Biomedical Named Entity Recognition

Andrew Mayne
University of Sydney
amay4350@uni.sydney.edu.au
306137666

Abstract

In this paper we explore several popular feature extraction and machine learning approaches to Named Entity Recognition (NER) in order to identify and classify technical terms in the domain of molecular biology (i.e. 'bio-entities'). We propose employing features based on both orthographic and morphological queues to correctly identify information. We use the large GENIA corpus to train and evaluate our methods and compare both Support Vector Machines (SVM) and Maximum Entropy (ME) based learning.

1 Introduction

The application of natural language processing (NLP) to bioinformatics has become a popular and key research area for computational linguists over the past 15 years. As the wealth of such biomedical information in the form of literature and experimental results increase, there is a rising need to assist researchers in extracting and curating this information. Many new technologies in molecular biology are high-throughput methods, such as gene expression arrays, yeast-2-hybrid screens or protein identification using mass spectrometry, which typically lead to experimental results for thousands of objects at one time (Leser and Hakenberg 2005).

Although newswire entity recognition has led to 'near human' levels of performance, with F-scores in the high 90's, even the state-of-the-art attempts in the biomedical area have only reported results in the low 80's. In newswire entity recognition, we can use simple techniques such as capitalization, gazetteers and context to achieve such high results (van Rijsbergen, 1979). In this paper, we will attempt the BioNLP/NLPBA 2004 shared task (Kim et al., 2004), which provides us with an extended version of the GENIA v3.02 corpus[1], a named entity corpus of MEDLINE[2] abstracts.

The key difference between newswire and biomedical text, and thus the challenge herein, is that biomedical text does not give away such simple orthographic queues, with many genes and proteins lacking capitalization and often only a neighbouring word that separates the two (e.g. HIV-1 is not an entity, but HIV-1 enhancer is a DNA entity). Challenges occur, for example, in ambiguity in the left boundary of entities, caused by descriptive naming, shortened forms due to abbreviation and aliasing within the document (Kim et al., 2004).

To properly evaluate accuracy, the task requires a more fine-grained error analysis beyond F-score statistics. There also remains the difficulty in creating such a large, complex and consistently annotated training data. Many irregularities and ambiguities arise in gene and protein nomenclature, which are mainly a result of a lack of naming conventions and widespread use of synonyms[3]. These ambiguities become apparent at the morphological, syntactic and semantic levels of full text abstracts. Until these types of guidelines are applied consistently and uniformly across all biomedical text, this will continue to be an obstacle for language processing systems. We will ignore such inconsistency in annotation and refer to the GENIA corpus as our 'gold-standard'.

2 Data

Our corpus is derived from GENIA corpus v3.02, which contains 1999 distinct MEDLINE

---

[1] The GENIA v3.02 corpus is available at http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/home/wiki.cgi?page=GENIA+corpus

[2] For more information on MEDLINE (Medical Literature Analysis and Retrieval System Online), see http://www.nlm.nih.gov/pubs/factsheets/medline.html

[3] There have been several guidelines to address this problem, such as MGI's (http://www.informatics.jax.org/mgihome/nomen/short_gene.shtml) and HGNC's (http://www.gene.ucl.ac.uk/nomenclature)

abstracts, retrieved using MeSH[4] terms 'human', 'blood cells' and 'transcription factors'. The Medline database contains approximately 16 million scientific abstracts with a growth rate of about 400,000 articles a year. The GENIA corpus' 36 terminal classes have been simplified into 5 broader classes, *protein, DNA, RNA, cell-line and cell-type*. The first three incorporate several sub-classes, while cell-line and cell-type are interesting in order to make such a task realistic for potential template-filling applications (Kim et al., 2004).

