Biomedical Term Recognition Using Discriminative Training

Sittichai Jiampojamarn, Grzegorz Kondrak and Colin Cherry
Department of Computing Science,
University of Alberta, Edmonton
Alberta, Canada T6G 2E8
{*sj, kondrak, colinc*}@*cs.ualberta.ca*

Abstract

We investigate the Perceptron HMM algorithm, an instance of the *averaged perceptron* approach, which incorporates discriminative training into the traditional Hidden Markov Model (HMM) approach. We demonstrate the efficiency of the algorithm by applying it to the biomedical term recognition problem. We show that the Perceptron HMM overcomes the limited expressiveness of the traditional, generative HMMs by incorporating additional, potentially overlapping features. This simple and elegant learning method produces performance that is comparable to the current state-of-the-art, while using only straightforward features derived from the provided training data. Our experiments illustrate the relative value of competing techniques that employ more complex learning algorithms and semantic features constructed from external resources.

Keywords

discriminative training, averaged perceptron, HMMs, biomedical term recognition, gene tagging, named entity extraction

1 Introduction

In recent years, discriminative training has become increasingly popular in natural language processing. Discriminative approaches allow us to incorporate a large number of features without concern for their independence. This gives these learners a significant advantage over more traditional generative techniques. However, some discriminative techniques, such as Conditional Random Fields (CRFs), are complex, difficult to implement, and expensive to train. Is it possible to combine the flexibility of feature independence with the elegance and conceptual simplicity of generative techniques?

In this paper, we investigate the Perceptron HMM algorithm, an instance of the *averaged perceptron* approach proposed by Collins [1]. The perceptron makes it possible to incorporate discriminative training into the traditional Hidden Markov Model (HMM) approach, and to augment it with potentially overlapping features. The Perceptron HMM uses the Viterbi algorithm with a simple perceptron update to train its feature weights. The Viterbi algorithm finds the best answer based on the current parameters while the perceptron algorithm updates the parameters when errors are made. The updating and decoding processes are iterated over the training data until the system converges.

We demonstrate the efficiency of the Perceptron HMM algorithm by applying it, along with a traditional HMM approach, to a specific problem — biomedical term recognition. We show that Perceptron HMM overcomes the limited expressiveness of the traditional HMM by incorporating additional interdependent features, such as part-of-speech, orthographic patterns, and affixes. Using a relatively small number of features that can be derived directly from the training data, we achieve results that are comparable to the current state-of-the-art systems that utilize external features derived from the Web or semantic knowledge-bases.

In the next section, we define the biomedical term identification task. The related work is discussed in Section 3. In Section 4, we describe a basic HMM approach. In Section 5, we introduce our proposed system based on the Perceptron HMM algorithm. In Section 6, we discuss our feature set. Experimental results and conclusions are given in Sections 7 and 8, respectively.

2 Biomedical term recognition

Every day, new scientific articles in the biomedical field are published and made available on-line. The articles contain many new terms and names involving proteins, DNA, RNA, and a wide variety of other substances. Given the large volume of new research articles, it is important to develop systems capable of extracting meaningful relationships between substances from these articles. Such systems need to recognize and identify biomedical terms in unstructured texts. Biomedical term recognition is thus a step toward information extraction from biomedical texts.

```
High-dose growth hormone does not
affect <protein>proinflammatory
cytokine</protein> (<protein>tumor
necrosis factor-alpha</protein>,
<protein>interleukin-6</protein>, and
<protein>interferon-gamma</protein>)
release from activated
<cell_type>peripheral blood mononuclear
cells</cell_type> or after minimal to
moderate surgical stress.
```

Fig. 1: *An annotated example of a biomedical research article*

The term recognition task attempts to locate biomedical terminology in unstructured texts. The texts are unannotated biomedical research publications written in English. Meaningful terms, including proteins, DNA, RNA, cell types and cell line names, are identified in order to facilitate further text mining tasks. The ability to identify important terms that represent biomedical concepts in the text is crucial to understanding research publications.

