned to "and" nodes.

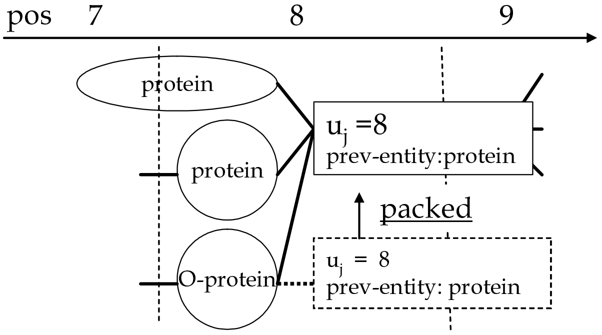

Figure 4: Example of packed representation of semi-CRFs. The states that have the same end position and prev-entity label are packed.

"and" nodes (*conjunctive* nodes), while boxes denote "or" nodes (*disjunctive* nodes). Feature functions are assigned to "and" nodes. We can use the information of the previous "and" node for designing the feature functions through the previous "or" node. Each sequence in a feature forest is obtained by choosing a conjunctive node for each disjunctive node. For example, Figure 3 represents $3 \times 3 = 9$ sequences, since each disjunctive node has three candidates. It should be noted that feature forests can represent an exponential number of sequences with a polynomial number of conjunctive/disjunctive nodes.

One can estimate a maximum entropy model for the whole sequence with dynamic programming by representing the probabilistic events, i.e. sequence of named entity tags, by feature forests (Miyao and Tsujii, 2002).

In the previous work (Lafferty et al., 2001; Sarawagi and Cohen, 2004), "or" nodes are considered implicitly in the dynamic programming framework. In feature forest models, "or" nodes are packed when they have same conditions. For example, "or" nodes are packed when they have same end positions and same labels in the first order semi-CRFs,

In general, we can pack different "or" nodes that yield equivalent feature functions in the following nodes. In other words, "or" nodes are packed when the following states use partial information on the preceding states. Consider the task of tagging *entity* and *O-entity*, where the latter tag is actually *O* tags that distinguish the preceding named entity tags. When we simply apply first-order semi-CRFs, we must distinguish states that have different previous states. However, when we want to distinguish only the preceding named entity tags rather than the immediate previous states, feature forests can represent these events more compactly (Figure 4). We can implement this as follows. In each "or" node, we generate the following "and" nodes and their feature functions. Then we check whether there exist "or" node which has same conditions by using its information about "end position" and "previous entity". If so, we connect the "and" node to the corresponding "or" node. If not, we generate a new "or" node and continue the process.

Since the states with label *O-entity* and *entity* are packed, the computational cost of training in our model (First order semi-CRFs) becomes the half of the original one.

5 Experiments

5.1 Experimental Setting

Our experiments were performed on the training and evaluation set provided by the shared task in COLING 2004 JNLPBA (Kim et al., 2004). The training data used in this shared task came from the GENIA version 3.02 corpus. In the task there are five semantic labels: protein, DNA, RNA, cell_line and cell_type. The training set consists of 2000 abstracts from MEDLINE, and the evaluation set consists of 404 abstracts. We divided the original training set into 1800 abstracts and 200 abstracts, and the former was used as the training data and the latter as the development data. For semi-CRFs, we used *amis*[3] for training the semi-CRF with feature-forest. We used *GENIA taggar*[4] for POS-tagging and shallow parsing.

We set $L = 10$ for training and evaluation when we do not state L explicitly , where L is the upper bound of the length of possible chunks in semi-CRFs.

5.2 Features

Table 3 lists the features used in our semi-CRFs. We describe the chunk-dependent features in detail, which cannot be encoded in token-level features.

"**Whole chunk**" is the normalized names attached to a chunk, which performs like the closed dictionary. "**Length**" and "**Length and End-Word**" capture the tendency of the length of a named entity. "**Count feature**" captures the tendency for named entities to appear repeatedly in the same sentence.

"**Preceding Entity and Prev Word**" are features that capture specifically words for conjunctions such as "*and*" or ", (comma)", e.g., for the phrase "*OCIM1 and K562*", both "*OCIM1*" and "*K562*" are assigned cell_line labels. Even if the model can determine only that "*OCIM1*" is a cell_line , this feature helps "*K562*" to be assigned the label cell_line.

5.3 Results

We first evaluated the filtering performance. Table 4 shows the result of the filtering on the training

[3] http://www-tsujii.is.s.u-tokyo.ac.jp/amis/

[4] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/ Note that the evaluation data are not used for training the GENIA tagger.