As context is closely linked to this classification problem, the task requires some form of chunking of text to encode such region information. We represent our training and evaluation data in the simple IOB2 format first proposed by Yamada et al. (2000). In IOB2 representation, region information is encoded in the entity class as prefixes "B-", "I-" and "O". "B-" represents the beginning of an entity, "I-" indicates the continuation of a named entity, and "O" indicates that the word doesn't belong to an entity class of interest. For each class C, we then have two IOB classes "B-C" and "I-C". With N named entity classes, the IOB model yields 2N+1 classes for classification by our machine learner (Kazama et al., 2002). In our experiments we also attempted to reduce this to N+1 (discarding IOB tags for machine learning), and then perform repairing for evaluation against our gold standard.

"While a ***nuclear***$_{[B\text{-}protein]}$ ***factor***$_{[I\text{-}protein]}$(s) from both ***peripheral***$_{[B\text{-}cell\_type]}$ ***blood***$_{[I\text{-}cell\_type]}$ ***monocytes***$_{[I\text{-}cell\_type]}$ and ***T***$_{[B\text{-}cell\_type]}$ ***cells***$_{[I\text{-}cell\_type]}$ binds the ***peri-kappa***$_{[B\text{-}DNA]}$ ***B***$_{[I\text{-}DNA]}$ ***site***$_{[I\text{-}DNA]}$, electrophoretic mobility shift assays suggest that either a different protein binds to this site in ***monocytes***$_{[B\text{-}cell\_type]}$ versus ***T***$_{[B\text{-}cell\_type]}$ ***cells***$_{[I\text{-}cell\_type]}$ or that the protein recognizing this ***enhancer***$_{[B\text{-}DNA]}$ ***element***$_{[I\text{-}DNA]}$ undergoes differential modification in ***monocytes***$_{[B\text{-}cell\_type]}$ and ***T***$_{[B\text{-}cell\_type]}$ ***cells***$_{[I\text{-}cell\_type]}$, thus supporting the transfection data."

Figure 1. An example of Bio NER using IOB2

3 Feature Extraction

Affixes, orthographic features and word shapes (ie. word classes) are commonly exploited in this annotation task as they provide important clues to a particular entity type, particularly if bio nomenclature is adhered to. In addition to sequences in the text, there are mutations, motifs, receptors, antibodies, hormones, channels, chromosomal locations and disease loci to consider (Tanabe and Wilbur, 2002). The semantic notion of a gene or protein name is then quite arbitrary, and we must look at a variety of methods to extract useful and informative features from the text.

3.1 Lower case words

Perhaps the most basic of feature extraction techniques, we treat each lower case term seen in the text as an individual feature and it is added to the vocabulary. This is a somewhat naïve approach, given that the corpus is relatively small with a wide variation in the words used for similar entity classes[5]. Naturally, we will generate a large number of features (vocabulary) included in the training data not included in the test set, and vice versa.

3.2 Feature Selection

Some basic feature selection was applied in our experiments, with slight boosts in performance. We calculated term frequency by class, and applied as a feature any (multi-) word expressions, which had a frequency above the corpora average.

3.3 Word class

Orthographic features are typically very useful in named entity recognition, as they can provide clues to an entity simply via capitalisation. Although this feature was tested, this is typically not the case in bio-molecular text, as many gene and protein names are not capitalised.

We use instead a term's word class (WC), first described by Collins (2002), where words are generalised into combinations of an alphabetic character, numerals and hyphenations. Brief word classes (BWC) are also employed, which are a more compact version of word class, collapsing consecutively cased characters into one. We compare this with word class, which implicitly encodes word length information. For example, the 'Interleukin-2' protein has features WC=Aaaaaaaaaaa-0 and BWC=Aa-0, and the 'p56lck' protein has features WC=a00aa and BWC=a0a. We also included other punctuation

[4] MeSH (Medical Subject Headings) can be found at http://www.nlm.nih.gov/mesh/mbinfo.html

[5] In other words, the distribution of the terms in each class is particularly flat, with many terms with a frequency of 1. Lowering the case of the terms in anticipation of this problem boosts performance marginally.

marks such as '+', '-' and apostrophes, as they are often indicative of certain entity types and boundaries. For example, "5'-GTTAAGGTTCGTAGGTCATGGA-3" and "3 kb 5'-flanking region" are indicative of DNA strands, where 5' (five prime) and 3 (three prime) are terminal starts and ends to a sequence[6].

