



Integrating Linguistic Knowledge into a Conditional Random Field Framework to Identify Biomedical Named Entities

Tzong-han Tsai^{1,2}, Wen-Chi Chou¹, Shih-Hung Wu³,
Ting-Yi Sung¹, Jieh Hsiang², Wen-Lian Hsu¹

¹*Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C.*

²*Graduate School of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.*

³*Department and Graduate Institute of CSIE, Chaoyang University of Technology, Taichung County 41349, Taiwan, R.O.C.*

Elsevier use only: Received date here; revised date here; accepted date here

Abstract

As new high-throughput technologies have created an explosion of biomedical literature, there arises a pressing need for automatic information extraction from the literature bank. To this end, biomedical named entity recognition (NER) from natural language text is indispensable. Current NER approaches include: dictionary based, rule based, or machine learning based. Since there is no consolidated nomenclature for most biomedical NEs, any NER system relying on limited dictionaries or rules does not seem to perform satisfactorily. In this paper, we consider a machine learning model, CRF, for the construction of our NER framework. CRF is a well-known model for solving other sequence tagging problems. In our framework, we do our best to utilize available resources including dictionaries, web corpora, and lexical analyzers, and represent them as linguistic features in the CRF model. In the experiment on the JNLPBA 2004 data, with minimal post-processing, our system achieves an F-score of 70.2%, which is better than most state-of-the-art systems. On the GENIA 3.02 corpus, our system achieves an F-score of 78.4% for protein names, which is 2.8% higher than the next-best system. In addition, we also examine the usefulness of each feature in our CRF model. Our experience could be valuable to other researchers working on machine learning based NER.

Keywords: Biomedical named entity recognition, Conditional random fields, Literature mining, Linguistic features

1. Introduction

Biomedical literature available on the web has experienced unprecedented growth in recent years (Figure 1). Therefore, demand for efficiently processing these documents is increasing rapidly. Biomedical named entity recognition is a critical task for automatically mining knowledge from biomedical literature. Since the 1990s, advances in computational and biological methods in various areas such as genome sequence analysis (Venter, 2001), gene identification within sequenced DNA (Korf, Flicek, Duan, & Brent, 2001), and property analysis tools for genes and proteins (Jaakkolay, Diekhansz, & Hausslerz, 2000) have remarkably changed the scale of biomedical research. These large-scale experimental methods produce large quantities of data. When processed, the data can provide actual information about gene expression patterns. Almost every known piece of information pertaining to genes, proteins, and their roles in biological processes is reported somewhere in published biomedical literature. Moreover, the advancement of genome sequencing techniques has created an overwhelming amount of literature on new gene discovery. The abundance of genes and literature produces a major bottleneck for interpreting and planning genome-wide experiments. Thus, the ability to rapidly survey this literature constitutes a necessary step toward both the design and the interpretation of any large-scale experiment. Moreover, automated literature mining offers a yet unexploited opportunity to integrate many fragments of information gathered by researchers from multiple fields of expertise into a complete picture exposing the interrelated roles of various genes, proteins, and chemical reactions in cells and organisms.

During the last few years, there has been a surge of interest in mining biomedical literature, (Andrade & Valencia, 1997; Leek, 1997; Fukuda, Tsunoda, Tamura, & Takagi, 1998; Shatkay, Edwards, Wilbur, & Boguski, 2000; Jenssen, Laegreid, Komorowski, & Hovig, 2001; Hanisch, Fluck, Mevissen, & Zimmer, 2003), ranging from relatively modest tasks such as finding reported gene location on chromosomes (Leek, 1997) to more ambitious attempts to construct putative gene networks based on gene-name co-occurrences within articles (Jenssen, Laegreid,

Komorowski, & Hovig, 2001). Since the literature covers all aspects of biology, chemistry, and medicine, there is almost no limit to the types of information that may be recovered through skillful and pervasive mining. Some possible applications for such efforts include the reconstruction and prediction of pathways, establishing connections between genes and disease, finding the relationships between genes and specific biological functions, and much more.

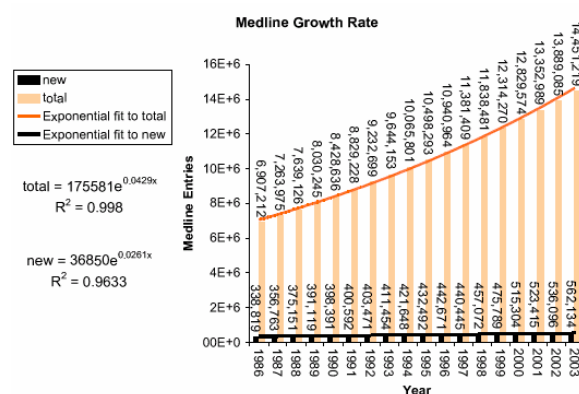


Fig. 1. Growth in Medline over the past 17 years. The hollow portion of the bar is cumulative size up to the preceding year; the solid portion is new additions in that year. (Cohen & Hunter, 2005)

Critical tasks for biomedical literature mining include named entity recognition, tokenization, relation extraction, indexing and categorization/clustering (Cohen & Hunter, 2005). With these technologies, we can construct an environment that aids biologists in the analysis of the output of high-throughput assays and helps the researcher exploit the flood of publications that fills Medline at the rate of 1500 abstracts a day (Cohen & Hunter, 2005). Among these technologies, named entity recognition (NER) is most fundamental. It is defined as recognizing objects of a particular class in plain text. Depending on required application, NER can extract objects ranging from protein/gene names to disease/virus names.

In general, biomedical named entities (NEs) do not follow any nomenclature (Shatkay & Feldman, 2003) and can comprise long compound words and short abbreviations (Pakhomov, 2002). Some NEs contain various symbols and other spelling variations (Hanisch, Fluck, Mevissen, & Zimmer, 2003). On average, any NE of interest has five synonyms.

Biomedical NER is a challenging problem. There are many different aspects to deal with. For example, one can have unknown acronyms, abbreviations, or words containing hyphens, digits, letters, and Greek letters; Adjectives preceding an NE may or may not be part of that NE depending on the context and applications; NEs with the same orthographical features may fall into different categories; An NE may also belong to multiple categories intrinsically; An NE of one category may contain an NE of another category inside it.