Table 3: Feature templates used for the chunk **s** := w_s w_{s+1} ... w_e where w_s and w_e represent the words at the beginning and ending of the target chunk respectively. p_i is the part of speech tag of w_i and sc_i is the shallow parse result of w_i.

Feature Name	description of features
Non-Chunk Features	
Word/POS/SC with Position	BEGIN + w_s, END + w_e, IN + w_{s+1}, ..., IN + w_{e-1}, BEGIN + p_s,...
Context Uni-gram/Bi-gram	w_{s-1}, w_{e+1}, w_{s-2} + w_{s-1}, w_{e+1} + w_{e+2}, w_{s-1} + w_{e+1}
Prefix/Suffix of Chunk	2/3-gram character prefix of w_s, 2/3/4-gram character suffix of w_e
Orthography	capitalization and word formation of $w_s ... w_e$
Chunk Features	
Whole chunk	w_s + w_{s+1} + ... + w_e
Word/POS/SC End Bi-grams	w_{e-1} + w_e, $p_{e-1} + p_e$, sc_{e-1} + sc_e
Length, Length and End Word	|**s**|, |**s**|+w_e
Count Feature	the frequency of $w_s w_{s+1} .. w_e$ in a sentence is greater than one
Preceding Entity Features	
Preceding Entity /and Prev Word	$PrevState$, $PrevState$ + w_{s-1}

Table 4: Filtering results using the naive Bayes classifier. The number of entity candidates for the training set was 4179662, and that of the development set was 418628.

Training set		
Threshold probability	reduction ratio	recall
1.0×10^{-12}	0.14	0.984
1.0×10^{-15}	0.20	0.993
Development set		
Threshold probability	reduction ratio	recall
1.0×10^{-12}	0.14	0.985
1.0×10^{-15}	0.20	0.994

and evaluation data. The naive Bayes classifiers effectively reduced the number of candidate states with very few falsely removed correct entities.

We then examined the effect of filtering on the final performance. In this experiment, we could not examine the performance without filtering using all the training data, because training on all the training data without filtering required much larger memory resources (estimated to be about 80G Byte) than was possible for our experimental setup. We thus compared the result of the recognizers with and without filtering using only 2000 sentences as the training data. Table 5 shows the result of the total system with different filtering thresholds. The result indicates that the filtering method achieved very well without decreasing the overall performance.

We next evaluate the effect of filtering, chunk information and non-local information on final performance. Table 6 shows the performance result for the recognition task. L means the upper bound of the length of possible chunks in semi-CRFs. We note that we cannot examine the result of $L = 10$ without filtering because of the intractable computational cost. The row “w/o Chunk Feature” shows the result of the system which does not employ Chunk-Features in Table 3 at training and inference. The row “Preceding Entity” shows the result of a system which uses **Preceding Entity** and **Preceding Entity and Prev Word** features. The results indicate that the chunk features contributed to the performance, and the filtering process enables us to use full chunk representation ($L = 10$). The results of McNemar's test suggest that the system with chunk features is significantly better than the system without it (the p-value is less than $1.0 < 10^{-4}$). The result of the preceding entity information improves the performance. On the other hand, the system with preceding information is not significantly better than the system without it[5]. Other non-local information may improve performance with our framework and this is a topic for future work.

Table 7 shows the result of the overall performance in our best setting, which uses the information about the preceding entity and 1.0×10^{-15} threshold probability for filtering. We note that the result of our system is similar to those of other sys-

[5] The result of the classifier on development data is 74.64 (without preceding information) and 75.14 (with preceding information).

Table 5: Performance with filtering on the development data. ($< 1.0 \times 10^{-12}$) means the threshold probability of the filtering is 1.0×10^{-12}.

	Recall	Precision	F-score	Memory Usage (MB)	Training Time (s)
Small Training Data = 2000 sentences					
Without filtering	65.77	72.80	69.10	4238	7463
Filtering ($< 1.0 \times 10.0^{-12}$)	64.22	70.62	67.27	600	1080
Filtering ($< 1.0 \times 10.0^{-15}$)	65.34	72.52	68.74	870	2154
All Training Data = 16713 sentences					
Without filtering	Not available			Not available	
Filtering ($< 1.0 \times 10.0^{-12}$)	70.05	76.06	72.93	10444	14661
Filtering ($< 1.0 \times 10.0^{-15}$)	**72.09**	**78.47**	**75.14**	15257	31636

Table 6: Overall performance on the evaluation set. L is the upper bound of the length of possible chunks in semi-CRFs.

