An Online Cascaded Approach to Biomedical Named Entity Recognition [*]

Shing-Kit Chan, Wai Lam, Xiaofeng Yu
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
Shatin, Hong Kong
{skchan, wlam, xfyu}@se.cuhk.edu.hk

Abstract

We present an online cascaded approach to biomedical named entity recognition. This approach uses an online training method to substantially reduce the training time required and a cascaded framework to relax the memory requirement. We conduct detailed experiments on the BioNLP dataset from the JNLPBA shared task and compare the results with other systems and published works. Our experimental results show that our approach achieves comparable performance with great reductions in time and space requirements.

1 Introduction

In the biomedical domain, the vast amount of data and the great variety of induced features are two major bottlenecks for further natural language processing on the biomedical literature. In this paper, we investigate the biomedical named entity recognition (NER) problem. This problem is particularly important because it is a necessary pre-processing step in many applications.

This paper addresses two main issues that arise from biomedical NER.

Long Training Time: Traditional approaches that depend on the maximum likelihood training method are slow even with large-scale optimization methods such as L-BFGS. This problem worsens with the sheer volume and growth rate of the biomedical literature. In this paper, we propose the use of an online training method that greatly reduces training time.

Large Memory Space: The total number of features used to extract named entities from documents is very large. To extract biomedical named entities, we often need to use extra features in addition to those used in general-purpose domains, such as prefix, suffix, punctuation, and more orthographic features. We need a correspondingly large memory space for processing, exacerbating the first issue. We propose to alleviate this problem by employing a cascaded approach that divides the NER task into a segmentation task and a classification task.

The overall approach is the online cascaded approach, which is described in the remaining sections of this paper: Section 2 describes the general model that is used to address the above issues. We address the issue of long training time in Section 3. The issue of large memory space is addressed in Section 4. Experimental results and analysis are presented in Section 5. We discuss related work in Section 6 and conclude with Section 7.

2 Model Descriptions

Our proposed model is similar to a conditional random field in a sequence labeling task, but we avoid directly dealing with the probability distribution. We use a joint feature representation $\mathbf{F}(\mathbf{x}, \mathbf{y})$ for each

[*] The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Nos: CUHK 4179/03E and CUHK4193/04E) and the Direct Grant of the Faculty of Engineering, CUHK (Project Codes: 2050363 and 2050391). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

input sequence $\mathbf{x}$ and an arbitrary output sequence $\mathbf{y}$, as follows.

$$\mathbf{F}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{|\mathbf{x}|} \mathbf{f}(\mathbf{x}, \mathbf{y}, i) \tag{1}$$

where each $\mathbf{f}(\mathbf{x}, \mathbf{y}, i)$ is a *local feature function* at position i. For example, in a segmentation task using the IOB2 notation, the k-th *local feature* in $\mathbf{f}(\mathbf{x}, \mathbf{y}, i)$ can be defined as

$$f_k(\mathbf{x}, \mathbf{y}, i) = \begin{cases} 1 & \text{if } x_i \text{ is the word "boy",} \\ & \text{and } y_i \text{ is the label "B"} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

With parameter $\mathbf{w}$, the best output sequence $\hat{\mathbf{y}}$ for an input sequence $\mathbf{x}$ can be found by calculating the best score:

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}'} \mathbf{w} \cdot \mathbf{F}(\mathbf{x}, \mathbf{y}') \tag{3}$$

3 Online Training

We propose to estimate the parameter $\mathbf{w}$ in an online manner. In particular, we use the online passive-aggressive algorithm (Crammer et al., 2006). Parameters are estimated by margin-based training, which chooses the set of parameters that attempts to make the "margin" on each training instance $(\mathbf{x_t}, \mathbf{y_t})$ greater than a predefined value γ,

$$\mathbf{w} \cdot \mathbf{F}(\mathbf{x_t}, \mathbf{y_t}) - \mathbf{w} \cdot \mathbf{F}(\mathbf{x_t}, \mathbf{y}') \geq \gamma \quad \forall \mathbf{y}' \neq \mathbf{y_t} \tag{4}$$