To circumvent these difficulties, we need to introduce various language resources. In the biomedical domain, there are more and more curated resources, including lexical resources such as LocusLink (Maglott, 2002) and ontologies such as GO (Ashburner, Ball, & Blake, 2000) and MeSH (NLM, 2003). To exploit these resources, we need a framework. There are three main approaches to NER: dictionary-based, rule-based, and machine learning based. One might think that systems relying solely on dictionary-based recognition could achieve satisfactory performance. However, according to (Cohen & Hunter, 2005), they typically perform quite poorly, with average recall rates in the range of only 10-30%. Rule-based approaches, on the other hand, are more accurate, but less portable across domains (Marquez, Padr'o, & Rodriguez, 2000). Therefore, we chose the machine learning based approach. There are several primary machine learning based methods, which will be summarized in more detail in Section 2.

In this paper, we describe how to construct a framework that can exploit as much useful language information as possible in the recognition of biomedical named entities. We used a well-known machine learning method, conditional random fields (CRF), as the basis of this framework. Our contribution is to integrate most of the linguistic features mentioned in other systems into this CRF framework, and our system achieves the best performance among all Markov model based systems, especially in protein name recognition. Hopefully, our experience of integrating various linguistic features may prove useful for those interested in constructing machine learning based NER system.

2. Related Work

Named Entity Recognition (NER) involves the identification of proper names in text and their classification into different types of named entities. The problem was first defined in the general-language domain in the context of the Message Understanding Conferences (Chinchor, 1998). It is also as subject of much interest for researchers in the biomedical domain. Of course, there are differences between general NER and Biomedical NER. In general-language domains, the set of entities tends to be fairly heterogeneous, ranging from names of individuals to monetary amounts, whereas in the biomedical domain, the set of entities is often restricted to just biomedical proper names such as protein, DNA, RNA, etc. Moreover, there are many well-curated resources in the biomedical domain, which computer linguists can exploit.

Biomedical NER falls into three general classes: dictionary-based approaches (see above), rule-based approaches, and machine learning based approaches. Rule-based approaches generally rely on combinations of regular expressions (templates) to define patterns that match biomedical NEs and rules for extending NE boundaries right and/or left. For example, a rule-based approach might use a regular expression such as `/^[a-z]+[0-9]+$/` (any sequence of one or more lower-case letters followed immediately by any sequence of one or more digits) to recognize that p53 is a gene name. One can also create a rule that uses categorical nouns to classify biomedical named entities. For example, compound words ending in "mRNA" have a high probability of being RNA. While rules of this type can be quite effective, they suffer from the weakness of being domain-specific. Thus, if the system is ported to a new domain, many rules will probably need to be modified. Fukuda's PROPER system is a representative rule-based system that is freely available for download at (Fukuda, 1998). In addition to (Fukuda, Tsunoda, Tamura, & Takagi, 1998), examples of rule-based approaches in the literature include (Narayanaswamy, Ravikumar, & Vijay-Shanker, 2003).

Machine-learning-based approaches are divided into two main categories: classifier-based and Markov model based. Classifier-based models

include decision trees, naïve Bayes, and Support Vector Machines (SVM¹) (Kazama, Makino, Ohta, & J. Tsujii, 2002). Markov model based models include hidden Markov models (HMM) (Zhao, 2004), Maximum Entropy Markov Models (MEMM²) (Finkel, Dingare, Nguyen, Nissim, Manning, & Sinclair, 2004), and CRF (Settles, 2004). Markov model based systems, in particular, excel at solving sequence tagging problems such as speech recognition (Rabiner, 1989) and part-of-speech (POS) tagging (Ratnaparkhi, 1996; Lee, Tsujii, & Rim, 2000).

Of the Markov models, we chose CRF as our framework. Its primary advantage over HMM is its conditional nature, which allows for the relaxation of independent assumptions that HMM requires to ensure tractable inference. Additionally, CRFs avoid the label bias problem (Lafferty, McCallum, & Pereira, 2001) exhibited by MEMMs (McCallum, Freitag, & Pereira, 2000) and other conditional Markov models based on directed graphical models (Wallach, 2004). CRFs outperform both MEMMs and HMMs on a number of real-world sequence labeling tasks (Lafferty, McCallum, & Pereira, 2001; Pinto, McCallum, Wei, & Croft, 2003; Sha & Pereira, 2003). In addition, they are flexible enough to capture many correlated features, including overlapping and non-independent features. We can thus use multiple features with more ease than on an HMM system. Conditional random fields also avoid a fundamental limitation of methods based on combining per-position classifiers.

The major problem faced by machine learning NER systems is the availability of large and consistent annotated corpora. Before 2003, almost all researchers used ad hoc small-scale annotated corpora. In 2003, BioCreative (Valencia & Blaschke, 2004) provided a corpus in which protein and gene names were annotated. It includes 7500 training

sentences, 2500 developing sentences, and 5000 test sentences. They also provided some standardization/tagging guidelines for this task. Shortly after, the Tsujii Lab released the GENIA corpus (Kim, Ohta, Teteisi, & Tsujii, 2003) which annotated 2000 Medline abstracts containing 36 categories of biomedical NEs as defined in the GENIA ontology. At present, many machine learning based NER systems use these two annotated corpora to evaluate their performance. In 2004, the JNLPBA 2004 task (Kim, Ohta, Tsuruoka, Tateisi, & Collier, 2004) provided an extra 404 abstracts that were selected by querying Medline. More biomedical NER systems were released for this task, the best one achieves an F-score around 70%. These results for this task tended to be more objective since the training set and test set did not come from the same query of Medline. The task of annotating biomedical literature is labor-intensive and relies upon hundreds of biomedical experts around the world. Manual curation creates a bottleneck in the amount of information that can be annotated and also causes the problem of inconsistency in the annotation. Following the JNLPBA 2004 task, the new trend is to use various resources, such as web corpora, and more dictionaries, such as LocusLink (Maglott, 2002) and SwissProt (Boeckmann, Bairoch, Apweiler, Blatter, Estreicher, Gasteiger, Martin, Michoud, O'Donovan, Phan, Pilbout, & Schneider, 2003). To test the effectiveness of each resource, we will use the same feature set, represent it in our system and compare it in our experiment.

