Recognizing Names in Biomedical Texts using Hidden Markov Model and SVM plus Sigmoid

ZHOU GuoDong
Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
Email: zhougd@i2r.a-star.edu.sg

ABSTRACT

In this paper, we present a named entity recognition system in the biomedical domain, called PowerBioNE. In order to deal with the special phenomena in the biomedical domain, various evidential features are proposed and integrated through a Hidden Markov Model (HMM). In addition, a Support Vector Machine (SVM) plus sigmoid is proposed to resolve the data sparseness problem in our system. Finally, we present two post-processing modules to deal with the cascaded entity name and abbreviation phenomena. Evaluation shows that our system achieves the F-measure of 69.1 and 71.2 on the 23 classes of GENIA V1.1 and V3.0 respectively. In particular, our system achieves the F-measure of 77.8 on the "*protein*" class of GENIA V3.0. It shows that our system outperforms the best published system on GENIA V1.1 and V3.0.

1. INTRODUCTION

With an overwhelming amount of textual information in molecular biology and biomedicine, there is a need for effective and efficient literature mining and knowledge discovery that can help biologists to gather and make use of the knowledge encoded in text documents. In order to make organized and structured information available, automatically recognizing biomedical entity names becomes critical and is important for protein-protein interaction extraction, pathway construction, automatic database curation, etc.

Such a task, called named entity recognition, has been well developed in the Information Extraction literature (MUC-6; MUC-7). In MUC, the task of named entity recognition is to recognize the names of persons, locations, organizations, etc. in the newswire domain. In the biomedical domain, we care about entities like gene, protein, virus, etc. In recent years, many explorations have been done to port existing named entity recognition systems into the biomedical domain (Kazama et al 2002; Lee et al 2003; Shen et al 2003; Zhou et al 2004). However, few of them have achieved satisfactory performance due to the special characteristics in the biomedical domain, such as long and descriptive naming conventions, conjunctive and disjunctive structure, causal naming convention and rapidly emerging new biomedical names, abbreviation, and cascaded construction. On all accounts, we can say that the entity names in the biomedical domain are much more complex than those in the newswire domain.

In this paper, we present a named entity recognition system in the biomedical domain, called PowerBioNE. In order to deal with the special phenomena in the biomedical domain, various evidential features are proposed and integrated effectively and efficiently through a Hidden Markov Model (HMM). In addition, a Support Vector Machine (SVM) plus sigmoid is proposed to resolve the data sparseness problem in our system. Finally, we present two post-processing modules to deal with the cascaded entity name and abbreviation phenomena to further improve the performance.

All of our experiments are done on the GENIA corpus, which is the largest annotated corpus in the molecular biology domain available to public (Ohta et al. 2002). In our experiments, two versions are used: 1) Genia V1.1 which contains 670 MEDLINE abstracts of 123K words; 2) Genia V3.0 which is a superset of GENIA V1.1 and contains 2000 MEDLINE abstracts of 360K words. The annotation of biomedical entities is based on the GENIA ontology (Ohta et al. 2002), which includes 23 distinct classes: multi-cell, mono-cell, virus, body part, tissue, cell type, cell component, organism, cell line, other artificial source, protein, peptide, amino acid monomer, DNA, RNA, poly nucleotide, nucleotide, lipid, carbohydrate, other organic compound, inorganic, atom and other.

2. FEATURES

In order to deal with the special phenomena in the biomedical domain, various evidential features are explored.

- **Word Formation Pattern (F_{WFP}):** The purpose of this feature is to capture capitalization, digitalization and other word formation

information. This feature has been widely used in the biomedical domain (Kazama et al 2002; Shen et al 2003; Zhou et al 2004). In this paper, the same feature as in Shen et al 2003 is used.