The biomedical term recognition task can only be adequately addressed with machine-learning methods. A straightforward dictionary look-up method is bound to fail because of the term variations in the text, especially when the task focuses on locating exact term boundaries [8]. Rule-based systems can achieve good performance on small data sets, but the rules must be defined manually by domain experts, and are difficult to adapt to other data sets [4, 3]. On the other hand, systems based on machine-learning employ statistical techniques, and can be easily re-trained on different data.

Biomedical term recognition involves the identification of biomedical terms in documents. The input documents are assumed to be written in English without any additional annotation. The identified terms may comprise several words. We also classify the identified terms into biomedical concepts: proteins, DNA, RNA, cell types, and cell lines. An example of an annotated biomedical research publication from the Genia corpus[1] is shown in Fig. 1, where each identified term is annotated by a pair of XML tags.

Another annotation method, referred to as IOB, is more appropriate for learning. It utilizes three types of tags: `<B>` for the beginning word of a term, `<I>` for the remaining words of a term, and `<O>` for non-term words. For the purpose of term classification, the IOB tags are augmented with the names of the biomedical classes; for example, `<B-protein>` indicates the first word of a protein term. The total number of IOB tags is thus $2n + 1$, where n is the number of classes.

Our biomedical term recognition task is defined as follows: for every document in a set, find and mark each occurrence of a biomedical term. A term is considered to be annotated correctly only if all its composite words are annotated correctly. Precision, recall and F-measure are determined by comparing the identified terms against the terms annotated in the gold standard.

3 Related work

Apart from early rule-based systems [4, 3], most biomedical term recognition systems employ machine-learning techniques, which have the advantages of scalability and generalization. We can divide machine-learning techniques used for this task into two main approaches: word-based methods, and sequence-based methods.

The word-based methods annotate each word without taking previously assigned tags into account. The ABTA system [5] approaches term annotation as a classification problem on a sliding window of words across sentences. Park *et al.* [11] and Lee *et al.* [9] proposed systems based on Support Vector Machines (SVMs), which classify each word in text as an IOB tag. These systems performed poorly in the Bio-Entity recognition task JNLPBA [7]. However, the SVM approach appears to lead to substantial improvements if used in combination with HMMs [18] or if incorporated in a sequence-based method [10].

The sequence-based methods take other annotation decisions into account in order to decide on the tag for the current word. Zhou and Su [18] employed a combination of the HMM and SVM approaches with rich features, obtaining the best performance at the JNLPBA. The features were word formation patterns, morphological patterns, part-of-speech tag information and dictionaries constructed from Swiss-Prot, LocusLink and annotated terms in the training data. Finkel *et al.* [2] used a large list of words, containing over a million names, to train a model based on the Maximum Entropy Markov Model (MEMM) technique. Words in gazetteers along with biomedical concept class indicators were submitted to the Google API in order to determine biomedical concept classes with the highest number of hits. Conditional Random Fields (CRFs) were used by Settles [13] with orthographic features playing the main role, and biomedical concept classes representing semantic features.

By combining the results submitted by the eight participants in the Bio-Entity recognition task at JNLPBA, Si *et al.* [14] were able to achieve a 0.92 F-measure. Since the submitted results involve only the test data, a portion of them were used to train a CRF model that learned relative weights to be assigned to each system.

In the BioCreAtIvE Task 1A [16], MEMM, CRF

[1] The Genia corpus 3.02 is available at: `http://www-tsujii.is.s.u-tokyo.ac.jp/~genia`

and SVM systems achieved best results. In general, these systems incorporate both internal and external features. The internal features are the ones that can be extracted directly from the training data, and include sets of words, part-of-speech information, orthographic patters, and sub-string affixes. The external features utilize larger resources such as the world-wide-web, gazetteers, and biomedical dictionaries.

4 The basic HMM system

We begin by presenting a traditional first-order HMM, which finds the best sequence of IOB tags $t_1 t_2 \ldots t_n$ for a sequence of words $w_1 w_2 \ldots w_n$. The HMM involves a number of trained parameters. The initial probability π_{t_i} is the probability of the tag t_i being the starting tag in the tag sequence. The transition probability $a_{t_i,t_j} = P(t_j|t_i)$ is the probability of the current tag t_j given the previous tag t_i. The emission probability $b_{t_j,w_j} = P(w_j|t_j)$ is the probability of the word w_j given the tag t_j. The add-one smoothing technique is applied to prevent the occurrence of zero probability values.

