CSU498 PROJECT
REPORT

NAMED ENTITY RECOGNITION AND CLASSIFICATION IN BIOMEDICAL DOMAIN

Submitted in partial fulfilment of
the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering
Submitted by

AJAS MOHAMMED JANSHER	B080437CS
BINU JASIM T	B080273CS
FEBIN A RASHEED	B080119CS
MUNEEB T H	B080438CS

Under the guidance of
Mr. Abdul Nazeer K. A , Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
NIT CAMPUS PO, CALICUT
KERALA, INDIA 673601
April 12, 2012

Abstract

In this project, we propose a machine learning approach for Named Entity Recognition and Classification(NERC) that can retrieve named entities in biomedical text. We utilized a variety of different linguistic as well as sequential features for this purpose. We trained the system with GENIA 3.0 JNLPBA tagged biomedical text. The assigning of class labels is done using the parameters learned from the training data and we propose a sequential feature based back processing technique to impove the boundary detection. The system achieved very good accuracy with 4 classes of entities namely protein, DNA, RNA and cells.

Contents

Chapter 1

Problem Definition

Build a highly efficient Named Entity Recogniser and Classifier(NERC) in Biomedical domain using machine learning methods. Our input will be large corpus of biomedical text and the Named Entity Recognitin and Classification System should identify different named entities as well as classify them properly into classes such as proteins, DNAs, cell types etc.. .

Sample input: *While specific constitutive binding to the peri-kappa B site is seen in monocytes, stimulation with phorbol esters induces additional, specific binding. Understanding the monocyte-specific function of the peri-kappa B factor may ultimately provide insight into the different role monocytes and T-cells play in HIV pathogenesis.*

Proposed output: While specific constitutive binding to the <*DNA*> *peri-kappa B site* <*DNA*> is seen in <*cell*> *monocytes* <*cell*> , stimulation with phorbol esters induces additional , specific binding . Understanding the monocyte-specific function of the <*protein*> *peri-kappa B factor* <*protein*> may ultimately provide insight into the different role <*cell*> *monocytes* <*cell*> *and* <*cell*> *T-cells* <*cell*> play in HIV pathogenesis.

Chapter 2

Introduction and Related work

2.1 Introduction

Named Entity Recognition is a subtask of Information Extraction which assigns tags such as Person, Place, Organization, Monetary values, quantities etc to atomic elements in a text. It has become more important nowadays due to the large amount of available electronic text, which makes it necessary to build systems that can automatically process and extract information from text.

Large amount of Bio-medical literatures are available online nowadays and for mining these resources for useful information retrieval, named entity recognition and classification of named entities is an essential task. Typical named entities in Biomedical domain are names of proteins, DNAs, RNAs, cell types and genes.

Named Entity Recognition in the newswire domain has achieved near human accuracy, but Biomedical named entity recognition still lags behind because of a number of reasons specific to biomedical entities such as they typically contain long sequence of words than proper nouns.Also they contain digits, special characters etc.. Moreover there is no well defined standard convention for naming biomedical named entities.

Various challenges faced in the biomedical NERC[9][10] domain are:

- Difficult to find the boundry of a given biomedical named entity. Named entities usually have pre-modifiers(.ie. name can be short or long).
  eg: *activated B cell lines* , *47 kDa sterol regulatory element binding factor.*

- Hard to resolve 2 or more biomedical named entities which share one head noun using conjunction constrution.

eg: *91 and 84 kDa proteins.*
In newswire domain 91 is read as digits, whereas in biomedical domain it is read as *91 kDa proteins and 84 kDa proteins.*

- Many cascaded named entities are there in this domain.One named entity may be embedded in another named entity.
eg: *kappa 3 binding factor* is a protein.
<PROTEIN><DNA> kappa 3< /DNA> binding factor < /PROTEIN>

In our project, we use a machine learning based technique for the named entity recognition and classification system. The clear advantage of a machine learning approach is that we don't have to explicitly state different rules for the detection of named entities from common english words. In a machine learning approach the system will be automatically learn to give weightage to different features of named entities will be able to classify future named entities from this data. Also it is not possible to save all the names of named entities and then exhaustively search in this huge data for the identification of named entities. We make use of a large number of features collected from various sources and we use a Naive Bayes Model, which is very simple to implement, but very elegant at the same time if provided with proper features. We also make use of a small set of commonly used words in english to reduce false positives and atlast we do post processing to reduce the error in boundary detection of named entities as much as possible.