- **Morphological Pattern (F_{MP})**: Morphological information, such as prefix and suffix, is considered as an important cue for terminology identification and has been widely applied in the biomedical domain (Kazama et al 2002; Lee et al 2003; Shen et al 2003; Zhou et al 2004). Same as Shen et al 2003, we use a statistical method to get the most useful prefixes/suffixes from the training data.
- **Part-of-Speech (F_{POS}):** Since many of the words in biomedical entity names are in lowercase, capitalization information in the biomedical domain is not as evidential as that in the newswire domain. Moreover, many biomedical entity names are descriptive and very long. Therefore, POS may provide useful evidence about the boundaries of biomedical entity names.
- **Head Noun Trigger (F_{HEAD}):** The head noun, which is the major noun of a noun phrase, often describes the function or the property of the noun phrase. In this paper, we automatically extract unigram and bigram head nouns from the training data, and rank them by frequency. For each entity class, we select 50% of top ranked head nouns as head noun triggers. Table 1 shows some of the examples.

Table 1: Examples of auto-generated head nouns

Class	Unigram	bigram
PROTEIN	interleukin interferon kinase	activator protein binding protein cell receptor
DNA	DNA cDNA chromosome	X chromosome binding motif promoter element

- **Name Alias Feature (F_{ALIAS}):** Besides the above widely used features, we also propose a novel name alias feature. The intuition behind this feature is the name alias phenomenon that relevant entities will be referred to in many ways throughout a given text and thus success of named entity recognition is conditional on success at determining when one noun phrase refers to the very same entity as another noun phrase.

During decoding, the entity names already recognized from the previous sentences of the document are stored in a list. When the system encounters an entity name candidate (e.g. a word with a special word formation pattern), a name alias algorithm (similar to Schwartz et al 2003) is invoked to first dynamically determine whether the entity name candidate might be alias for a previously recognized name in the recognized list. This is done by checking whether all the characters in the entity name candidate exist in a recognized entity name in the same order and whether the first character in the entity name candidate is same as the first character in the recognized name. For a relevant work, please see Jacquemin (2001). The name alias feature F_{ALIAS} is represented as *ENTITYnLm* (L indicates the locality of the name alias phenomenon). Here *ENTITY* indicates the class of the recognized entity name and *n* indicates the number of the words in the recognized entity name while *m* indicates the number of the words in the recognized entity name from which the name alias candidate is formed. For example, when the decoding process encounters the word "*TCF*", the word "*TCF*" is proposed as an entity name candidate and the name alias algorithm is invoked to check if the word "*TCF*" is an alias of a recognized named entity. If "*T cell Factor*" is a "*Protein"* name recognized earlier in the document, the word "*TCF*" is determined as an alias of "*T cell Factor*" with the name alias feature *Protein3L3* by taking the three initial letters of the three-word "*protein"* name "*T cell Factor*".

3. METHODS

3.1 Hidden Markov Model

Given above various features, the key problem is how to effectively and efficiently integrate them together and find the optimal resolution to biomedical named entity recognition. Here, we use the Hidden Markov Model (HMM) as described in Zhou et al 2002. A HMM is a model where a sequence of outputs is generated in addition to the Markov state sequence. It is a latent variable model in the sense that only the output sequence is observed while the state sequence remains "hidden".

Given an observation sequence $O_1^n = o_1 o_2 ... o_n$, the purpose of a HMM is to find the most likely state sequence $S_1^n = s_1 s_2 ... s_n$ that maximizes $P(S_1^n \mid O_1^n)$. Here, the observation $o_i =< f_i, w_i >$, where w_i is the word and $f_i =< F_{WFP}^i, F_{MP}^i, F_{POS}^i, F_{HEAD}^i, F_{ALIAS}^i >$ is the feature set of the word w_i, and the state s_i is structural and $s_i = BOUNDARY_i \_ ENTITY_i \_ FEATURE_i$, where $BOUNDARY_i$ denotes the position of the current word in the entity; $ENTITY_i$ indicates the class of the entity; and $FEATURE_i$ is the feature set used to model the ngram more precisely.