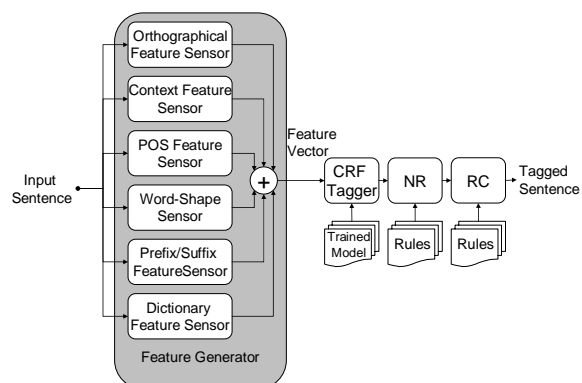


Fig. 2. Data processing flow

¹ Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2(2): 121-167. contains a comprehensive tutorial of SVM

² A good introduction to MEMM can be found in the paper by McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. *Proceedings of the 17th International Conf. on Machine Learning*.

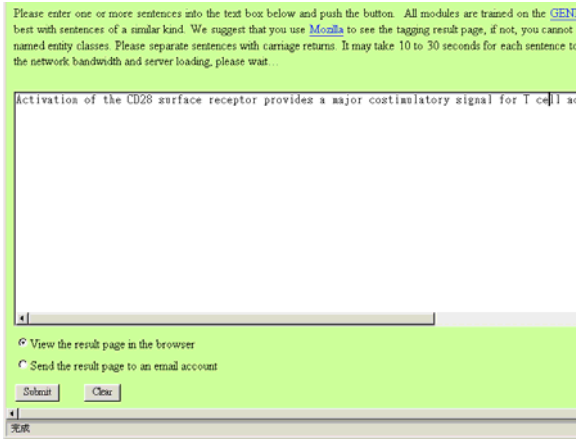


Fig. 3. Web interface of our NER system

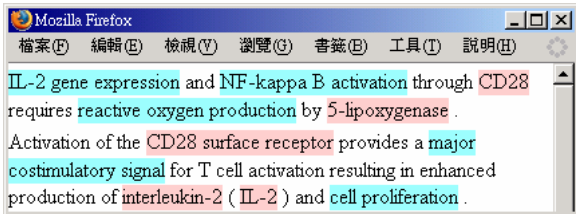


Fig. 4. Output page of our NER system

3. Data Processing Flow

Our system provides a web query interface, as shown in Figure 3. Users can enter up to 10 input sentences in the textbox. After receiving the input sentences, our system will use GPML format (Tateisi, 2001) to annotate the input sentence. If you use Mozilla browser, you will see named entities in the input text highlighted in various colors (Fig. 4). Therefore, with our system, users can easily locate all biomedical named entities of interest. In addition, our system enhances the readability of biomedical text.

After receiving text from the web interface, the Sentence Splitter module will first parse it into sentences. Then it passes the sentences one by one to the Feature Generator. Figure 2 shows the data processing flow for each input sentence. The Feature Generator will pass an input sentence to all installed Feature Sensors, which will extract information from the sentence. Some sensors reference outside sources. For example, Dictionary Feature Sensors uses external dictionaries and POS Feature Sensors invoke

a memory-based POS tagger. Detailed operation of each Feature Sensor will be described in Section 6. Finally, each Feature Sensor will output some information, which is then combined by the Feature Generator. The Feature Generator then passes its output to the CRF tagger.

The CRF tagger uses the algorithm described in Section 4 to assign a tag to all tokens in the input sentence. The tagged sentence then passes through two post-processing modules, the Nested NE Resolution module (NR) and the Reclassification module (RC), which are described in Section 6. This produces our final output.

4. The CRF-based NE Recognizer

In this section, we have integrated two descriptions of CRF-based tagging framework (Sha & Pereira, 2003) (Wallach, 2004) into a coherent version. We use an open source CRF package (Sarawagi & Imran, 2004) which is originally developed on Java. For efficiency, we port it to the .NET framework. We create a J#.NET project and import individual java source files into that project. Most java APIs are supported in J#, but a few are not, in which case we rewrite those parts. According to our experience, both the execution time and memory expense are much reduced. Further details on the performance comparison of .NET and java can be seen in <http://www.gotdotnet.com/team/compare/>.

4.1. Formulation

In the NER problem, we regard each word in a sentence as a token. Each token is associated with a tag that indicates the category of the NE and the location of the token within the NE, for example, B_c , I_c where c is a category. These two tags denote respectively the beginning token and the following token of an NE in category c . In addition, we use the tag O to indicate that a token is not part of an NE. The NER problem can then be phrased as the problem of assigning one of $2n + 1$ tags to each token, where n is the number of NE categories. In the JNLPBA 2004 task, there are 5 named entity categories and 11 tags. For example, one way to tag the phrase *IL-2 gene expression*, *CD28*, and *NF-*

kappa B in a paper is “B-DNA, I-DNA, O, O, B-protein, O, O, B-protein, I-protein”.

4.2. Conditional Random Fields

Conditional random fields (Lafferty, McCallum, & Pereira, 2001) (CRF) are probabilistic frameworks for labeling and segmenting sequential data. They are probabilistic tagging models that provide the conditional probability of a possible tag sequence $\mathbf{y} = y_1, y_2, \dots, y_n$ given the input token sequence $\mathbf{x} = x_1, x_2, \dots, x_n$. We use two random variables \mathbf{X} and \mathbf{Y} to denote any input token sequences and tag sequences, respectively.

A CRF on (\mathbf{X}, \mathbf{Y}) is specified by a vector \mathbf{f} of local features and a corresponding weight vector λ . There are two kinds of local features: the state feature $s(y_i, \mathbf{x}, i)$ and the transition feature $t(y_{i-1}, y_i, \mathbf{x}, i)$, where y_{i-1} and y_i are tags at positions $i-1$ and i in the tag sequence, respectively; i is the input position.

Each dimension of a feature vector is a distinct function. When defining feature functions, we construct a set of real-valued features $b_j(\mathbf{x}, i)$ of the observation to express some characteristic of the empirical distribution of the training data that should also hold true of the model distribution.