The initial, transition and emission probabilities are calculated using maximum likelihood statistics from the training data. These probabilities are then used to find the most likely tag sequences in the test data. The probability value of a candidate tag sequence $t_{1..n}$ given a sequence of words $w_{1..n}$ is the product of the partial probabilities as shown in Equation 1.

$$P(t_{1..n}|w_{1..n}) = \pi_{t_1} b_{t_1 w_1} \prod_{i=1}^{n-1} a_{t_i,t_{i+1}} b_{t_{i+1},w_{i+1}} \quad (1)$$

Given a sequence words $w_1 w_2 \ldots w_n$ and the model probabilities, the mostly likely tag sequence can be found by using the Viterbi algorithm [6].

5 The Perceptron HMM algorithm

The Perceptron HMM algorithm combines the Viterbi and perceptron algorithms to replace a traditional HMM's conditional probabilities with discriminatively trained parameters. Adapting an HMM for perceptron learning and arbitrary features requires a substantial shift in notation. First of all, given a complete tag sequence t for a word sequence w, we define $\Psi(w,t)$ to be a vector of features describing t and its interactions with w. Our learned parameters are also represented by a vector α, which assigns a weight to each component feature of $\Psi(w,t)$. The weight of each feature can be either positive, to indicate evidence that t is the correct tag sequence for w, negative to indicate evidence against t, or zero to indicate no evidence.

Given a useful weight vector α, we also need a way to find the tag sequence t with the most evidence. That is, we need to search for:

$$\hat{t} = \arg\max_{t \in T} \left[\alpha \cdot \Psi(w,t)\right] \quad (2)$$

where T is the set of all possible tag sequences. If we formulate our features carefully, the Viterbi algorithm will provide the necessary $\arg\max$ operator. We will define our $\Psi(w,t)$ so that it never needs more information than what is available during a first-order Viterbi search:

$$\Psi(w,t) = \sum_{i=1}^{n-1} \psi(w, t_i, t_{i+1}) \quad (3)$$

where ψ is a feature vector that describes the subset of Ψ's features that are relevant to the interactions between an adjacent tag pair and a word sequence.

Now that we have a feature representation for a tag sequence, and a method to find the tag sequence with the most evidence according to α, our goal in learning α is clear. We want to find an α that separates the correct tag sequence from all other possible tag sequences. For every sentence-tag sequence pair (w,t) in our training set, we require:

$$\forall \bar{t} \in T \setminus t: \quad \alpha \cdot \Psi(w,t) > \alpha \cdot \Psi(w,\bar{t}) \quad (4)$$

It has been shown in [1] that a perceptron algorithm will find a separating α if it exists. In the case of unseparable data, an averaged perceptron will provide a useful approximation to this separator.

The training algorithm for the Perceptron HMM is sketched in Algorithm 1. In each iteration, for each training example, the perceptron adjusts its weight parameters α according to the features of its current best guess. The Viterbi algorithm finds the best sequence of tags $\hat{t}$ for w, given the current α. If this $\hat{t}$ is not the correct tag sentence, then α is altered slightly to prefer $\Psi(w,t)$ over $\Psi(w,\hat{t})$.

Algorithm 1 The perceptron training algorithm

```
$\alpha = \vec{0}$
for $K$ iterations over training set do
    for all sentence-tag sequence pairs $(w,t)$ in the training set do
        $\hat{t} = \arg\max_{\bar{t} \in T} [\alpha \cdot \Psi(w,\bar{t})]$
        $\alpha = \alpha + \Psi(w,t) - \Psi(w,\hat{t})$
    end for
end for
return $\alpha$
```

For example, suppose that in our training data we have the following sentence w with its correct annotation t. The current best guess found by our Viterbi algorithm is $\hat{t}$:

w	IL-2	gene	expression	and
t	B-DNA	I-DNA	O	O
$\hat{t}$	B-DNA	I-protein	O	O

If our features consist only of indicators for word-tag pairs and tag bigrams, the weight vector α is altered as follows:

- Weights corresponding to the features *(gene, I-DNA)* and *(B-DNA, I-DNA)* are incremented by 1

- Weights corresponding to *(gene, I-protein)* and *(B-DNA, I-protein)* are decremented by 1.