By rewriting $\log P(S_1^n \mid O_1^n)$, we have:

$$\log P(S_1^n \mid O_1^n) = \log P(S_1^n) + \log \frac{P(S_1^n, O_1^n)}{P(S_1^n) \cdot P(O_1^n)} \quad (1)$$

The second term in Equation (1) is the mutual information between S_1^n and O_1^n. In order to simplify the computation of this term, we assume mutual information independence:

$$MI(S_1^n, O_1^n) = \sum_{i=1}^{n} MI(s_i, O_1^n) \quad \text{or}$$

$$\log \frac{P(S_1^n, O_1^n)}{P(S_1^n) \cdot P(O_1^n)} = \sum_{i=1}^{n} \log \frac{P(s_i, O_1^n)}{P(s_i) \cdot P(O_1^n)} \quad (2)$$

That is, an individual tag is only dependent on the output sequence O_1^n and independent on other tags in the tag sequence S_1^n. This assumption is reasonable because the dependence among the tags in the tag sequence S_1^n has already been captured by the first term in Equation (1). Applying the assumption (2) to Equation (1), we have:

$$\log P(S_1^n \mid O_1^n) = \log P(S_1^n) - \sum_{i=1}^{n} \log P(s_i) + \sum_{i=1}^{n} \log P(s_i \mid O_1^n) \quad (3)$$

From Equation (3), we can see that:

- The first term can be computed by applying chain rules. In ngram modeling (Chen et al 1996), each tag is assumed to be dependent on the N-1 previous tags.
- The second term is the summation of log probabilities of all the individual tags.
- The third term corresponds to the "lexical" component (dictionary) of the tagger.

The idea behind the model is that it tries to assign each output an appropriate tag (state), which contains boundary and class information. For example, "*TCF 1 binds stronger than NF kB to TCEd DNA*". The tag assigned to token "*TCF"* should indicate that it is at the beginning of an entity name and it belongs to the "*Protein"* class; and the tag assigned to token "*binds"* should indicate that it does not belong to an entity name. Here, the Viterbi algorithm (Viterbi 1967) is implemented to find the most likely tag sequence.

The problem with the above HMM lies in the data sparseness problem raised by $P(s_i \mid O_1^n)$ in the third term of Equation (3). Ideally, we would have sufficient training data for every event whose conditional probability we wish to calculate. Unfortunately, there is rarely enough training data to compute accurate probabilities when decoding on new data. Generally, two smoothing approaches (Chen et al 1996) are applied to resolve this problem: linear interpolation and back-off. However, these two approaches only work well when the number of different information sources is limited. When a few features and/or a long context are considered, the number of different information sources is exponential. In this paper, a Support Vector Machine (SVM) plus sigmoid is proposed to resolve this problem in our system.

3.2 Support Vector Machine plus Sigmoid

Support Vector Machines (SVMs) are a popular machine learning approach first presented by Vapnik (1995). Based on the structural risk minimization of statistical learning theory, SVMs seek an optimal separating hyper-plane to divide the training examples into two classes and make decisions based on support vectors which are selected as the only effective examples in the training set. However, SVMs produce an un-calibrated value that is not probability. That is, the unthresholded output of an SVM can be represented as

$$f(x) = \sum_{i \in SV} a_i \cdot y_i \cdot k(x_i, x) + b \quad (4)$$

To map the SVM output into the probability, we train an additional sigmoid model(Platt 1999):

$$p(s_i \mid f_i) = \frac{1}{1 + \exp(Af_i + B)} \quad (5)$$

Basically, SVMs are binary classifiers. Therefore, we must extend SVMs to multi-class (e.g. K) classifiers. For efficiency, we apply the *one vs. others* strategy, which builds K classifiers so as to separate one class from all others, instead of the *pairwise* strategy, which builds K*(K-1)/2 classifiers considering all pairs of classes. Moreover, we only apply the simple linear kernel, although other kernels (e.g. polynomial kernel) and pairwise strategy can have better performance. Finally, for each state s_i, there is one sigmoid $p(s_i \mid f_i)$. Therefore, the sigmoid outputs are normalized to get a probability distribution using $p(s_i \mid O_1^n) = \frac{p(s_i \mid f_i)}{\sum_i p(s_i \mid f_i)}$.

3.3 Post-Processing

Two post-processing modules, namely cascaded entity name resolution and abbreviation resolution, are applied in our system to further improve the performance.