3.4 Modifiers

The use of modifiers was proposed by Zhou and Su (2004), through cascaded entity name resolution in the GENIA corpus, similar to Shen et al.'s (2003) approach. To simplify the annotation task to a simple linear sequential analysis problem, Kim et al. (2004) removed any embedded structures in entity names, leaving the outer most structures. Our corpus' structure has then been reduced from "<RNA><DNA>*CIITA*</DNA> *mRNA* </RNA>" to "<RNA>*CIITA mRNA*</RNA>". We do not then explore cascaded entity names explicitly, but instead apply their use of a modifier pattern to an entity. Modifiers such as 'anti', 'pseudo', 'bi', 'tri', 'multi' and 'non' were used. For example, 'anti' is indicative of protein, as it typically represents an antibody[7].

3.5 Stop words

Stop words or 'stop lists'[8] are words that should not contain important significance towards our entity recognition task, including prepositions, articles and other function words. Stop words frequently occur in all full text articles, and help us separate more meaningful entities from functional words. It is important to note, however, that there is a small percentage of multi token entity sequences that do contain stop words, such as the DNA string "human and mouse gene" or the cell types "erythroid, myeloid and lymphoid cell types".

3.6 Part of Speech

Part of speech (POS) tagging is important in named entity recognition as it provides important semantic clues with which to determine entity inclusion. The eight basic parts of speech, verb, noun, pronoun, adjective, adverb, preposition, conjunction and interjection aid us in determining the subject matter and context within a sentence. In the case of biomedical information however, this does not paint the complete picture.

Many entities in the corpus are made up not only of nouns, but combinations of gerunds, other verbs, prepositions etc., for instance, the DNA entity 'minimal EBNA-2-responsive LMP-1 promoter'. It is thus important to not only look at part of speech in isolation, but also together with other orthographic and morphological features. To obtain accurate parts of speech, we employ a combination of unigram, bigram, trigram and affix taggers with back-off regular expressions. The tagger is then retrained using Brill templates, a technique best summarized as an "error-driven transformation based tagger" (Brill, 1992). Brill tagging is found to give effective performance gains in other subject matter (i.e. news text). We train our POS tagger on the PennBioIE[9] corpus, which is a part of speech tagged corpus specifically for biomedicine.

3.7 Affixes

Both prefixes and suffixes provide great orthographic queues for the entity recognition task, as they indicate different bio molecular attributes and have widely recognized nomenclature. Some common affixes in the corpus include: *-ase, -like, -gene, -ene, -thymic, -sis, -ionic, anti-, propyl-, iso-, di-, epi-*[10].

3.8 Closed Dictionaries

We explored the use of several large dictionaries that help serve as examples of a specific entity class. Exact matching of dictionary entries was used (ignoring case), using a binary search algorithm. We find this simple and very precise NER method to be suitable, however it generally lead to lower levels of recall. The Swissprot protein dictionary[11], Cell Line Database (CLDB)[12] and

---

[6] See 'The Structure of Nucleic Acids' at http://www.vivo.colostate.edu/hbooks/genetics/biotech/basics/nastruct.html

[7] See the defintion of 'antibody' at http://www.nlm.nih.gov/medlineplus/ency/article/002223.htm

[8] We used the Natural Language Toolkit's (NLTK) 'stop list' corpora; containing 127 commonly used English stop words.

[9] PennBioIE available at: http://bioie.ldc.upenn.edu/publications/latest_release/

[10] An extremely comprehensive medical suffix and prefix dictionary, by Eugene McCarthy, can be found at http://www.macroevolution.net/suffix-prefix-dictionary.html

[11] Swissprot: http://www.uniprot.org/downloads

[12] CLDB: http://bioinformatics.istge.it/cldb/cldb.php

the Rfam RNA database[13] were included in our experiments. In addition we used a short list of the Greek alphabet, as these letters are commonly used throughout the abstracts.