A *hinge loss* function $\ell(\mathbf{w}; \mathbf{x_t})$ is defined as

$$\ell(\mathbf{w}; \mathbf{x_t}) = \begin{cases} 0 & \text{if } \gamma_t \geq \gamma \\ \gamma - \gamma_t & \text{otherwise} \end{cases} \tag{5}$$

where γ_t is the margin on input $\mathbf{x_t}$ defined as

$$\gamma_t = \mathbf{w} \cdot \mathbf{F}(\mathbf{x_t}, \mathbf{y_t}) - \max_{\mathbf{y}' \neq \mathbf{y_t}} \mathbf{w} \cdot \mathbf{F}(\mathbf{x_t}, \mathbf{y}') \tag{6}$$

In online training, the parameter $\mathbf{w}$ is updated iteratively. Formally speaking, in the t-th iteration with the parameter $\mathbf{w_t}$ and the training instance $\mathbf{x_t}$, we try to solve the following optimization problem.

$$\mathbf{w_{t+1}} = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{2} \|\mathbf{w} - \mathbf{w_t}\|^{\mathbf{2}} + C\xi \tag{7}$$

$$\text{such that } \ell(\mathbf{w}; (\mathbf{x_t}, \mathbf{y_t})) \leq \xi$$

where $C > 0$ is a user-defined *aggressiveness parameter* and $\xi \geq 0$ is a slack term for the training data when it is not *linearly-separable*. C controls the penalty of the slack term and the *aggressiveness* of each update step. A larger C implies a more aggressive update and hence a higher tendency to overfit. The solution to Problem (7) is

$$\mathbf{w_{t+1}} = \mathbf{w_t} - \tau_t[\mathbf{F}(\mathbf{x_t}, \mathbf{y_t}) - \mathbf{F}(\mathbf{x_t}, \hat{\mathbf{y}}_\mathbf{t})] \tag{8}$$

$$\text{where} \quad \tau_t = \min \left\{ C, \frac{\ell(\mathbf{w_t}; (\mathbf{x_t}, \mathbf{y_t}))}{\|\mathbf{F}(\mathbf{x_t}, \mathbf{y_t}) - \mathbf{F}(\mathbf{x_t}, \hat{\mathbf{y}}_\mathbf{t})\|^2} \right\} \tag{9}$$

The *passiveness* of this algorithm comes from the fact that the parameter $\mathbf{w_t}$ is not updated when the hinge loss for $\mathbf{x_t}$ is zero. It can be proved that the relative loss bound on the training data (and which also bounds the number of prediction mistakes on the training data) cannot be much worse than the best fixed parameter chosen in hindsight. See (Crammer et al., 2006) for a detailed proof.

Following most of the work on margin-based training, in this paper we choose γ to be a function of the correct output sequence $\mathbf{y}$ and the predicted output sequence $\hat{\mathbf{y}}$.

$$\gamma(\mathbf{y}, \hat{\mathbf{y}}) = \begin{cases} 0 & \text{if } \mathbf{y} = \hat{\mathbf{y}} \\ \sum_{i=1}^{|\mathbf{y}|} [[y_i \neq \hat{y_i}]] & \text{otherwise} \end{cases} \tag{10}$$

where $[[z]]$ is 1 if z is true, and 0 otherwise.

The major computation difficulty in this online training comes from Equation (3). Finding the best output $\hat{\mathbf{y}}$ is in general an intractable task. We follow the usual first-order independence assumption made in a linear-chained CRF (Lafferty et al., 2001) model and calculate the best score using the Viterbi algorithm.

4 Cascaded Framework

We divide the NER task into a segmentation task and a classification task. In the segmentation task, a sentence $\mathbf{x}$ is segmented, and possible segments of biomedical named entities are identified. In the classification task, the identified segments are classified into one of the possible named entity types or rejected.