Cascaded Entity Name Resolution

It is found (Shen et al 2003) that 16.57% of entity names in GENIA V3.0 have cascaded constructions, e.g.

<RNA><DNA>CIITA</DNA> mRNA</RNA>.

Therefore, it is important to resolve such phenomenon.

Here, a pattern-based module is proposed to resolve the cascaded entity names while the above HMM is applied to recognize embedded entity

names and non-cascaded entity names. In the GENIA corpus, we find that there are six useful patterns of cascaded entity name constructions:

- <ENTITY> := <ENTITY> + head noun, e.g. <PROTEIN> binding motif→<DNA>
- <ENTITY> := <ENTITY> + <ENTITY>, e.g. <LIPID> <PROTEIN>→<PROTEIN>
- <ENTITY> := modifier + <ENTITY>, e.g. anti <Protein>→<Protein>
- <ENTITY> := <ENTITY> + word + <ENTITY>, e.g. <VIRUS> infected <MULTICELL>→<MULTICELL >
- <ENTITY> := modifier + <ENTITY> + head noun
- <ENTITY> := <ENTITY> + <ENTITY> + head noun

In our experiments, all the rules of above six patterns are extracted from the cascaded entity names in the training data to deal with the cascaded entity name phenomenon.

Abbreviation Resolution

While the name alias feature is useful to detect the inter-sentential name alias phenomenon, it is unable to identify the inner-sentential name alias phenomenon: the inner-sentential abbreviation. Such abbreviations widely occur in the biomedical domain.

In our system, we present an effective and efficient algorithm to recognize the inner-sentential abbreviations more accurately by mapping them to their full expanded forms. In the GENIA corpus, we observe that the expanded form and its abbreviation often occur together via parentheses. Generally, there are two patterns: "expanded form (abbreviation)" and "abbreviation (expanded form)".

Our algorithm is based on the fact that it is much harder to classify an abbreviation than its expanded form. Generally, the expanded form is more evidential than its abbreviation to determine its class. The algorithm works as follows: Given a sentence with parentheses, we use a similar algorithm as in Schwartz et al 2003 to determine whether it is an abbreviation with parentheses. This is done by starting from the end of both the abbreviation and the expanded form, moving from right to left and trying to find the shortest expanded form that matches the abbreviation. Any character in the expanded form can match a character in the abbreviation with one exception: the match of the character at the beginning of the abbreviation must match the first alphabetic character of the first word in the expanded form. If yes, we remove the abbreviation and the parentheses from the sentence. After the sentence is processed, we restore the abbreviation with parentheses to its original position in the sentence. Then, the abbreviation is classified as the same class of the expanded form, if the expanded form is recognized as an entity name. In the meanwhile, we also adjust the boundaries of the expanded form according to the abbreviation, if necessary. Finally, the expanded form and its abbreviation are stored in the recognized list of biomedical entity names from the document to help the resolution of forthcoming occurrences of the same abbreviation in the document.

4. EXPERIMENTS AND EVALUATION

We evaluate our PowerBioNE system on GENIA V1.1 and GENIA V3.0 using precision/recall/F-measure. For each evaluation, we select 20% of the corpus as the held-out test data and the remaining 80% as the training data. All the experimentations are done 5 times and the evaluations are averaged over the held-out test data. For cascaded entity name resolution, an average of 59 and 97 rules are extracted from the cascaded entity names in the training data of GENIA V1.1 and V3.0 respectively. For POS, all the POS taggers are trained on the training data with POS imported from the corresponding GENIA V3.02p with POS annotated.

Table 2 shows the performance of our system on GENIA V1.1 and GENIA V3.0, and the comparison with that of the best reported system (Shen et al 2003). It shows that our system achieves the F-measure of 69.1 on GENIA V1.1 and the F-measure of 71.2 on GENIA V3.0 respectively, without help of any dictionaries. It also shows that our system outperforms Shen et al (2003) by 6.9 in F-measure on GENIA V1.1 and 4.6 in F-measure on GENIA V3.0. This is largely due to the superiority of the SVM plus sigmoid in our system (improvement of 3.7 in F-measure on GENIA V3.0) over the back-off approach in Shen et al (2003) and the novel name alias feature (improvement of 1.2 in F-measure on GENIA V3.0). Finally, evaluation also shows that the cascaded entity name resolution and the abbreviation resolution contribute 3.4 and 2.1 respectively in F-measure on GENIA V3.0.