An example of such a feature is:

$$b_j(\mathbf{x}, i) = \begin{cases} 1 & \text{if the token at position } i \text{ is "kinase"} \\ 0 & \text{otherwise.} \end{cases}$$

Each feature function f_j (the j th dimension of \mathbf{f}) takes on the value of one of these real-valued observation features $b_j(\mathbf{x}, i)$ if the current state (in the case of a state function) or previous and current states (in the case of a transition function) take on particular values. All feature functions are therefore real-valued. For example, consider the following transition function:

$$t(y_{i-1}, y_i, \mathbf{x}, i) = \begin{cases} b_j(\mathbf{x}, i) & \text{if } y_{i-1} = \text{B-protein and} \\ & y_i = \text{I-protein} \\ 0 & \text{otherwise} \end{cases}$$

To make the notation more uniform, we also write:

$$\begin{aligned} s(\mathbf{y}, \mathbf{x}, i) &= s(y_i, \mathbf{x}, i) \\ t(\mathbf{y}, \mathbf{x}, i) &= \begin{cases} t(y_{i-1}, y_i, \mathbf{x}, i) & i > 1 \\ 0 & i = 1 \end{cases} \end{aligned}$$

for any state feature vector \mathbf{s} and transition feature vector \mathbf{t} . Typically, features depend on the inputs around the given position, although they may depend on global properties of the input. They may also be non-zero only at some positions, for example, the features that pick out the first or last tags.

The CRF's global feature vector for some input sequence \mathbf{x} and its corresponding tag sequence \mathbf{y} is given by the equation:

$$\mathbf{F}(\mathbf{y}, \mathbf{x}) = \sum_i \mathbf{f}(\mathbf{y}, \mathbf{x}, i) \quad (1)$$

where i ranges from the first to the last input position. The conditional probability distribution defined by the CRF is then

$$p_\lambda(\mathbf{Y}, \mathbf{X}) = \frac{\exp \lambda \mathbf{F}(\mathbf{Y}, \mathbf{X})}{\mathbf{Z}_\lambda(\mathbf{X})} \quad (2)$$

where

$$\mathbf{Z}_\lambda(\mathbf{x}) = \sum_{\mathbf{y}} \exp \lambda \mathbf{F}(\mathbf{y}, \mathbf{x})$$

Any positive conditional distribution $p(\mathbf{Y} | \mathbf{X})$ that obeys the *Markov property*

$$p(Y_i | \{Y_j\}_{j \neq i}, \mathbf{X}) = p(Y_i | Y_{i-1}, Y_{i+1}, \mathbf{X})$$

can be written in the form (2) for appropriate choice of feature functions and weight vector (Hammersley & Clifford, 1971).

The most probable sequence for input sequence \mathbf{x} is

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} p_\lambda(\mathbf{y} | \mathbf{x}) = \arg \max_{\mathbf{y}} \exp \lambda \mathbf{F}(\mathbf{y}, \mathbf{x})$$

because $\mathbf{Z}_\lambda(\mathbf{x})$ does not depend on \mathbf{y} . According to (1), $\mathbf{F}(\mathbf{y}, \mathbf{x})$ can be decomposed into a sum of local feature vectors, $\mathbf{f}(\mathbf{y}, \mathbf{x}, i)$. In addition, when $i > 1$, $\mathbf{f}(\mathbf{y}, \mathbf{x}, i)$ can be rewritten to $\mathbf{f}(y_{i-1}, y_i, \mathbf{x}, i)$, which is a function of the tags at $i-1$ th and i th position. Therefore, we can find the most likely \mathbf{y} with the Viterbi search algorithm. The detailed calculation can be carried out by matrix computation as mentioned by (Wallach, 2004).

Given a training set $T = \{(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})\}_{k=1}^N$, we train a CRF by maximizing the log-likelihood of T , which we assume as fixed for the rest of this section:

$$\begin{aligned} L_\lambda &= \sum_k \log p_\lambda(\mathbf{y}^{(k)} | \mathbf{x}^{(k)}) \\ &= \sum_k [\lambda \mathbf{F}(\mathbf{y}^{(k)}, \mathbf{x}^{(k)}) - \log \mathbf{Z}_\lambda(\mathbf{x}^{(k)})] \end{aligned}$$

This function is concave, guaranteeing convergence to global maximum. To perform this optimization, we seek the zero of the gradient.

$$\nabla L_\lambda = \sum_k [F(\mathbf{y}^{(k)}, \mathbf{x}^{(k)}) - E_{p_\lambda(\mathbf{Y}|\mathbf{x}^{(k)})} F(\mathbf{Y}, \mathbf{x}^{(k)})] \quad (3)$$

The maximum log-likelihood of T is reached when the empirical average of the global feature vector equals its model expectation. This expectation $E_{p_\lambda(\mathbf{Y}|\mathbf{x}^{(k)})} F(\mathbf{Y}, \mathbf{x}^{(k)})$ can be computed efficiently using a variant forward-backward algorithm. For a given \mathbf{x} , define the transition matrix for position i as

$$M_i[y_{i-1}, y_i] = \exp \lambda f(y_{i-1}, y_i, \mathbf{x}, i)$$

Let f_i be a local feature, $f_j[y_{i-1}, y_i]^{(i)} = f_j(y_{i-1}, y_i, \mathbf{x}, i)$,

$F_j(\mathbf{y}, \mathbf{x}) = \sum_i f_j(y_{i-1}, y_i, \mathbf{x}, i)$, Then

$$E_{p_\lambda(\mathbf{Y}|\mathbf{x}^{(k)})} F_j(\mathbf{Y}, \mathbf{x}^{(k)}) = \sum_{\mathbf{y}} p_\lambda(\mathbf{y} | \mathbf{x}^{(k)}) F_j(\mathbf{y}, \mathbf{x}^{(k)}) \quad (4)$$

If we calculate it in a naïve fashion, we can easily find that such calculations are intractable since there are $n^{|\mathcal{Y}|}$ possible corresponding label sequences if \mathbf{x} has n tokens.

Fortunately, the right-hand side of (4) can be rewritten as

$$\sum_i \sum_{y', y} p_\lambda(y_{i-1} = y', y_i = y | \mathbf{x}^{(k)}) f_j(y', y, \mathbf{x}^{(k)}, i), \quad (5)$$

eliminating the need to sum over $n^{|\mathcal{Y}|}$ sequences. Furthermore, we calculate $p_\lambda(y_{i-1} = y', y_i = y | \mathbf{x}^{(k)})$ in a forward-backward algorithm like fashion.