2.2 Motivation

With an increasing amount of textual information availabe in biomedical field ,it is necessary to extract information from large chunks of data. This calls for effective literature mining and information extraction that can help biologists and researchers in medical field to gather and make use of the knowledge encoded in text document which cannot be extracted by simple keyword search done in search engines. With an efficient Named Entity Recognition system we will be able to annotate and classify complex datas in a biomedical text which is not possible if done manually. So through machine learning techniques we can train the computer to classify different entities.

Also in our mini project titled *Automatic Question Generation using Natural Language Processing(NLP) tools.* we had used a named entity recognizer provided by stanford university for finding named entities like person, place, organization etc. Also new researches are going on in BioInformatics about data mining and information extraction.This motivated us to explore the possibility of doing a named entity recognizer in Biomedical field.

Chapter 3

Related Work

The earlier ways of doing named entity recognition were rule based and dictionary based approaches[12]. It is very tedious to develop all rules for proper detection of named entities as well as we need the work of domain experts to do this. Nearly all of the works in named entity recognition nowadays are done using machine learning techniques or using a combination of machine learning and rule based approach. The main hurdle before machine learning approach is the lack of training data, but with GENIA 3.0 JNLPBA tagged training set which contains around 2000 abstracts of biomedical texts, it is possible implement a high performing named entity recognition in the biomedical domain.

An example of a hybrid method is Abner which can extracts protein, DNA, RNA, cellline, and cell type. Another machine learning based project is BANNER[4][6] which employes Conditional Random Field for classifying the named entities. Other named entity recognition systems include LingPipe[8] which involves the supervised training of a statistical model or more direct methods like dictionary matching or regular expression matching. Machine learning based approaches typically achieve a performance of 70 percentage[11][10].

In Supervised machine learning based projects, various machine learning techniques have been used like Maximum Entropy models (ME) as in Jon Patrick et.al.[10] and Hidden Markov Models (HMMs) as in Zhou Guodong et.al.[9]. A variety of different features have been designed and utilized by many people for the recognition of biomedical entities. This is often done with the help of domain experts. But named entity recognition can be effectively done by some simple set of features like orthographic features, prefix-suffix features etc. as illustrated in the works of Zhou et.al.[3][2] and Jon Patrick et.al.[10]

Apart from orthographic and morphological features, Jon Patrick et.al.[10] used part of speech tagging features and bigram features. But bigram feature were shown to be not contributing little to the total performance of the system. By using these features implemented on Zhang Les Maximum Entropy Tool Kit, they achieved an f-factor of 68% . Zhou et.al.[9] proposed a rich set of features, including orthographic, morphological,part-of-speech and semantic trigger features. All these features were integrated via a Hidden Markov Model. Furthermore, they proposed a method for biomedical abbreviation recognition and two methods for cascaded named entity recognition. Evaluation on the GENIA V3.02 and V1.1 gave them a performance of 66.5 and 62.5 F-measure respectively. But their work showed that Abbreviate Recognition only improved the system performance by 1.2 f factor. They also provided a rule based post processing to improve the system performance significantly.

Chapter 4

Design

4.1 Training

Inorder to train the system, we are using tagged text as input to the training system. We have already designed a set of features which will be described in section 4.5. Using these features, we have to learn the parameters of the model. That is the probability of occurence of each feature in each of classes proteins, DNAs, RNA, and Cells is learned. We have to store this learned parameters in a persistent data storage. The sentences in the training set are extracted and each token with its tag is passed into the training module.