Table 2: Performance of our PowerBioNE system

Performance	P	R	F
Shen et al on GENIA V3.0	66.5	66.6	66.6
Shen et al on GENIA V1.1	63.1	61.2	62.2
Our system on GENIA V3.0	72.7	69.8	71.2
Our system on GENIA V1.1	70.4	67.9	69.1

Table 3: Performance of different entity classes on GENIA V3.0

Entity Class	Number of instances in the training data	F
Cell Type	6034	81.8
Lipid	1602	68.6
Multi-Cell	1463	78.1
Protein	21380	77.8
DNA	7538	70.8
Cell Line	3216	68.5
RNA	695	56.2
Virus	873	67.2

One important question is about the performance of different entity classes. Table 3 shows the performance of some of the biomedical entity classes on GENIA V3.0. Of particular interest, our system achieves the F-measure of 77.8 on the class "*Protein*". It shows that the performance varies a lot among different entity classes. One reason may be due to different difficulties in recognizing different entity classes. Another reason may be due to the different numbers of instances in different entity classes. Though GENIA V3.0 provides a good basis for named entity recognition in the biomedical domain and probably the best available, it has clear bias. Table 3 shows that, while GENIA V3.0 is of enough size for recognizing the major classes, such as *"Protein", "Cell Type", "Cell Line", "Lipid*" etc, it is of limited size in recognizing other classes, such as *"Virus"*.

5. ERROR ANALYSIS

In order to further evaluate our system and explore possible improvement, we have implemented an error analysis. This is done by randomly choosing 100 errors from our recognition results. During the error analysis, we find many errors are due to the strict annotation scheme and the annotation inconsistence in the GENIA corpus, and can be considered acceptable. Therefore, we will also examine the *acceptable* F-measure of our system, in particular, the *acceptable* F-measure on the "*protein*" class.

All the 100 errors are classified as follows:

- **Left boundary errors** (14): It includes the errors with correct class identification, correct right boundary detection and only wrong left boundary detection. We find that most of such errors come from the long and descriptive naming convention. We also find that 11 of 14 errors are acceptable and ignorance of the descriptive words often does not make a much difference for the entity names. In fact, it is even hard for biologists to decide whether the descriptive words should be a part of the entity names, such as "*normal", "activated"*, etc. In particular, 4 of 14 errors belong to the "*protein*" class. Among them, two errors are acceptable, e.g. "*classical <PROTEIN>1,25 (OH) 2D3 receptor</PROTEIN>" => "<PROTEIN>classical 1,25 (OH) 2D3 receptor</PROTEIN>"* (with format of "annotation in the corpus => identification made by our system"), while the other two are unacceptable, e.g. *"<PROTEIN>viral transcription factor</PROTEIN> => viral <PROTEIN>transcription factor</PROTEIN>"*.
- **Cascaded entity name errors** (15): It includes the errors caused by the cascaded entity name phenomenon. We find that most of such errors come from the annotation inconsistence in the GENIA corpus: In some cases, only the embedded entity names are annotated while in other cases, the embedded entity names are not annotated. Our system tends to annotate both the embedded entity names and the whole entity names. Among them, we find that 13 of 16 errors are acceptable. In particular, 2 of 16 errors belong to the "*protein*" class and both are acceptable, e.g. *"<DNA>NF kappa B binding site</DNA>" => "<DNA><PROTEIN>NF kappa B</PROTEIN> binding site</DNA>"*.
- **Misclassification errors** (18): It includes the errors with wrong class identification, correct right boundary detection and correct left boundary detection. We find that this kind of errors mainly comes from the sense ambiguity of biomedical entity names and is very difficult to disambiguate. Among them, 8 errors are related with the "*DNA*" class and 6 errors are related with the "*Cell Line*" and "*Cell Type*" classes. We also find that only 3 of 18 errors are acceptable. In particular, there are 6 errors related to the "*protein*" class. Finally, we find that all the 6 errors are caused by misclassification of the "*DNA*" class to the "*protein*" class and all of them are unacceptable, e.g. *"<DNA>type I IFN<DNA>" => "<PROTEIN>type I IFN</PROTEIN>"*.
- **True negative** (23): It includes the errors by missing the identification of biomedical entity names. We find that 16 errors come from the "*other*" class and 10 errors from the "*protein*" class. We also find that the GENIA corpus annotates some general noun phrases as biomedical entity names, e.g. "protein" in "the protein" and "cofactor" in "a cofactor". Finally, we find that 11 of 23 errors are acceptable. In particular, 9 of 23 errors related to the "*protein*" class. Among them, 3 errors are acceptable, e.g. "*the <PROTEIN>protein</PROTEIN> => "the*