3.9 Neighbour Windows

Compound word names are particularly common in the GENIA corpus, with many entity groups containing digits, Greek letters, functional descriptors (*adhesion*), organism identifiers (*hamster*), activity descriptors (*promoting*), placement indicators (*early*), and generic descriptors (*light*) (Tanabe and Wilbur, 2002). We find that adding contextual features vastly improves performance, and explore neighbouring words, POS tags, affixes and word classes in varying range windows.

4 Machine Learning

4.1 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are a popular machine learning approach first presented by Vapnik (1995). Its ability to handle large feature sets and sparse data, in effectively linear training time makes it an attractive machine learner (our final experiments contain over 150,000 unique features). From a given set of feature vectors, SVMs deduce linear combinations of features from appropriate examples called support vectors. These support vectors define a hyperplane in this multidimensional feature space, separating (ideally all) positive examples from all negative examples (Leser and Hakenberg, 2005). Unlike other popular approaches, which typically decompose a multiclass problem into multiple independent binary classification tasks, SVM$^{multi-class}$'s notion of margin yields a direct method for training multiclass predictors, and achieving state-of-art accuracy (Crammer and Singer, 2001).

SVM has been implemented as our baseline machine-learning algorithm, as SVMs have yielded high performance in various classification tasks (Joachims, 1998; Kudo and Matsumoto, 2001). We use SVM^{multiclass} [14], a multiclass distribution of SVM^{light}, with a linear kernel, a high penalty factor (C = 1,000,000) and default epsilon (e = 0.01). Our training for the majority of our experiments ranged from 400 to 2000 iterations[15], and approximately 8 to 15 minutes training time.

4.2 Maximum Entropy Model (MEM)

Maximum entropy models are very popular in information theory, particularly when applied to natural language processing, which was first proposed by Berger et al. (1996). Maximum entropy machine learners choose features based on the highest information gain (i.e. a maximization problem). The MegaM software package[16] is an implementation of maximum likelihood and maximum posterior optimization of the parameters of the model. The algorithms used here are much more efficient than the iterative scaling techniques used in other maxent packages available[17].

Our experiments were run with max iterations of 100 and a minimum change in perplexity (dpp) of -9999. Although these parameters were applied consistently in experiments, earlier experiments would have benefited from a smaller dpp to be comparable to our SVM results (see Table 2).

5 Experimental Results

Experiments were performed with a variety of different feature sets, using both machine classifiers. Each machine learner was trained on 90% of the corpora, and is tested on the final 10%. Although we experimented with reducing classes to N+1, we found this had little effect on performance and has thus not been reported.

5.1 Evaluation Measures

An arguably complete measure of a system's performance is given by the precision-recall curve, and the balanced F-Score (Goutte and Gaussier 2005). Precision may be defined as the probability that an token is relevant given that it is returned by the system, while the recall is the probability that a relevant token is actually returned. Precision (*p*) and Recall (*r*) are calculated as follows:

$$p = \frac{TP}{TP + FP} \qquad r = \frac{TP}{TP + FN}$$

[13] Rfam: http://rfam.sanger.ac.uk/

[14] SVM^{multiclass}: http://svmlight.joachims.org/svm_multiclass.html

[15] Larger, more informative feature sets required fewer iterations and less time, as support vectors (divisions) were easier to detect and construct with more features

[16] MegaM: http://www.cs.utah.edu/~hal/megam/

[17] Such as OpenNLP's Maxent: http://maxent.sourceforge.net/about.html

Neighbour Range	[-1,+1]	[-2,+2]	[-3,+3]	[-4,+4]	[-5,+5]
Part of Speech	**+1.04**	+0.95	+0.64	+0.52	+0.65
Brief word class	+1.67	**+2.08**	+1.98	+1.78	+1.64
Word class	+5.48	+6.67	+6.78	**+7.02**	+6.52
Prefixes [1-5]	+17.79	**+23.98**	+21.42	+20.93	+19.93
Word (lower case)	+17.92	**+24.46**	+22.62	+22.07	+21.23
Suffixes [1-5]	+19.29	**+24.12**	+21.31	+21.34	+19.95
Affixes [1-5]	+21.55	**+27.55**	+24.43	+23.73	+22.55

Table 1. An evaluation of overall (macro averaged) F-scores for neighbour ranges and features, from [-1,+1] to [-5,+5], against a baseline of 33.61 (using pos-tags and word-lower features and the development corpus).