Defining forward and backward vectors - α_i and β_i respectively:

$$\alpha_i = \begin{cases} \alpha_{i-1} M_i & 0 < i \leq n \\ 1 & i = 0 \end{cases}$$

$$\beta_i^T = \begin{cases} M_{i+1} \beta_{i+1}^T & 0 < i \leq n \\ 1 & i = n \end{cases}$$

Then, equation (5) can be rewritten as follows:

$$\sum_i \frac{\alpha_{i-1} (f_j^{(i)} M_i) \beta_i^T}{Z_\lambda(\mathbf{x}^{(k)})}$$

$$Z_\lambda(\mathbf{x}^{(k)}) = \alpha_n \cdot 1^T$$

4.3. Training method

(Lafferty, McCallum, & Pereira, 2001) used iterative scaling algorithms for CRF training, following earlier work on maximum entropy models for natural language (Berger, Pietra, & Pietra, 1996). But they are too slow when many correlated features are involved. Our CRF package uses limited-memory

quasi-Newton (L-BFGS) (Nocedal & Wright, 1999) method to perform comparably on very large problems (about millions of features).

Newton methods for nonlinear optimization use second order information to find search directions. It is not practical to obtain exact second order information for CRF training. L-BFGS estimates the curvature numerically from previous gradients and updates. (Malouf, 2002) indicates that L-BFGS performs well in maximum-entropy classifier training. More detailed description of this method can be found in (Nocedal & Wright, 1999).

5. Linguistic Features

Feature selection is critical to the success of machine learning approaches. In this section, we describe the features available for our system. We will illustrate how to calculate values of feature functions. The effectiveness of each feature is also discussed.

5.1. Orthographical Features

Table 1 lists some orthographical features used in our system. In our experience, ALLCAPS, CAPSMIX, and INITCAP are more useful than others.

5.2. Context Features

Words preceding or following the target word may be useful for determining its category. Take the sentence “The IL-2 gene localizes to bands BC on mouse Chromosome 3” for example. If the target word is “IL-2,” the following word “gene” will help the CRF model to distinguish IL-2 gene from the protein of the same name. Obviously, the more context words analyzed the better and more precise the results. However, widening the context window quickly leads to an explosion of the number of possibilities to calculate. In our experience, a suitable window size is five. (That is, the two preceding words, the current word, and the two following words). Such a window size is also suitable for most tagging problem, such as POS tagging (Giménez & Márquez, 2004).

Table 1
Orthographical features

Feature name	Regular Expression
INITCAP	[A-Z].*
CAPITALIZED	[A-Z][a-z]+
ALLCAPS	[A-Z]+
CAPSMIX	.*[A-Z][a-z].* .*[a-z][A-Z].*
ALPHANUMERIC	.*[A-Za-z].*[0-9].* .*[0-9].*[A-Za-z].*
SINGLECHAR	[A-Za-z]
SINGLEDIGIT	[0-9]
DOUBLEDIGIT	[0-9][0-9]
INTEGER	-?[0-9]+
REAL	-?[0-9][.,]+[0-9]+
ROMAN	[IVX]+
HASROMAN	.*\\b[IVX]+\\b.*
HASDASH	.*-.*
INITDASH	-.*
ENDDASH	.*-
PUNCTUATION	[,:;?!+]
QUOTE	[\"'\"]

5.3. Part-of-speech Features

Part of speech information is quite useful for identifying named entities. Verbs and prepositions usually indicate an NE's boundaries, whereas nouns not found in the dictionary are usually good candidates for named entities. Unlike context-word features, where extending the window increases accuracy, including extended POS information often introduces more noise. Our experience indicates that five is also a suitable window size. The MBT POS tagger (Daelemans, Zavrel, Berck, & Gillis, 1996) is used to provide POS information. We trained it on GENIA 3.02p and achieves 97.85% accuracy.

5.4. Word Shape Features

Certain kinds of named entities, which belong to the same class, are similar to each other—for example, IL-2 and IL-4. So we have come up with

simple way to normalize all similar words. According to our method, capitalized characters are all replaced by 'A', digits are all replaced by '0', non-English characters are replaced by '_' (underscore), and non-capitalized characters are replaced by 'a'. For example, Kappa-B will be normalized as "Aaaaa_A". To further normalize these words, we shorten consecutive strings of identical characters to one character. For example, "Aaaaa_A" is normalized to 'Aa_A'. After applying the first normalization method, the two proteins "IL-1" and "IL-2" will get the same feature value. After applying the second normalization method, "IL-2" and "IL-21" will get the same feature value. Therefore, applying our normalization methods will group similar names into the same NE class.

5.5. Prefix and Suffix Features

Some prefixes and suffixes can provide good clues for classifying named entities. For example, words which end in "~ase" are usually proteins. However, short prefixes or suffixes are too common to be of any help in classification. For example, it would be difficult to guess to which category a word ending in "~es" belongs. In our experience, the acceptable length for prefixes and suffixes is 3-5 characters. The longer the prefix or suffix, the fewer matches will occur.

5.6. Dictionary Features

Depending on the quality of a given dictionary, our system uses one of two different lexicon features to estimate the possibility of a token in a biomedical named entity.

Exact Matching

We start by setting one or more dictionary feature functions for every token in the corpus. These features, described as "dictionary name"+length of token string, tell us whether a token matches a dictionary entry and whether it is part of a multi-token string that matches a compound NE or NE phrase in the dictionary. Imagine our corpus contains the token string "interlukin-2 gene," a two-word compound NE. Given that we are using dictionary D, the token interluken-2's D1 and D2 features will be

activated because it appears in the dictionary twice, both by itself and as part of a two-word compound NE. The token “gene,” on the other hand, will only have its D2 feature activated because it does not appear alone in the dictionary. We proceed as described above, setting features cumulatively for token strings of up to five words.

We use this exact matching method with two dictionaries. The first is from the BioCreative 2004 task (Valencia & Blaschke, 2004). This dictionary consists mainly of proteins and genes, which effectively reduces false negatives. The second is an edited version of the 240,000-word Longman dictionary from which we excluded any terms found in biomedical NEs in the GENIA corpus. In our experiments, we found that when these two dictionaries are used individually, no improvement in performance results, but when they are used together, they function complementarily, effectively improving performance. Therefore, this feature function input aids our CRF model in differentiating between biomedical and non-biomedical terms.