protein", while the other 6 are unacceptable, e.g. " *<PROTEIN>80 kDa</PROTEIN> => "80 kDa*".

- **False positive** (15): It includes the errors by wrongly identifying biomedical entity names which are not annotated in the GENIA corpus. We find that 9 of 15 errors come from the "*other*" class. This suggests that the annotation of the "*other*" class is much lack of consistency and most problematic in the GENIA corpus. We also find that 7 of 15 errors are acceptable. In particular, 2 of 15 errors are related to the "*protein*" class and both are acceptable, e.g. "*affinity sites"=> "<PROTEIN>affinity sites</PROTEIN>*".
- **Miscellaneous** (14): It includes all the other errors, e.g. combination of the above errors and the errors caused by parentheses. We find that only 1 of 14 errors is acceptable. We also find that, among them, 2 errors are related with the "*protein*" class and both are unacceptable, e.g. *"<PROTEIN>17 amino acid epitope</PROTEIN>" => "17 <RNA>amino acid epitope</RNA>"*.

From above error analysis, we find that about half (46/100) of errors are acceptable and can be avoided by flexible annotation scheme (e.g. regarding the modifiers in the left boundaries) and consistent annotation (e.g. in the annotation of the "*other*" class and the cascaded entity name phenomenon). In particular, about one third (9/25) of errors are acceptable on the "*protein*" class. This means that the acceptable F-measure can reach about 84.4 on the 23 classes of GENIA V3.0. In particular, the acceptable F-measure on the "*protein*" class is about 85.8. In addition, this performance is achieved without using any extra resources (e.g. dictionaries). With help of extra resources, we think an acceptable F-measure of near 90 can be achieved in the near future.

6. RELATED WORK

Previous approaches in biomedical named entity recognition typically use some domain specific heuristic rules and heavily rely on existing dictionaries (Fukuda et al 1998, Proux et al 1998 and Gaizauskas et al 2000).

The current trend is to apply machine learning approaches in biomedical named entity recognition, largely due to the development of the GENIA corpus. The typical explorations include Kazama et al 2002, Lee et al 2003, Tsuruoka et al 2003, Shen et al 2003. Kazama et al 2002 applies SVM and incorporates a rich feature set, including word feature, POS, prefix feature, suffix feature, previous class feature, word cache feature and HMM state feature. The experiment on GENIA V1.1 shows the F-measure of 54.4. Tsuruoka et al 2003 applies a dictionary-based approach and a naïve Bayes classifier to filter out false positives. It only evaluates against the "*protein*" class in GENIA V3.0, and receives the F-measure of 70.2 with help of a large dictionary. Lee et al 2003 uses a two phase SVM-based recognition approach and incorporates word formation pattern and part-of-speech. The evaluation on GENIA V3.0 shows the F-measure of 66.5 with help of an entity name dictionary. Shen et al 2003 proposes a HMM-based approach and two post-processing modules (cascaded entity name resolution and abbreviation resolution). Evaluation shows the F-measure of 62.2 and 66.6 on GENIA V1.1 and V3.0 respectively.