In other words, in the segmentation task, the sentence $\mathbf{x}$ are segmented by

$$\hat{\mathbf{y}}_{\mathbf{s}} = \underset{\mathbf{y}'}{\operatorname{argmax}}\, \mathbf{w}_{\mathbf{s}} \cdot \mathbf{F}_{\mathbf{s}}(\mathbf{x}, \mathbf{y}') \tag{11}$$

where $\mathbf{F}_{\mathbf{s}}(\cdot)$ is the set of segment features, and $\mathbf{w}_{\mathbf{s}}$ is the parameter for segmentation.

In the classification task, the segments (which can be identified by $\mathbf{y}_{\mathbf{s}}$) in a sentence $\mathbf{x}$ are classified by

$$\hat{\mathbf{y}}_{\mathbf{c}} = \underset{\mathbf{y}'}{\operatorname{argmax}}\, \mathbf{w}_{\mathbf{c}} \cdot \mathbf{F}_{\mathbf{c}}(\mathbf{x}, \mathbf{y}_{\mathbf{s}}, \mathbf{y}') \tag{12}$$

where $\mathbf{F}_{\mathbf{c}}(\cdot)$ is the set of classification features, and $\mathbf{w}_{\mathbf{c}}$ is the parameter for classification.

In this cascaded framework, the number of possible labels in the segmentation task is N_s. For example, $N_s = 3$ in the IOB2 notation. In the classification task, the number of possible labels is $N_c + 1$, which is the number of entity types and one label for "Other". Following the first-order independence assumption, the maximum total number of features in the two tasks is $O(\max(N_s^2, N_c^2))$, which is much smaller than the single-phase approach in which the total number of features is $O((N_s N_c)^2)$.

Another potential advantage of dividing the NER task into two tasks is that it allows greater flexibility in choosing an appropriate set of features for each task. In fact, adding more features may not necessarily increase performance. (Settles, 2004) reported that a system using a subset of features outperformed one using a full set of features.

5 Experiments

We conducted our experiments on the GENIA corpus (Kim et al., 2003) provided in the JNLPBA (Kim et al., 2004) shared task[1]. There are 2,000 MEDLINE abstracts in the GENIA corpus with named entities tagged in the IOB2 format. There are 18,546 sentences and 492,551 words in the training set, and 3,856 sentences and 101,039 words in the evaluation set. The line indicating the MEDLINE abstract ID boundary information is not used in our experiments. Each word is tagged with "B-X", "I-X", or "O" to indicate that the word is at the "beginning" (B) or "inside" (I) of a named entity of type X, or "outside" (O) of a named entity. The named entity types are: DNA, RNA, cell_line, cell_type, and protein.

System	F_1
(Zhou and Su, 2004)	72.55
Online Cascaded	**72.16**
(Okanohara et al., 2006)	71.48
(Kim et al., 2005)	71.19
(Finkel et al., 2004)	70.06
(Settles, 2004)	69.80

Table 1: Comparisons with other systems on overall performance (in percentage).

5.1 Features

The features used in our experiments mainly follow the work of (Settles, 2004) and (Collins, 2001). For completeness, we briefly describe the features here. They include word features, orthographic features, parts-of-speech (POS), and two lexicons. The word features include unigram, bigram, and trigram (e.g. the previous word, the next word, and the previous two words), whereas the orthographic features include capital letter, dash, punctuation, and word length. *Word class* (WC) features are also added, which replace a capital letter with "A", a lower case letter with "a", a digit with "0", and all other characters with "_". Similar *brief word class* (BWC) features are added by collapsing all of the consecutive identical characters in the *word class* features into one character. For example, for the word `NF-kappa`, WC = AA_aaaaa, and BWC = A_a. These are listed in Tables 2 and 3. The POS features are added by the GENIA tagger[2].