Distance Measurement

In the biomedical domain, it’s difficult to find a dictionary which contains all possible variations of biomedical names. For example, it’s difficult to find a dictionary which contains all names of IL-family proteins. Therefore, it is useful to measure the similarity (distance) between tokens and words in an external biomedical dictionary and set this as a feature function. The following definition and description mainly comes from (Cohen & Sarawagi, 2004).

Let D be a dictionary of entity names and d be a distance metric for entity names. Define $g_{D/d}(x)$ to be the minimum distance between a token x and any entity name e in D :

$$g_{D/d}(x) = \min_{e \in D} d(e, x)$$

For example, if D contains the two strings “frederick flintstone” and “barney rubble”, and d is the Jaro-Winkler distance metric (Winkler, 1995), then $g_{D/d}(\text{"Fred"}) = 0.84$ and $g_{D/d}(\text{"Fred please"}) = 0.4$, since $d(\text{"Fred"}, \text{"frederick flintstone"}) = 0.84$ and $d(\text{"Fred please"}, \text{"frederick flintstone"}) = 0.4$.

Here, we extended the distance features described to tokens—i.e., for each distance d , we compute as a

feature of token x the minimum distance between x and an entity in the dictionary D . These features are less natural for tokens than for segments, but turned out to be surprisingly useful (Cohen & Sarawagi, 2004), perhaps because weak partial matches to entity names are informative.

We used two dictionaries: Gene and GeneAlias, which are extracted from Swissprot and LocusLink, and three distance functions from the SecondString open source software package (Cohen & Ravikumar, 2003): Jaccard, Jaro-Winkler, and SoftTFIDF. We describe the relationships among these three distance functions in the next paragraph.

Briefly, the Jaccard distance between two sets S and S' is $|S \cap S'| / |S \cup S'|$: in SecondString, this is applied to strings by treating them as sets of words. The Jaro-Winkler distance is a character-based distance, rather than a word-based distance. It is based on the number of characters which appear in approximately the same position in both strings. TFIDF is another word-based measure. As with Jaccard distance, TFIDF scores are based on the number of words in common between two strings; however, rare words are weighted more heavily than common words. SoftTFIDF is a hybrid measure, which modifies TFIDF by considering words with small Jaro-Winkler distance to be common to both strings.

5.7. Noneffective Features

Below we list some features mentioned in other biomedical NER systems that do not seem to improve our system.

POS+Target Word Feature

Some combinations of POSes and target words (W) are good indicators of NEs. For example, a phrase constructed by an adjective + a protein name (nuclear BSAP) is usually still a protein name. Therefore, we created a feature group as follows:

$$w_i = \text{and } p_{i-1} =$$

where w_i means the current word and

p_{i-1} means the preceding word's POS tag

In our experiment, this configuration actually reduces performance by 0.60%. We find that certain combinations of $w_i = \text{and } p_{i-1}$ are not useful. For

example, if the target word is not a noun, then this feature generally does not boost accuracy. In general, it is useful to look at the target word rather than its combination with the previous word.

Verb+Word Feature

The proximity of certain verbs appearing before w_i can sometimes be used to determine the target word's category. For example, in the phrase "infect noncycling B cells," the appearance of the verb "infect" prior to our target word, "cells," may help disambiguate w_i 's tag from cell line to cell type.

We also design an experiment for this feature; however, performance drops by 0.4%. We presume that, similar to the POS+Target Word feature, only certain combinations can help assign the tag of the target word.

The above two composite features were used by (Finkel, Dingare, Nguyen, Nissim, Manning, & Sinclair, 2004). However, they did not report the contribution of features individually. Thus, from this experiment, we could only conclude that composite features may indeed be quite useful so long as they are carefully selected.

6. The post-processing method

In our system, we have two post-processing modules. One uses NE nesting rules to fix boundary errors. The other uses the rightmost word of an NE to fix classification errors. We describe them in the following sections.

6.1. Nested NE Resolution

According to (Shen, Zhang, Zhou, Su, & Tan, 2003), 16.57% of entity names in GENIA V3.02 have nested constructions, e.g.

<DNA> <protein> IL-2 </protein> gene </DNA>

In the JNLPBA 2004 task, all nested mark-up tags are removed leaving only outside annotation. Even though our system is trained to recognize outside NEs, in some cases it still makes errors. Take the phrase "IL-2 gene expression" for example. Our system tends to tag "IL-2" as a stand-alone protein

because in the GENIA corpus, "IL-2" appears alone far more frequently than it appears collocated with "gene." When our system is trained, it calculates a higher value [0.36] for $P(y_i=B\text{-protein} | x_i=IL-2)$ than that [0.18] for $P(y_i=B\text{-DNA} | x_i=IL-2)$. Furthermore, $P(y_i=O | x_i=expression, y_{i-1}=O)$ is also higher than $P(y_i=O | x_i=expression, y_{i-1}=I\text{-DNA})$. Referring to Table 2, one can see that $P(Y\#1|X)$ is higher than $P(Y\#2|X)$. So, even though "gene" has high probability of being tagged as "I-DNA," the sequence with the highest probability is still [B-Protein, O, O], which is denoted as $Y\#1$ in the following table.

Table 2
Probability table for the phrase "IL-2 gene expression"

X	IL-2	gene	expression	P(Y X)
Y#1	B-protein	O	O	0.066
$P(y_i x_i,y_{i-1})$	0.36	0.32	0.57	
Y#2	B-DNA	I-DNA	O	0.032
$P(y_i x_i,y_{i-1})$	0.18	0.80	0.22	

*Y#n denotes the nth candidate for Y

To resolve the nested NEs, (Zhou, 2004) uses a pattern-based module. He found six patterns of nested entity name constructions. Here, we extract all NEs which contain one or more shorter NEs, and use them to produce rules. For example, if we extract from the following NE

<DNA> <protein> IL-2 </protein> gene </DNA>

Then we can produce a context-free rule: < protein > +gene →< DNA > .