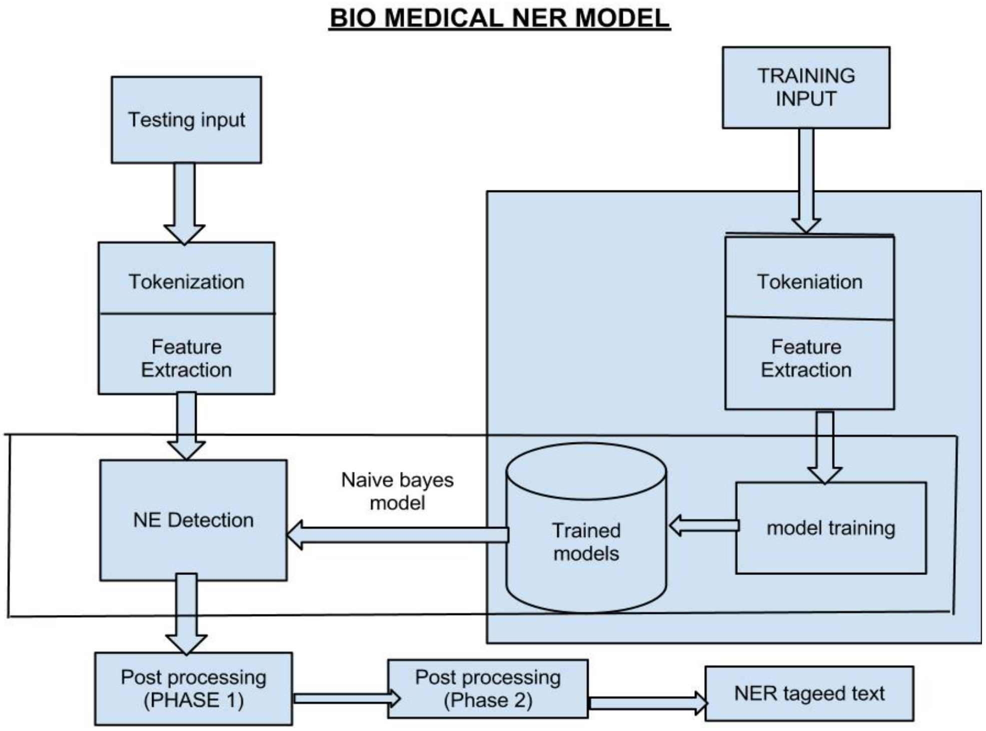
4.2 Testing

Inorder to recognize and classify the named entities present in biomedical text, we pass the input text to the test module. The text to be tested is also tokenized and initially the inference is done on individual tokens using the parameters learned during the training process. The the class with the maximum probability is selected as the class of that particular token. We are coining this phase as forward processing. Now we will again process the tokens to detect the boundary of named entities by backward processing. In backward processing, we will consider the sequential features. That is the probability of occurence of a particular entity before another entity.

4.3 Design Diagram

The system consists of mainly two parts, training and inference(testing). The input to system is natural language text which is to be tokenized and chunked. Then the relevant tokens (all tokens except common english words) is passed to the NER system.

Our basic design is summarized in figure below.



4.4 Methods

4.4.1 Training the System

GENIA corpus

GENIA corpus is the largest annotated biomedical text extracted from MEDLINE database. GENIA corpus version 3.0 consisting of 2000 MEDLINE abstracts has been released with more than 400,000 words and almost 100,000 annotations for biological terms[14]. It contains 48 classes of biological entities. For our training purpose we are using Genia JNLPBA which contains 5 classes - proteins, DNA, RNA, cell type and cell line. It contains around 30,000 proteins, 10,000 DNA, 1000 RNA, 7000 cell types and 4000 cell lines.

4.4.2 Naive Bayes Model

There are many machine learning models for classification such as Support Vector Machines(SVMs), Conditional Random Fields(CRFs), Hidden

Markov Models(HMMs) etc. Naive Bayes Model is a very simple to implement model, at the same powerful enough for supervised classification tasks if provided with enough feature sets as well as sequential information.

So in our problem, we model the probability for each of the features for each class seperately. We will find out the probability *P(f/C_k)* for features $f = f_1, f_2, f_3, ..., f_n$. This is done for all classes separately. Finding these probabilities is the training part. For this we used an annotated training set.

- Model class conditionals *P(f_i/C_k)* and Prior *P(C_k)* for each, where f_i is a feature and C_k is the kth class.

  Posterior *P(C_k/f) α P(f/C_k).P(C_k)*

The Naive Bayes algorithm affords fast, highly scalable model building and inference. It scales linearly with the number of features. Naive Bayes can be used for both binary and multiclass classification problems.[12] Naive Bayes Model is generally used for modelling independent features, but features can be made dependent by using sequential features.

4.4.3 Post Processing

In order to detect the longest chain of an entity, we have to do certain post processing. The traning data is tagged as IOB format. B means the token is at the beginning of a named entity, I means the token is inside the named entity and O means the token is not in the named entity[14].

We employ several post processing such as

1. *single I tag - Which is changed to a B tag.*
2. *O tag between I and I or B and I - O is changed to I tag.*
3. *tags on either sides of 'and' is made the same entity type.*

4.5 Named Entity Features

4.5.1 Orthographic features($\mathbf{F}_o$)

Orthographic features are used to find the similarities between the trained and observed entities. It is also used to find the boundaries of the given input. They are very useful in differentiating many classes such as proteins.[7][9]

Suppose ”$IL - 2$” is the trained data and if the given input are like ”$IL - 12$” or ”$IL - 7$” etc, the system can easily identify that the given input has the same named entity as $IL - 2$.