Term annotation is a complex problem; we are unlikely to find an α that perfectly separates our training data, no matter how good our features are. In order to compensate for this, instead of returning the final α as shown in Algorithm 1, we return the average α over all updates. This averaged perceptron tends to be more effective on unseen data [1].

6 The extended feature set

Our feature set is composed entirely of standard, internal features that have been incorporated in many systems [7]. These features can be divided into three broad classes according to how they generalize the training data: by words, characters or part-of-speech. Word features allow the system to remember common annotations for words that occur frequently in the training data. More general character-based features, such as orthography, prefix and suffix features, help the system recognize unseen words by memorizing linguistic patterns. Part-of-speech features provide syntactic information at the sentence level, which allows the system to take advantage of the fact that most terms are noun phrases. An example sequence of words and tags in the training set is shown below. Its corresponding features are shown in Table 1.

word	...	of	E1A-immortalized	cells	...
tag	...	O	B-cell_line	I-cell_line	...
POS	...	IN	CD	NNS	...

The part-of-speech tag features are obtained by using the Lingua::EN::Tagger[2]. The orthography features encode the spelling characteristics of a word, such as uppercase letters (U), lowercase letters (L), digits (D), and symbols (S). For example, the orthography feature for the word “E1A-immortalized” has the following value: “U D U S L”. The prefix and suffix features are the k first and last characters of words. For $k = 3$, the prefix and suffix features for the word “E1A-immortalized” have the values “E1A” and “zed”, respectively.

Feature template	Example
Word features & Current tag	
Current word	E1A-immortalized & B-cell_line
Previous word	of & B-cell_line
Next word	cells & B-cell_line
Bigram word	of E1A-immortalized & B-cell_line E1A-immortalized cells & B-cell_line
Part-of-Speech tag features & Current tag	
Current POS	CD & B-cell_line
Previous POS	IN & B-cell_line
Next POS	NNS & B-cell_line
Bigram POS	IN CD & B-cell_line CD NNS & B-cell_line
Orthography features & Current tag	
Current ORTH	U D U S L & B-cell_line
Previous ORTH	L & B-cell_line
Next ORTH	L & B-cell_line
Bigram ORTH	L U D U S L & B-cell_line U D U S L L & B-cell_line
Prefix features & Current tag	
Current PRE	E1A & B-cell_line
Previous PRE	of & B-cell_line
Next PRE	cel & B-cell_line
Bigram PRE	of E1A & B-cell_line E1A cel & B-cell_line
Suffix features & Current tag	
Current SUF	zed & B-cell_line
Previous SUF	of & B-cell_line
Next SUF	lls & B-cell_line
Bigram SUF	of zed & B-cell_line zed lls & B-cell_line

Table 1: *The feature template and example used in the experiments*

7 Results and discussions

We evaluated our system on the JNLPBA Bio-Entity recognition task. The training set contains 2,000 Medline abstracts labeled with biomedical classes in the IOB style. Our development set was constructed by randomly selecting 10% of the sentences from the available training set. The number of iterations for training was determined by observing the point where the performance on the held-out set starts to level off. The test set is composed of new 404 Medline abstracts.

The performance of the basic HMM system on the test data is shown in Table 2. Overall, the F-measure performance on the testing data was about 10% lower than on the training data. The highest F-measure was obtained on the protein class. The basic HMM completely fails to identify cell line terms.

Table 3 shows the results of our Perceptron HMM system on all five classes. Notice the impressive improvement over the basic HMM system, which is particularly evident for the terms of type RNA, cell type, and cell line.