This module can be used in any machine learning Biomedical NER system, which uses GENIA corpus as their training/test dataset. For example, (Zhou, 2004) improved F-score by 3.1% on his HMM-based Biomedical NER system. We have used it with our CRF-based system, and it improved F-score by 0.36%.

6.2. Reclassification based on the rightmost word

We count the occurrences of a word x appearing in the rightmost position of all NEs in each category. Let the maximum occurrence be n , and the

Table 3
Basic statistics for the data sets

	# abstracts	# sentences	# words
Training Set	2,000	18,546	472,006 (236.00/abs) (22.97/sen)
Test Set	404	3,856	96,780 (239.55/abs) (22.72/sen)

corresponding category c . The total number of occurrences of x in the rightmost position of an NE is T . c/T is the consistency rate of x . According to our analysis of the training set of the JNLPBA 2004 data, 75% of words have a consistency rate of over 95%. We record this 75% of words and their associated categories in a table. After testing, we crosscheck all the rightmost words of NEs found by our system against this table. If they match, we overwrite the NEs' categories with those from the table.

7. Experiment

7.1. Datasets

In our experiment, to compare with other biomedical NER systems, we use two corpora. One is the GENIA version 3.02 corpus (Kim, Ohta, Teteisi, & Tsujii, 2003) and the other is the dataset used in the JNLPBA 2004 shard task. The GENIA corpus is formed from a controlled search on MEDLINE using the MeSH terms 'human', 'blood cells' and 'transcription factors'. In that search, 2,000 abstracts are selected and annotated by hand according to a small taxonomy of 48 classes. Among these, 36 terminal nodes in the taxonomy are used for annotation. Several biomedical NER systems use the GENIA corpus as training and test data (Lee, Hwang, & Rim, 2003; Zhou, Zhang, Su, Shen, & Tan, 2004).

In the JNLPBA 2004 shared task, the GENIA corpus is still used as training data. However, the original 36 classes are simplified to 5 classes: protein, DNA, RNA, cell line and cell type.

To simplify the annotation task to a simple linear sequential analysis problem, embedded structures have been removed leaving only the outermost structures. Consequently, a group of coordinated

entities involving ellipsis are annotated as one structure as in the following example:

... in [lymphocytes] and [T- and B-lymphocyte] count in ...

In the example, "*T- and B-lymphocyte*" is annotated as one structure but involves two entity names, "T-lymphocyte" and "B-lymphocyte", whereas "lymphocytes" is annotated as one entity and involves as many entity names.

To ensure objectivity of the evaluation, 404 newly-annotated MEDLINE abstracts from the GENIA project are used as test data. They were annotated with the same five entity categories. Half of these abstracts were from the same domain as the training data and the other half were from the super-domain of "blood cells" and "transcription factors". We hope this can provide an important test of generality in the methods used. The basic statistics for the training and test data are summarized in Table 3.

7.2. Evaluation Methodology

Results are given as F-scores using a modified version of the CoNLL evaluation script and are defined as $F = (2PR)/(P + R)$, where P denotes the precision and R denotes the recall. P is the ratio of the number of correctly found NE chunks to the number of found NE chunks, and R is the ratio of the number of correctly found NE chunks to the number of true NE chunks. The script outputs three sets of F-scores according to exact boundary match, right and left boundary matching. In the right boundary matching only right boundaries of entities are considered without matching left boundaries and vice versa.

Table 4
Absolute (and relative) frequencies for NEs in each data set.

	protein	DNA	RNA	cell type	cell line	all
Training Set	30,269 (15.1)	9,533 (4.8)	951 (0.5)	6,713 (3.4)	3,830 (1.9)	51,301 (25.7)
Test Set	5,067 (12.5)	1,056 (2.6)	118 (0.3)	1,921 (4.8)	500 (1.2)	8,662 (21.4)

Table 5
NER performance of each configuration on the JNLPBA 2004 data. F_{ort} , F_{con} , F_{pos} , F_{wor} , $F_{pre/suf}$, F_{dic} , F_{pos+T} , and F_{vW} denote orthographical, context, POS, word-shape, POS+Target word, and Verb+Word features, respectively. NR and RC have been explained in Section 3.

	F_{ort}	F_{con}	F_{pos}	F_{wor}	$F_{pre/suf}$	F_{dic}	F_{pos+T}	F_{vW}	NR	RC	Precision	Recall	F-score
conf#1	√	√	√	√	√	√					68.6	70.9	69.7
conf#2	√	√	√	√	√	√	√				68.2	70.0	69.1
conf#3	√	√	√	√	√	√		√			67.6	70.5	69.0
conf#4	√	√	√	√	√	√			√		69.0	71.2	70.1
conf#5	√	√	√	√	√	√			√	√	69.1	71.3	70.2

7.3. Results

In Table 5, the NER performance of each configuration is compared based on the feature groups and post-processing methods used in it. We can see that our NER model with the first six feature groups (conf#1) achieves an F-score of 69.7. Obviously, the ineffective features listed in Section 6.7 lowered the F-score by at most 0.7% (conf#2 and conf#3). We can see that both post-processing methods slightly improve the F-score in conf#4 and conf#5.

In Table 6, we list precision, recall, and F-scores for each category of NE. We can see that F-scores for protein and cell-type are comparably high. We think this is because protein and cell type are among the top three most frequent categories in the training set (as shown in Table 4). One notices, however, that although DNA is the second most frequent category, it does not have a high F-score. We think this discrepancy is due to the fact that DNA names are commonly used in proteins, causing a substantial overlap between these two categories. RNA's performance is comparably low because its training set is much smaller than other categories. Cell line's performance is the lowest since it overlaps heavily with cell type and its training set is also very small.

Table 6
NER performance of each NE category on the JNLPBA 2004 data

NE category	Precision	Recall	F-score
protein	67.9	75.9	71.7
DNA	68.3	65.6	67.0
RNA	58.6	63.6	61.0
cell line	55.8	56.4	56.1
cell type	78.8	66.7	72.3
Overall	69.1	71.3	70.2

In Table 7, we compare our system with other pure Markov model based systems in the JNLPBA 2004 task. Our system performs slightly better than (Finkel, Dingare, Nguyen, Nissim, Manning, & Sinclair, 2004) and (Settles, 2004). This is probably due to the nature of CRF and the effective use of external dictionaries.