The feature allCap is useful for detecting abbreviations, and ATCG sequence can differentiate DNAs etc.. Table 3.1 shows the list of orthographic features we have used in our system.

Feature Name	Example
AllCaps	HIV
Greek letters	alpha, beta
ATCG sequence	CGGATC
isDigit	78
singleCap	B M T
allSmall	protein
capsAndDigit	MEK1
InitCapDigit	Am80
initCapLower	Ctx
twoCaps	FasL
lowCapsMix	dNTPs
RomanLetter	I II ii
hyphen	-
letterAndDigit	ETh1
initDigit	15B7
StopWord	in,at

Table 4.1: Orthographic features(F_o)

From the table, we can find that the features, such as *Greek Letter, Roman Digit, ATCG sequence* are specially designed for biomedical domain. Comma, dot, stopword etc are to provide information to detect the boundaries of named entities.

4.5.2 Morphological features($\mathbf{F}^m$)

Morphological information is considered as one of the most important classification features in biomedical NERC. We select prefixes and suffixes from the training data as candidates. We gathered a lot of prefix suffix features based on their frequency in the training data. They have been widely used in named entity recognition systems[3][9][10].

For example, the suffix *-cyte* usually comes in cell types, suffix *-nase* indicates protein, etc.. We extracted 8000 most frequent prefixes and suffixes from the training data.

F_m Name	Prefix/Suffix	Example
Protein	-nase	kinase
	-factor	kappa-binding factor
	activator	DNA-binding transcription activator
	-receptors	IFNgamma receptors
cell type	-ytes	monocytes, leukocytes
	-cells	HeLa cells
	-lines	human MM cell-lines
protein	STAT-	STAT1s
DNA	-gene	interleukin gene
	-site	NF-kappa B site

Table 4.2: Morphological features(F^m)

4.5.3 Sequential features

POS feartures are widely used in the biomedical domains, because many biomedical entities are in lower case, and capitalization information in the biomedical domain is not as evidential as that in the newswire domain[9].

Our training set is IOB tagged. An IOB Tag Set, (also know as BIO or IBO) is a Tag Set composed of (In, Out, Beginning) that is use by string chunking task.[13]
For Example *RBP-Jkappa binding site* is tagged as *RBP-Jkappa<B-DNA> binding<I-DNA> site<I-DNA>*. These are very useful in boundary detection of entities having more than one words as well as cascaded named entities. So instead of Part of Speech tagging, we are using IOB tagging.

We are using the features such as

- ”I-protein always comes after a B-protein or an I-protein”
- ”B-protein only comes at the beginning of an entity name”
- ”I-protein is mostly followed by I-proteins and Other”

4.5.4 Head Noun Features

Entity classes are differentiated based on Head noun semantic trigger(F^{hn})

- Head Noun (F^{hn}) Noun which is used to describe functions and properties of the given compound word is known as head noun. It is the decisive factor for distinguishing entity classes.[7][9][10]

  eg: In the named entity "*activated human B cells*" ,*B cells* is the head noun. *IFN-gamma treatment* and *IFN-gamma activation sequence* are tagged as different entities because there is difference in head nouns.

Class	Head Nouns
PROTEIN	kinase, binding, interleukin activator, protein, interferon receptor, ligand, subunit antibody, complex
DNA	DNA, X-chromosome, breakpoint alpha, promoter, cDNA binding, motif, chromosome promoter, element
Cells	Lymphocyte, macrophage monocyte, neutrophils
RNA	RNA, transcripts

Table 4.3: Head Noun Triggers

4.5.5 Dictionary Features

We are making use of a dictionary of most common words in english such as 'a' 'an' 'the', 'is' 'was' etc. We need not have to do feature checking on these words. We can simply classify them as others. We collected a list of more than 2000 common words in english.

Chapter 5

Implementation

We implemented the naive bayes model from scratch in Java. We are training with Genia JNLPBA which is already annotated with entity classes.

Training:

The features are modelled as binary features meaning a feature is either present in an entity or not present. We calculate the probability of occurence of that feature in a particular class and that frequency is later used for infering the class of testing entity.

All the probabilities of each features is saved in a persistent storage using java serialization. So the system should be trained only once and we can use this probability table during every testing.

In the probability table, we are denoting each class as follows

Smoothing:

When we train the system, the probability for a feature occuring can be zero in a class because of insufficient training samples(P(f_i/C_k)=0). So during testing, the probability of the entity belonging to that class will multiply to zero even if the entity has some other features which satisfies the class.