Table 4 presents a comparison of our results with the results of eight participants at the JNLPBA shared tasks, which are taken from the task report [7]. The table also includes the basic HMM described in Sec-

[2] Lingua-EN-Tagger-0.13 by Aaron Coburn is available at `http://search.cpan.org/~acoburn`

Class (# of terms)	Recall	Precision	F-measure
Protein (5,067)	59.33%	58.84%	59.08%
DNA (1,056)	50.76%	53.17%	51.94%
RNA (118)	21.19%	55.56%	30.67%
cell_type (1,921)	49.97%	48.41%	49.18%
cell_line (500)	0.00%	0.00%	0.00%
ALL(8,662)	52.26%	55.57%	53.86%

Table 2: *The performance of the basic HMM system on the testing set*

Class	Recall	Precision	F-measure
protein	76.73 %	66.04 %	70.99 %
DNA	63.54 %	65.53 %	64.52 %
RNA	66.10 %	64.46 %	65.27 %
cell_type	64.65 %	78.56 %	70.93 %
cell_line	53.20 %	51.65 %	52.41 %
ALL	70.94 %	67.32 %	69.08 %

Table 3: *The performance of the proposed system on the test set with respect to each biomedical concept class*

tion 4, and the baseline system provided for the competition, which is based on longest string matching against a list of terms from the training data. The "Ext." column in Table 4 indicates whether a system includes a use of external resources. The external resources include gazetteers from dictionaries and Gene Ontology, various Word Wide Web (WWW) resources, British National Corpus, MEDLINE corpus, Penn Treebank II corpus, and tags from other gene/protein name taggers.

In terms of F-measure,our system ranks fourth. The performance gap between our system and the best systems in Table 4 can be attributed to the use of external features. When compared against other systems that use only internal features, our system achieves the highest F-measure.

The listed systems stratify into several categories, which should help elucidate the importance of external data. The three systems at the bottom of the list (our basic HMM, [9], [11]) use either sequence-based or discriminative learning, but not both; only the discriminative methods use external data. This shows that the use of an expressive sequence-based method is important in achieving competitive results. Among the next four systems, we have three methods that combine discriminative and sequence learning ([12], [15], and our P-HMM), along with the only generative sequence method to use external data [17]. Finally, the sequence-based discriminative systems that incorporate external data dominate the top of the list. With our approach, we have shown nearly a 3-point improvement in achievable performance when no external information sources are employed, greatly narrowing the gap between data-poor and data-rich features.

System	Method	Ext.	F-measure
Zhou and Su [18]	SVM-HMM	Y	72.6 %
Finkel *et al.* [2]	MEMM	Y	70.1 %
Settles [13]	CRF	Y	69.8 %
Our system	**P-HMM**	**N**	**69.1 %**
Song *et al.* [15]	SVM-CRF	N	66.3 %
Zhao [17]	HMM	Y	64.8 %
Rössler [12]	SVM-HMM	N	64.0 %
Park *et al.* [11]	SVM	Y	63.0 %
Basic HMM	**HMM**	**N**	**53.9 %**
Lee *et al.* [9]	SVM	Y	49.1 %
Baseline	Matching	N	47.7 %

Table 4: *The performance comparison*

Features	Precision	Recall	F-measure
word	64.27	61.85	63.04
word+POS	66.71	60.53	63.47
word+ORTH	65.59	61.97	63.73
word+PRE	61.53	65.31	63.37
word+SUF	64.48	64.75	64.61

Table 5: *The complete match performance of each included feature on the test set*

The full system uses all features described in Section 5: word, part-of-speech tag (POS), orthography (ORTH), prefix (PRE), and suffix (SUF) features. In order to measure the impact of these feature types, we trained several systems using a single feature class along with the basic word features. As one can see, each type of feature contributes very little on its own, increasing F-measure by at most 1.5 points. But together, these features are literally worth more than the sum of their parts, increasing F-measure by 6 points from 63 to 69. These additional features are internal features which can be directly obtained from the training set.

In order to compare the performance between traditional HMM and Perceptron HMM learning objectives, we limited the feature set in the Perceptron HMM to only the current word feature (the first line in Table 1). Thus, both the HMM and the Perceptron HMM have the same feature set, but the Perceptron HMM trains those features discriminatively. While the traditional HMM system achieves a 53.9% F-measure, the Perceptron HMM system achieves an F-measure of 56.9%. This 3-point increase shows the value of discriminative training when all other variables are held constant; performance increases before we even begin to take advantage of the perceptron's smooth handling of overlapping features.