	SVM		MEM
Feature Set	**Dev**	**Test**	**Test**
Word (lower) & POS Tag	33.61	23.98	8.65
+TF Feature Selection	35.49	27.91	6.87
+Modifier	34.38	28.59	6.87
+Stop words	34.41	27.96	6.90
+Word Class	36.79	34.38	23.85
+Brief Word Class	37.03	34.33	27.88
+Dictionaries	37.63	32.49	26.26
+Prefixes	38.05	32.34	27.77
+Suffixes	38.60	34.37	26.90
+Affixes	38.99	32.54	26.75
+Neighbour (WC) [-4,+4]	42.30	40.26	30.91
+Neighbour (POS) [-1,+1]	43.75	42.97	35.19
+Neighbour (BWC) [-2,+2]	44.62	43.70	31.08
+Neighbour (WORD) [-2,+2]	62.72	58.76	32.12
+Neighbour (Prefixes 1-5) [-2,+2]	64.27	60.33	36.53
+Neighbour (Suffixes 1-5) [-2,+2]	64.53	60.45	36.69
+Neighbour (Affixes) [-2,+2]	65.31	61.08	40.60

Table 2. A cumulative evaluation of features for SVM (C=1,000,000 & e=0.01) and MEM models (MaxIter=100 & dpp=-9999). The F-Scores here are macro-averaged across all entities, and each feature is additive.

Overall	*Precision*	*Recall*	*F-Score*
Fully Correct	67.49%	55.78%	61.08%
Correct Left Bdry	74.51%	61.58%	67.43%
Correct Right Bdry	81.99%	67.77%	74.20%

DNA	*Precision*	*Recall*	*F-Score*
Fully Correct	55.49%	45.64%	50.09%
Correct Left Bdry	62.31%	51.25%	56.24%
Correct Right Bdry	74.62%	61.37%	67.35%

Protein	*Precision*	*Recall*	*F-Score*
Fully Correct	74.66%	58.26%	65.45%
Correct Left Bdry	82.26%	64.19%	72.11%
Correct Right Bdry	87.01%	67.90%	76.28%

Cell-Type	*Precision*	*Recall*	*F-Score*
Fully Correct	59.86%	61.46%	60.65%
Correct Left Bdry	65.12%	66.86%	65.98%
Correct Right Bdry	76.11%	78.14%	77.11%

Cell-Line	*Precision*	*Recall*	*F-Score*
Fully Correct	51.20%	37.26%	43.13%
Correct Left Bdry	60.20%	43.81%	50.72%
Correct Right Bdry	69.40%	50.51%	58.47%

RNA	*Precision*	*Recall*	*F-Score*
Fully Correct	60.17%	48.97%	53.99%
Correct Left Bdry	64.41%	52.41%	57.79%
Correct Right Bdry	81.36%	66.21%	73.00%

Table 3. A final system evaluation performed on the full corpus, using word-lower, pos tags, term-frequency feature selection, modifiers, stopwords, word class, brief word class, dictionaries, affixes, neighbouring POS, word class, brief word class and word-lower features. (SVM c=1,000,000 e=0.01)

Predicted as ->	**B-protein**	**I-protein**	**B-cell-type**	**I-cell-type**	**B-cell-line**	**I-cell-line**	**B-DNA**	**I-DNA**	**B-RNA**	**I-RNA**	**O**
B-protein	**4160**	54	13	0	9	0	72	1	6	0	*752*
I-protein	746	**2968**	6	11	5	5	44	67	3	3	*916*
B-cell-type	93	3	**1241**	48	96	10	5	0	0	0	*425*
I-cell-type	42	23	317	**1863**	86	104	2	0	0	0	*554*
B-cell-line	22	0	43	3	**304**	19	1	0	0	0	*108*
I-cell-line	6	7	37	80	92	**614**	0	0	0	0	*153*
B-DNA	204	1	3	0	1	0	**651**	20	0	0	*176*
I-DNA	74	72	1	0	0	0	209	**1107**	0	0	*326*
B-RNA	27	0	0	0	0	0	2	0	**76**	1	*12*
I-RNA	9	18	0	0	0	0	2	4	31	**103**	*20*
O	*1168*	*399*	*193*	*81*	*105*	*28*	*278*	*160*	*28*	*22*	**79185**