All of these features except for the prefix/suffix features are applied to the neighborhood window $[i-1, i+1]$ for every word. Two lexicons for cell lines and genes are drawn from two online public databases: the Cell Line Database[3] and the BBID[4]. The prefix/suffix and lexicon features are applied to position i only. All of the above features are com-

[1] http://research.nii.ac.jp/~collier/workshops/JNLPBA04st.htm

[2] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/

[3] http://www.biotech.ist.unige.it/cldb/cname-tz.html

[4] http://bbid.grc.nia.nih.gov/bbidgene.html

Unigram	(w_{-2}), (w_{-1}), (w_0), (w_1), (w_2)
Bigram	$(w_{-2}\ w_{-1})$, $(w_{-1}\ w_0)$, $(w_0\ w_1)$, $(w_1\ w_2)$
Trigram	$(w_{-2}\ w_{-1}\ w_0)$, $(w_{-1}\ w_0\ w_1)$, $(w_0\ w_1\ w_2)$

Table 2: Word features used in the experiment: w_0 is the current word, w_{-1} is the previous word, etc.

Word features	as in Table 2
Prefix/suffix	Up to a length of 5
Word Class	WC
Brief Word Class	BWC

Capital Letter	`^[A-Z][a-z]`
	`[A-Z]{2,}`
	`[a-z]+[A-Z]+`

Digit	`[0-9]+`
	`^[^0-9]*[0-9][^0-9]*$`
	`^[^0-9]*[0-9][0-9][^0-9]*$`
	`^[0-9]+$`
	`[0-9]+[,.][0-9,.]+`
	`[A-Za-z]+[0-9]+`
	`[0-9]+[A-Za-z]+`

Dash	`[-]+`
	`^[-]+`
	`[-]+$`

Punctuation	`[,;:?!-+’"\/]+`
Word length	length of the current word x_i

Table 3: Features used in the JNLPBA experiment. The features for *Capital Letter*, *Digit*, *Dash*, and *Punctuation* are represented as regular expressions.

bined with the previous label y_{i-1} and the current label y_i to form the final set of features.

In the segmentation task, only three labels (i.e. `B, I, O`) are needed to represent the segmentation results. In the classification task, the possible labels are the five entity types and "`Other`". We also add the segmentation results as features in the classification task.

5.2 Results

We tried different methods to extract the named entities from the JNLPBA dataset for comparisons. These programs were developed based on the same basic framework. All of the experiments were run on a Unix machine with a 2.8 GHz CPU and 16 GB RAM. In particular, the CRF trained by maximum-likelihood uses the L-BFGS algorithm (Liu and Nocedal, 1989), which converges quickly and gives a good performance on maximum entropy models (Malouf, 2002; Sha and Pereira, 2003). We compare our experimental results in several dimensions.

Training Time: Referring to Table 4, the training time of the online cascaded approach is substantially shorter than that of all of the other approaches. In the single-phase approach, training a CRF by maximum likelihood (ML) using the L-BFGS algorithm is the slowest and requires around 28 hours. The online method greatly reduces the training time to around two hours, which is 14 times faster. By employing a two-phase approach, the training time is further reduced to half an hour.

Memory Requirement: Table 4 shows the number of features that are required by the different methods. For methods that use the single-phase approach, because the full set of features (See Section 4) is too big for practical experiments on our machine, we need to set a higher cutoff value to reduce the number of features. With a cutoff of 20 (i.e. only features that occur more than 20 times are used), the number of features can still go up to about 8 million. However, in the two-phase approach, even with a smaller cutoff of 5, the number of features can still remain at about 8 million.

F_1**-measure:** Table 4 shows the F_1-measure in our experiments, and Table 1 compares our results with different systems in the JNLPBA shared tasks and other published works[5]. Our performance of the single-phase CRF with maximum likelihood training is 69.44%, which agrees with (Settles, 2004) who also uses similar settings. The single-phase online method increases the performance to 71.17%. By employing a cascaded framework, the performance is further increased to 72.16%, which can be regarded as comparable with the best system in the JNLPBA shared task.

6 Related Work

The online training approach used in this paper is based on the concept of "margin" (Cristianini, 2001). A pioneer work in online training is the perceptron-like algorithm used in training a hidden Markov model (HMM) (Collins, 2002). (McDonald

[5] We are aware of the high F_1 in (Vishwanathan et al., 2006). We contacted the author and found that their published result may be incomplete.