8 Conclusion and future work

We have proposed a new approach to the biomedical term recognition task using the Perceptron HMM algorithm. Our system achieves a 69.1% F-measure with a simple and elegant machine-learning method, and a relatively small number of features that can be derived directly from the training data. The performance we achieve with this approach is comparable to the current state-of-the-art.

CRFs, SVM-HMMs and Perceptron HMMs are all discriminative training methods that have similar feature representations and learning objectives. Among them, the Perceptron HMM is by far the most straightforward in its implementation. It is our hope that our experiments help illustrate the relative value of the slower CRF and SVM approaches. Along the same lines, we have demonstrated just how far one can advance without having to resort to features mined from the web or semantic knowledge-bases.

Finally, we have provided a detailed comparison of the Perceptron HMM with a traditional HMM with maximum-likelihood parameters. We have illustrated the value of discriminative training, and we have shown that overlapping features allow a giant leap forward in performance while using the same Viterbi algorithm.

Acknowledgments

We would like to thank Susan Bartlett and other members of the Natural Language Processing research group at University of Alberta for their helpful comments and suggestions. This research was supported by the Natural Sciences and Engineering Research Council of Canada.

References

[1] M. Collins. Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Langauge Processing (EMNLP)*, 2002.

[2] J. Finkel, S. Dingare, H. Nguyen, M. Nissim, G. Sinclair, and C. Manning. Exploiting context for biomedical entity recognition: From syntax to the web. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.

[3] K. Franzen, G. Eriksson, F. Olsson, L. Asker, P. Liden, and J. Coster. Protein names and how to find them. In *International Journal of Medical Informatics special issue on Natural Language Processing in Biomedical Applications*, pages 49–61, 2002.

[4] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: Identifying protein names from biological papers. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 707–718, 1998.

[5] S. Jiampojamarn, N. Cercone, and V. Keselj. Biological named entity recognition using N-grams and classification methods. In *Proceedings of the Conference Pacific Assiciation for Computatioanl Linguistics (PACLING'05)*, 2005.

[6] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing*. Prentice Hall, 2000.

[7] J. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, and N. Collier. Introduction to the bio-entity recognition task at JNLPBA. In *Processings of the Joint Workshop on Natural Langauge Processing in Biomedicine and its Applications (JNLPBA-2004)*, 2004.

[8] M. Krauthammer and G. Nenadic. Term identification in the biomedical literature. In *Journal of Biomedical Informatics (Special Issue on Named Entity Recognition in Biomedicine)*, volume 37(6), pages 512–526, 2004.

[9] C. Lee, W. Hou, and H. Chen. Annotating multiple types of biomedical entities: A single word classificication approach. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.

[10] T. Mitsumori, S. Fation, M. Murata, K. Doi, and H. Doi. Gene/protein name recognition based on support vector machine using dictionary as features. In *BMC Bioinformatics 2005, 6(Suppl 1):S8*, 2005.

[11] K. Park, S. Kim, D. Lee, and H. Rim. Boosting lexical knowledge for biomedical named entity recognition. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.

[12] M. Rössler. Adapting an NER-system for german to the biomedical domain. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.

[13] B. Settles. Biomedical named entity recognition using conditional random fields and novel feature sets. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.

[14] L. Si, T. Kanungo, and X. Huang. Boosting performance of bio-entity recognition by combining results from multiple systems. In *BIOKDD '05: Proceedings of the 5th international workshop on Bioinformatics*, pages 76–83, New York, NY, USA, 2005. ACM Press.

[15] Y. Song, E. Kim, G. G. Lee, and B. Yi. POSBIOTM-NER in the shared task of BioNLP/NLPBA 2004. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.

[16] A. Yeh, A. Morgan, M. Colosimo, and L. Hirschman. BioCreAtIvE Task 1A: gene mention finding evaluation. In *BMC Bioinformatics 2005, 6(Suppl 1):S2*, 2005.

[17] S. Zhao. Name entity recognition in biomedical text using a HMM model. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.

[18] G. Zhou and J. Su. Exploring deep knowledge resources in biomedical name recognition. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its applications (JNLPBA-2004)*, 2004.