Table 4. A confusion matrix for the final system evaluation presented in Table 3

F-Score is calculated as the weighted harmonic average of precision and recall (with β=1)

$$F_\beta = (1+\beta^2)\frac{p\,r}{r+\beta^2 p}$$

In the case of NER, there is a naturally skewed distribution between entities and unrelated words (junk) in the corpus, thus reporting overall F-Scores is irrelevant (in the corpus, approximately 10% of tokens are entities). More fine-grained analysis is necessary, and hence we will look at correct classification within each of the 5 entity classes.

5.2 Confusion Matrices

Confusion Matrices provide us with a detailed description of the performance of the machine classifier. The rows of the matrix describe the 'gold-standard' entity, while the columns tell us the class predicted by the classifier. We present a confusion matrix for our final system in Table 4.

5.3 Discussion

Cell line, cell type and DNA perform particularly badly in tests without neighbouring information (see Table 1). We find that these classes benefit the most from contextual information with a vast improvement in the left boundary F-score. Generally all classes displayed (unsurprisingly) high right boundary F-scores in the majority of experiments, and this is primarily because single token entities were more easily identified, especially proteins (as they are found to have a lot more single tokens in comparison to other classes).

Interestingly, features that were added consecutively in development testing, with positive results, are found to have variable impacts (often detrimental) on the full corpora (see Table 2). It is suspected that this originates from an overfitting of the development data, and may be attributed to the fact that these features are not as robust when met with larger corpora.

We discovered during testing that suffixes produced better results than prefixes in the text. This should come as no surprise to a biologist, as many suffixes dictate a specific entity type (such as protein or cell type). We found that suffixes in the n-gram character range 1-5 were the most helpful in classification.

The phenomena of high precision, lower recall in our final system results presented in Table 3 suggests that we are missing some relevant entities in the corpora (both in development and testing). It indicates that the system is producing a lot of false negatives, and a possible reason for this is the lack of nomenclature guidelines adhered to in the larger corpus. This is seen across all classes, in which the number of answers for each entity was generally 1.2 times the actual number of entities in the gold standard.

It is important to note that when contextual features were added (neighbour windows) running SVM^{multiclass} with the same parameters performed less iterations, with almost double the performance (i.e. this information was less confusing to the SVM, and linearly separable). We report a satisfactory final F-Score of 65.31% in development and 61.08% in our test corpus.

The disappointing maximum entropy results in Table 2 may be attributed to the lack of tweaking in the machine learner, which would've benefited from changes to input parameters, particularly a smaller dpp. A smaller dpp would mean the algorithm would tolerate a lower error

rate and more iterations. However in the time allotted, there was little room for trial and error and optimization of both machine classifiers. Others have reported very good results for named entity recognition using maximum entropy (Chieu and Ng, 2003 and Lin et al., 2004).

The confusion matrix presented in Table 4 produces some interesting features. The most confusion arises from classification of proteins, with many false negatives, particularly with DNA. Confusion between B-entity and I-entities is quite high, which is not surprising. However the far right column paints a more disappointing picture of the many tokens missed as entities altogether (approximately 9%).

Because of the way the system was written, many features found in the training data (word vocabulary, affixes etc) were not found in the test data and became obsolete data. A possible improvement would be to exclude features not found in both training and test data.

6 Conclusion

The system presented in this paper produces some promising results, especially when considering the difficult nature of this entity recognition task. The main difficulty in this area is the lack of nomenclature guidelines that are uniformly adhered to. Nonetheless, these guidelines often change and even if they were adhered to, we would still have a large amount of textual data previously written under old guidelines. Thus, it remains important to improve performance regardless of nomenclature.