Experiments		no. of features	training time	F_1	rel. err. red. on F_1
single-phase	CRF + ML	8,004,392	1699 mins	69.44	–
	CRF + Online	8,004,392	116 mins	71.17	5.66%
two-phase	Online + Cascaded	seg: 2,356,590 class: 8,278,794	14 + 15 = 29 mins	72.16	8.90%

Table 4: The number of features, training time, and F_1 that are used in our experiments. The cutoff thresholds for the single-phase CRFs are set to 20, whereas that of the online cascaded approach is set to 5 in both segmentation and classification. The last column shows the relative error reductions on F_1 (compared to CRF+ML).

Experiments	R	P	F_1
Segmentation	80.13	73.68	76.77
Classification	92.75	92.76	92.76

Table 5: Results of the individual task in the online cascaded approach. The F_1 of the classification task is 92.76% (which is based on the fully correct segmented testing data).

et al., 2005) also proposed an online margin-based training method for parsing. This type of training method is fast and has the advantage that it does not need to form the dual problem as in SVMs. A detailed description of the online passive-aggressive algorithm used in this paper and its variants can be found in (Crammer et al., 2006). The Margin Infused Relaxed Algorithm (MIRA), which is the ancestor of the online passive-aggressive algorithm and mainly for the *linearly-separable* case, can be found in (Crammer and Singer, 2003).

(Kim et al., 2005) uses a similar two-phase approach but they need to use rule-based post-processing to correct the final results. Their CRFs are trained on a different dataset that contains all of the other named entities such as *lipid*, *multi cell*, and *other organic compound*. Table 1 shows the comparisons of the final results.

In the JNLPBA shared task, eight NER systems were used to extract five types of biomedical named entities. The best system (Zhou and Su, 2004) uses “deep knowledge”, such as name alias resolution, cascaded entity name resolution, abbreviation resolution, and in-domain POS. Our approach is relatively simpler and uses a unified model to accomplish the cascaded tasks. It also allows other post-processing tasks to enhance performance.

7 Conclusion

We have presented an online cascaded approach to biomedical named entity recognition. This approach substantially reduces the training time required and relaxes the memory requirement. The experimental results show that our approach achieves performance comparable to the state-of-the-art system.

References

Michael Collins. 2001. Ranking algorithms for named-entity extraction: boosting and the voted perceptron. In *ACL ’02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 489–496.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *EMNLP ’02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 1–8.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Nello Cristianini. 2001. Support vector and kernel machines. ICML tutorial. Available at http://www.support-vector.net/icml-tutorial.pdf.

J. Finkel, S. Dingare, H. Nguyen, M. Nissim, C. Manning, and G. Sinclair. 2004. Exploiting context for biomedical entity recognition: from syntax to the web. In *Proceedings of the International Joint Workshop on*

Natural Language Processing in Biomedicine and its Applications (NLPBA), pages 88–91.

J.d. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. Genia corpus - a semantically annotated corpus for biotextmining. *Bioinformatics (Supplement: Eleventh International Conference on Intelligent Systems for Molecular Biology)*, 19:180–182.

J. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, and N. Collier. 2004. Introduction to the bio-entity recognition task at JNLPBA. In N. Collier, P. Ruch, and A. Nazarenko, editors, *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA), Geneva, Switzerland*, pages 70–75, August 28–29. held in conjunction with COLING'2004.

Seonho Kim, Juntae Yoon, Kyung-Mi Park, and Hae-Chang Rim. 2005. Two-phase biomedical named entity recognition using a hybrid method. In *Proceedings of The Second International Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 646–657.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289.

D. C. Liu and J. Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(3):503–528.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of CoNLL-2002*, pages 49–55.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 91–98.

Daisuke Okanohara, Yusuke Miyao, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2006. Improving the scalability of semi-markov conditional random fields for named entity recognition. In *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 465–472.

B. Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)*, pages 104–107.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 134–141.

S. V. N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin P. Murphy. 2006. Accelerated training of conditional random fields with stochastic gradient methods. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 969–976.

GuoDong Zhou and Jian Su. 2004. Exploring deep knowledge resources in biomedical name recognition. In *COLING 2004 International Joint workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP) 2004*, pages 99–102.