So inorder to avoid this problem Laplace Smoothing is used.[14]

So maximum likelihood estimation of P(f_i/C_k) ie, $\frac{x}{N}$ is smoothed as $\frac{x+\alpha}{N+\alpha k}$

where N is the number of training set in each class,

x is the number of training set in which the feature f_i occur,

α is the smoothing parameter (typically 1) and

k is the number of classes

Inference:

In the inference process, a testing entity is given to be tagged. We will make use of the probability table obtained from the training process. We will check for each feature in the testing entity, and if the feature is present, the probability of that feature occuring in that class is multiplied. And finally the class with the highest probability is choosen as the class of that entity. The whole probability of each class is calculated using the learned parameters.

Feature Checking:

A class is implemented to check for different features. The methods such as allCaps(), morphologicalFeatures() etc is implemented to check for all types of features. These methods are called from both training and testing classes.

Back Processing:

In the first phase of inference, we used morphological, orthographic and headnoun features, but not sequential features because, at that time no information about B-tag or I-tag or O-tag is available. This processing is termed as forward processing. Inorder to make use of these sequential features, we do inference of the sentence one more time using the tag information obtained from the first inference. But we noted that,we are having more features for words at the end of an entity such as *factor*, *monocyte* etc. So starting from the last entity we test each entity using all the features, namely morphological, orthographic, headnoun and sequential features. This was able to improve the boundary correction as well as cascaded entity recognition.

Chapter 6

Experimentation and Results

6.1 Results

We tested our Named Entity Recognition system with several test cases. It parses the sentence line by line and it gave the tagged output as shown :

1. input: peri-kappa B factor
output : *peri-kappa<B-protein> B<I-protein> factor<I-protein>*

2. input: peri-kappa B site
output: *peri-kappa<B-DNA> B<I-DNA> site<I-DNA>*

3. input: These effects were specifc in that the potentiated phagocytosis of apoptotic neutrophils was completely blocked by the glucocorticoid receptor
output: *These effects were specifc in that the potentiated<I-protein> phagocytosis<I-protein> of apoptotic<B-cell_ type> neutrophils<I-cell_ type> was completely blocked by the glucocorticoid<I-protein> receptor<I-protein>*

6.2 Performance

The performance of our system is evaluated using the metrics precision, recall and F-measure.

Precision is calculated as the ratio of the number of correctly found named entities to the total number of named entities found by our model.

Recall is calculated as the ratio of the number of correctly found named entities to the number of true named entities.
F-measure is defined by the formula [10]

$$F - measure = \frac{2 * precision * recall}{precision + recall}$$

Table 6.1 shows the precision, recall and f-measure of each entities without using sequential features. (During Forward processing)
In forward processing, we are considering each token as standing alone without considering any entity before or after. This is the typical usage of a Naive Bayes Model also called bag of words model. With this naive approach we achieved an accuracy of more than 50% .

Entity	Precision	Recall	F-measure
Protein	0.482357	0.669614	0.560765
DNA	0.373129	0.376801	0.374956
RNA	0.588652	0.272131	0.372197
Cell	0.534434	0.415597	0.467582
Total	0.480499	0.536643	0.507022

Table 6.1: Performance of entity categories in forward processing

The inference after incorporating sequential features as well, the performance of the system improved by around 10% . This shows the effectiveness of the backward processing used by us.

Entity	Precision	Recall	F-measure
Protein	0.482120	0.761789	0.590515
DNA	0.529068	0.540598	0.534770
RNA	0.642857	0.354098	0.456659
Cell	0.653981	0.524742	0.582276
Total	0.526088	0.644767	0.579437

Table 6.2: Performance of entity categories after backward processing

Our System, without using complicated features and implemented using the Naive Bayes Model, has achieved really good performance. Eventhough this is 10% . less than the state of the art techniques, still our back processing method has provided a nearly good substite for Part of Speech Tagging which will consume a lot of time for the tagging as well as require the usage of third party part of speech taggers.