Some possible improvements to our task would be to include the expanded-short-form detection algorithm, and name alias resolution, presented in Zhou and Su, 2004. Other papers report boosts in performance by implementing a 'fuzzy' dictionary search, as opposed to matching exact terms, where similar terms are matched (such as lemmatizing words) or using BLAST. Shen et al., 2003, also found that 16.57% of the GENIA corpus had cascaded constructions and as mentioned previously, this information was discarded for our task, which perhaps was an oversight, generating greater confusion among the classes. We reported higher precision and lower recall, and dependent on the task, the system could be tuned; whereby if it is necessary to catch a higher percentage of entities, we will also generate a greater number of false positives. In addition, ten fold cross-validation should've been performed in order to present more reliable results.

We also continue to support the use of Support Vector Machines as a powerful machine learning tool in entity recognition, as we find even with very basic feature sets, it is able to achieve reasonable performance through a large number of iterations (2000+).

References

Berger A, Della Pietra S, Della Pietra V. 1996. *A Maximum Entropy Approach to Natural Language Processing.* Computational Linguistics.

Brill E. 1992. *A Simple Rule-Based Part of Speech Tagger.* Proceedings of the Third Conference on Applied Computational Linguistics (ACL). Trento, Italy.

Chieu H, Ng H. 2003. *Named entity recognition with a maximum entropy approach*. Proceedings of the Seventh Conference on Natural Language Learning, Edmonton, Canada.

Collins M. 2002. *Ranking Algorithms for Named-Entity Extraction: Boosting and the voted perceptron*. Proceedings of the Association for Computational Linguistics Conference. Philadelphia, USA. p 489-496.

Crammer K, Singer Y. 2001. *On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines*. Journal of Machine Learning Research 2:265-292.

Goutte C, Gaussier E. 2005. *A probabilistic interpretation of Precision, Recall and F-Score, with Implication for Evaluation.* Proceedings of the 27th European Conference on Information Retrieval. p 345-359.

Joachims T. 1998. *Text categorization with support vector machines: Learning with many relevant features*. Proceedings of the European Conference on Machine Learning.

Kazama Ji, Makino T, Ohta Y, Tsujii Ji. 2002. *Tuning Support Vector Machiens for Biomedical Named Entity Recongition.* Workshop on Natural Language Processing in the Biomedical Domain. Philadelphia. p 1-8.

Kim J-D, Ohta T, Tsuruoka Y, Tateisi Y, Collier N. 2004. *Introduction to the Bio-Entity Recognition Task at JNLPBA*. International Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-04). p 70-75.

Kudo T, Matsumoto Y. 2001. *Chunking with support vector machines.* Proceedings of NAACL. p 192-199.

Leser U, Hakenberg J. 2005. *What makes a gene name? Named entity recognition in the biomedical literature.* Briefings in BioInformatics 6:357-369.

Lin, Y. F, Tsai, T. H, Chou, W. C, et al. 2004. *A maximum entropy approach to biomedical named entity recognition.* Proceedings of the 4th ACM SIGKDD Workshop on Data Mining in Bioinfomratics'. Seattle, WA. P 56-61

Sang EFTK, Meulder FD. 2003. *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition.* The seventh conference on Natural language learning at HLT-NAACL. Edmonton, Canada. p 142-147.

Shen D, Zhang J, Zhou G, Tan CL. 2003. *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. p 49-56.

Tanabe L, Wilbur WJ. 2002. *Tagging Gene and Protein Names in Full Text Articles.* Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain. Philadelphia, USA. p 9-13.

van Rijsbergen CJ. 1979. *Information Retrieval.* Butterworths, London.

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

Yamada H, Kudo T, Matsumoto Y. 2000. *Using Substrings for Technical Term Extraction and Classification*. IPSJ SIGNotes 2000:77-84.

Zhou G, Su J. 2004. *Exploring Deep Knowledge Resources in Biomedical Name Recognition*. Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications. Geneva, Switzerland. p 70-75.