	Recall	Precision	F-score
$L < 5$	64.33	65.51	64.92
$L = 10$ + Filtering ($< 1.0 \times 10.0^{-12}$)	70.87	68.33	69.58
$L = 10$ + Filtering ($< 1.0 \times 10.0^{-15}$)	72.59	70.16	71.36
w/o Chunk Feature	70.53	69.92	70.22
+ Preceding Entity	**72.65**	**70.35**	**71.48**

tems in several respects, that is, the performance of cell_line is not good, and the performance of the right boundary identification (78.91% in F-score) is better than that of the left boundary identification (75.19% in F-score).

Table 8 shows a comparison between our system and other state-of-the-art systems. Our system has achieved a comparable performance to these systems and would be still improved by using external resources or conducting pre/post processing. For example, Zhou et. al (2004) used post processing, abbreviation resolution and external dictionary, and reported that they improved F-score by 3.1%, 2.1% and 1.2% respectively. Kim et. al (2005) used the original GENIA corpus to employ the information about other semantic classes for identifying term boundaries. Finkel et. al (2004) used gazetteers, web-querying, surrounding abstracts, and frequency counts from the BNC corpus. Settles (2004) used semantic domain knowledge of 17 types of lexicon. Since our approach and the use of external resources/knowledge do not conflict but are complementary, examining the combination of those techniques should be an interesting research topic.

Table 7: Performance of our system on the evaluation set

Class	Recall	Precision	F-score
protein	77.74	68.92	73.07
DNA	69.03	70.16	69.59
RNA	69.49	67.21	68.33
cell_type	65.33	82.19	72.80
cell_line	57.60	53.14	55.28
overall	72.65	70.35	71.48

Table 8: Comparison with other systems

System	Recall	Precision	F-score
Zhou et. al (2004)	75.99	69.42	72.55
Our system	72.65	70.35	71.48
Kim et.al (2005)	72.77	69.68	71.19
Finkel et. al (2004)	68.56	71.62	70.06
Settles (2004)	70.3	69.3	69.8

6 Conclusion

In this paper, we have proposed a single probabilistic model that can capture important characteristics of biomedical named entities. To overcome the prohibitive computational cost, we have presented an efficient training framework and a filtering method which enabled us to apply first order semi-CRF models to sentences having many labels and entities with long names. Our results showed that our filtering method works very well without decreasing the overall performance. Our system achieved an F-score of 71.48% without the use of gazetteers, post-processing or external resources. The performance of our system came close to that of the current best performing system which makes extensive use of external resources and rule based post-processing.

The contribution of the non-local information introduced by our method was not significant in the experiments. However, other types of non-local information have also been shown to be effective (Finkel et al., 2005) and we will examine the effectiveness of other non-local information which can be embedded into label information.

As the next stage of our research, we hope to apply our method to shallow parsing, in which segments tend to be long and non-local information is important.

References

Daniel M. Bikel, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Proc. of the Fifth Conference on Applied Natural Language Processing*.

Jenny Finkel, Shipra Dingare, Huy Nguyen, Malvina Nissim, Gail Sinclair, and Christopher Manning. 2004. Exploiting context for biomedical entity recognition: From syntax to the web. In *Proc. of JNLPBA-04*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. of ACL 2005*, pages 363–370.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proc. of JNLPBA-04*, pages 70–75.

Seonho Kim, Juntae Yoon, Kyung-Mi Park, and Hae-Chang Rim. 2005. Two-phase biomedical named entity recognition using a hybrid method. In *Proc. of the Second International Joint Conference on Natural Language Processing (IJCNLP-05)*.

Zhenzhen Kou, William W. Cohen, and Robert F. Murphy. 2005. High-recall protein entity recognition using a dictionary. *Bioinformatics 2005 21*.

Micahel Krauthammer and Goran Nenadic. 2004. Term identification in the biomedical literature. *Jornal of Biomedical Informatics*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML 2001*.

Yusuke Miyao and Jun'ichi Tsujii. 2002. Maximum entropy estimation for feature forests. In *Proc. of HLT 2002*.

Peshkin and Pfeffer. 2003. Bayesian information extraction network. In *IJCAI*.

Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *NIPS 2004*.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proc. of JNLPBA-04*.

Beth M. Sundheim. 1995. Overview of results of the MUC-6 evaluation. In *Sixth Message Understanding Conference (MUC-6)*, pages 13–32.

Charles Sutton and Andrew McCallum. 2004. Collective segmentation and labeling of distant entities in information extraction. In *ICML workshop on Statistical Relational Learning*.

Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2005. Chunk parsing revisited. In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT 2005)*.

GuoDong Zhou and Jian Su. 2004. Exploring deep knowledge resources in biomedical name recognition. In *Proc. of JNLPBA-04*.