7. CONCLUSION

In the paper, we describe our HMM-based named entity recognition system in the biomedical domain, named PowerBioNE. Various lexical, morphological, syntactic, semantic and discourse features are incorporated to cope with the special phenomena in biomedical named entity recognition. In addition, a SVM plus sigmoid is proposed to effectively resolve the data sparseness problem. Finally, we present two post-processing modules to deal with cascaded entity name and abbreviation phenomena.

The main contributions of our work are the novel name alias feature in the biomedical domain, the SVM plus sigmoid approach in the effective resolution of the data sparseness problem in our system and its integration with the Hidden Markov Model.

In the near future, we will further improve the performance by investigating more on conjunction and disjunction construction, the synonym phenomenon, and exploration of extra resources (e.g. dictionary).

REFERENCES

Chen and Goodman. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proceedings of the 34th Annual Meeting of the Association of Computational Linguistics (ACL'1996)*. pp310-318. Santa Cruz, California, USA.

Fukuda K., Tsunoda T., Tamura A., and Takagi T. 1998. Toward information extraction: identifying protein names from biological papers. In *Proc. of the Pacific Symposium on Biocomputing'98 (PSB'98)*, 707-718.

Gaizauskas R., Demetriou G. and Humphreys K. 2000. Term Recognition and Classification in Biological Science Journal Articles. In *Proc. of the Computational Terminology for Medical and Biological Applications Workshop of the 2nd International Conference on NLP*, 37-44.

Jacquemin C. 2001. Spotting and Discovering Terms through Natural Language Processing, Cambridge: MIT Press

Kazama J., Makino T., Ohta Y., and Tsujii J. 2002. Tuning Support Vector Machines for Biomedical Named Entity Recognition. In *Proc. of the Workshop on Natural Language Processing in the Biomedical Domain (at ACL'2002)*, 1-8.

Lee K.J. Hwang Y.S. and Rim H.C. Two-phase biomedical NE Recognition based on SVMs. In *Proceedings of the ACL'2003 Workshop on Natural Language Processing in Biomedicine*. pp.33-40. Sapporo, Japan.

MUC6. 1995. Morgan Kaufmann Publishers, Inc. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Columbia, Maryland.

MUC7. 1998. Morgan Kaufmann Publishers, Inc. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Fairfax, Virginia.

Ohta T., Tateisi Y., Kim J., Mima H., and Tsujii J. 2002. The GENIA corpus: An annotated research abstract corpus in molecular biology domain. In *Proc. of HLT 2002*.

Platt J. 1999. Probabilistic Outputs for Support Vector Machines and comparisions to regularized Likelihood Methods. *MIT Press*.

Proux D., Rechenmann F., Julliard L., Pillet V. and Jacq B. 1998. Detecting Gene Symbols and Names in Biological Texts: A First Step toward Pertinent Information Extraction. In *Proc. of Genome Inform Ser Workshop Genome Inform*, 72-80.

Schwartz A.S. and Hearst M.A. 2003. A Simple Algorithm for Identifying Abbreviation Definitions in Biomedical Text. In *Proc. of the Pacific Symposium on Biocomputing (PSB 2003)* Kauai.

Shen Dan, Zhang Jie, Zhou GuoDong, Su Jian and Tan Chew Lim, Effective Adaptation of a Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain, *Proceedings of ACL'2003 Workshop on Natural Language Processing in Biomedicine*, Sapporo, Japan, 11 July 2003. pp49-56.

Tsuruoka Y. and Tsujii J. 2003. Boosting precision and recall of dictionary-based protein name recognition. In *Proceedings of the ACL'2003 Workshop on Natural Language Processing in Biomedicine*. pp.41-48. Sapporo, Japan.

Vapnik V. 1995. *The Nature of Statistical Learning Theory*. NY, USA: Springer-Verlag.

Viterbi A.J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 260-269.

Zhou G.D. and Su J. 2002. Named Entity Recognition using an HMM-based Chunk Tagger. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 473-480.