	Precision	Recall	F-measure
Jon Patrick et al.[10]	0.700	0.666	0.682
Zhou et al.[7]	0.727	0.698	0.712
Experimental System	0.526088	0.644767	0.579437

Table 6.3: A comparison to other systems

Chapter 7

Conclusion and Future Work

In this project, we designed and implemented a machine learning based named entity recognition system for recognizing and classifying named entities such as proteins, cell types etc.. in large corpuses of biomedical texts. We gathered together a large number of features for the efficient classification such as orthographic features, morhphological features, sequential features etc.. which were implemented using a Naive Bayes Model. The Naive Bayes Model was implemented all by our own without using any tool kits. We also used a dictionary of 2000 most frequent words in english to avoid classifying common words in english as any of the named entities. We also proposed a novel way of making use of sequential features, a back processing technique, in which the sequential feature is taken as the probability of occurence of a particlar type of entity before other entities.

Our system performs well in the identification and classification of entities as well as in boundary detection. The system achieves an F-score of 60 . The performance with the identification of protein is especially high, because of the availability of large training sets for protein. .but there are some problems like false classification of DNAs as proteins as well. In future work, we will incorporate more features to impove the recall of entities which occurs less frequently in the training data. Collecting more training data is also essential for improving the performance of our system. So we will pursue ways for incorporating incremental training in our system, so that we won't have to train system again with the old trained data.

Also we will switch to graphical models such as HMMs or CRFs which are more powerful than Naive Bayes model, so that it can capture far more sequential feature that what we have provided as features. Collecting more features as well as adding contextual features is also a major aim in our future work.

Bibliography

[1] David Nadeau, "A survey of named entity recognition and classification" , Satoshi Sekine National Research CouncilCanada New York University *Special issue of Lingvistic Investigationes. 30(1) pp. 3-26.*

[2] GuoDong Zhou and Jian Su, "Named Entity Recognition using an HMM-based Chunk Tagger" , Laboratories for Information Technology 21 Heng Mui Keng Terrace Singapore. *Received on August 5, 2003; revised on October 9, 2003; accepted on November 12, 2003. Advance Access publication February 10, 2004.*

[3] ZHOU GuoDong SU Jian, "Exploring Deep Knowledge Resources in Biomedical Name Recognition", Institute for Infocomm Research 21 Heng Mui Keng Terrace Singapore. *In Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 473-480.*

[4] Bob Leaman, "BANNER Named Entity Recognition System", *http://cbioc.eas.asu.edu/banner/index.html, Oct. 10, 2011.*

[5] Tong Zhang and David Johnson, "A Robust Risk Minimization based Named Entity Recognition System" , Watson Research Center Yorktown Heights New York, 10598, USA *in: Proceeding CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*

[6] Robert Leaman Department of Computer Science and Engineering, Arizona State University) Graciela Gonzales (Department of Biomedical Informatics, Arizona State University),"Banner: an executable survey of advances in biomedical named entity recognition." *In Proc, Pacific Symposium on Biocomputing 13:652-663(2008).*

[7] Burr Settles, "Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets", Department of Biostatistics and Medical Informatics University of Wisconsin-Madison Madison, WI, USA. *Appears in Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA). Geneva, Switzerland. 2004.*

[8] Bob Leaman, "LingPipe",*http://alias-i.com/lingpipe/index.html, Oct. 12, 2000.*

[9] Zhang Jie, Shen Dan, Zhou GuoDong, Su Jian and Tan Chew Lim. "Enhancing HMM-based Biomedical Named Entity Recognition by Studying Special Phenomena." *Journal of Biomedical Informatics, Special Issue on Natural Language Processing in Biomedicine: Aims, Achievements and Challenge. 37(6). 411-422. 2004.*

[10] Jon Patrick and Yefeng Wang. "Biomedical Named Entity Recognition System." Sydney Language Technology Research Group. School of Information Technologies, University of Sydney. *The Tenth Australian Document Computing symposium(ADCS 2005) 12 Decembe 2005.*

[11] Daniel Lowd and Pedro Domingos. "Naive Bayes Models for Probability Estimation". Department of Computer Science and Engineering, University of Washington. *Proceedings of the Twenty-Second International Conference on Machine Learning (pp. 529-536), 2005. Bonn, Germany: ACM Press.*

[12] Hirschman L, Morgan AA, Yeh AS. The MITRE Corporation. "Rutabaga by any other name: extracting biological names". *J Biomed Inform. 2002 Aug;35(4):247-59*

[13] Sunita Sarawagi. Department of CSE, IITB. "Efficient Inference on Sequence Segmentation Models." *In: Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*

[14] J.D. Kim, T. Ohta, Y. Tateisi and J. Tsujii, University of Tokyo, Japan. "GENIA corpusa semantically annotated corpus for bio-textmining". *in: Eleventh International Conference on Intelligent Systems for Molecular Biology, Brisbane, Australia June 29 - July 3 2003.*