Table 7
NER performance comparison of Markov model based Systems on the JNLPBA 2004 data

System	Precision	Recall	F-score
Our System (CRF)	69.1	71.3	70.2
Finkel et al., 2004 (MEMM)	68.6	71.6	70.1
Settles et al., 2004 (CRF)	69.3	70.3	69.8
Zhao, 2004 (HMM)	61.0	69.1	64.8

In Table 8, we compare our system with others that use GENIA V3.02 as their training/test corpus. Like other systems, we apply 10-fold cross validation on GENIA V3.02. Since there’s no agreement on which NE should be used to evaluate these systems, we use the most popular and representative categories: protein and DNA. Our system outperforms other systems of this type by an increase in protein F-score of at least 2.6%. For DNA names, our performance is close to the best system.

Table 8
Protein and DNA name recognition performance on the GENIA V3.02 corpus

System	Protein	DNA
Our System (CRF)	78.4	66.3
Zhou et al., 2004 (HMM)	75.8	63.3
Lee et al., 2003 (2Phase SVM)	70.6	66.4

In Table 9, we report F-scores for different boundary matching criteria: exact boundary match (Exact Match), right boundary match (Right Match) and left boundary match (Left Match). We can see that with relaxed boundary matching, the F-scores increase from 4.3% (Left Match) to 8.2%. Relaxed boundary matching may cause some NEs that have descriptive preceding adjectives or rightmost head nouns to be tagged correctly. However, it also results an increase of other types of errors. We think the degree of relaxation should be based on NER applications.

Table 9
F-score of each NE category for different matching criteria on the JNLPBA 2004 data

NE category	Exact Match	Left Match	Right Match
protein	71.7	77.4	79.0
DNA	67.0	69.5	75.4
RNA	61.0	64.2	76.4
cell line	56.1	59.7	66.1
cell type	72.3	73.6	81.9
Overall	70.2	74.5	78.4

8. Analysis and discussion

Recognition disagreement between our system and GENIA is caused by the following two factors:

1. Annotation problems in GENIA corpus:

Although there are no inter-annotator agreement results for the GENIA corpus, we have found that some studies of inter-annotator agreement for biomedical named entities have measured agreement between 87% (Hirschman, 2003) and 89% (Demetrious & Gaizauskas, 2003). We further summarize the annotation problems into four sub-problems. All these problems are caused mainly by inconsistent annotation.

(a) Preceding adjective problem

Some descriptive adjectives are annotated as parts of the following NE, but some are not. In fact, it is even hard for biologists to decide whether descriptive adjectives, such as “normal”, “activated”, etc, should be part of entity names. Take “human” for example. Of the 1790 times it occurred before or at the beginning of an entity in the training data, it was not recognized as a part of an entity 110 times. But in test data, in only one out of 130 appearances is it excluded from an NE. This irregularity really confuses NER systems and weakens the reliability of evaluation results on the GENIA corpus.

(b) Nested NE problem

In GENIA, we found that in some instances only embedded NEs are annotated while in other instances, only the outside NE is annotated. However, according to the GENIA tagging guidelines, the outside NE should be tagged. For example, in the training set of the JNLPBA 2004 data, in 59 instances of the phrase “IL-2 gene”, “IL-2” is tagged as a protein 13 times, while in the other 46 it is tagged as a DNA. This irregularity can confuse machine learning based systems.

(c) Cell-line/cell-type confusion

NEs in the cell line class are from certain cell types. For example, the HeLa cell line is from human origin or cellular products. Given the abbreviated content of an abstract, it is difficult even for an expert to distinguish them. In GENIA, most instances of “granulocytic colonies” are tagged as cell line; however, in the phrase “stimulated primary murine bone marrow cells to form granulocytic colonies in

vitro”, the same phrase “granulocytic colonies” is tagged as a cell type.

(d) Missing tag

In the training data of the JNLPBA 2004 data, NEs of each category, especially of cell line, are not tagged. This incorrect annotation causes a large number of false negatives, especially in the cell line category. We see many instances of “T cell”, “Peripheral blood neutrophil”, and “NK cell” not tagged as cell lines.

2. System recognition errors

The other cause of disagreement is our system’s tagging errors. We categorize errors into four subtypes:

(a) Misclassification

Some protein molecules or regions are misclassified as DNA molecules or regions. These errors may be solved by exploiting more context information—that is, more understanding of the sentences.

(b) Coordinated phrase problem

In GENIA, most conjunction phrases are tagged as single NEs. However, conjunction phrases are usually composed of several NEs, punctuation, and conjunctions such as “and”, “or” and “but not”. The construction of a conjunction phrase is sometimes long and complicated, containing more than three NEs and mixed with other words. Therefore, our system sometimes only tags one of these NE components. For example, in the phrase “c-Fos and c-Jun family members”, only “c-Jun family members” is tagged as a protein by our system, while in GENIA, the whole phrase is tagged as a protein.

(c) False positive

Some entities appeared without accompanying a specific name, for example, only mention about “the epitopes” rather than which kind of epitopes. The GENIA corpus tends to ignore these entities, but their contexts are similar to the entities with specific names, therefore, our system sometimes incorrectly recognizes them as an NE.

9. Conclusion

Our system successfully integrates linguistic features into the CRF framework. We have made quite an effort to find the appropriate use for every kind of linguistic information available. Our experimental results indicate that most of these linguistic features are effective besides some composite ones. Through these broad linguistic features and the nature of CRF, our system outperforms state-of-the-art Markov model based systems, especially in the recognition of protein names.

From our analysis, it is still difficult to recognize long, complicated NEs and to distinguish between two highly overlapped NE classes, such as cell-line and cell-type. This is due to the fact that, biomedical texts have complicated sentence structures and involve more expert knowledge than general domain news articles. Another serious problem is the annotation inconsistency, which confuses the machine learning models and makes the evaluation difficult.

Certain errors, such as those in boundary identification, are more tolerable if the main purpose is to discover relations between NEs. We shall exploit more linguistic features such as composite features and external features. Finally, to reduce human annotation efforts and to alleviate the scarcity of available annotated corpora, we shall develop machine learning techniques to learn from Web corpora in different biomedical domains.

Acknowledgments

We would like to thank Sunita Sarawagi and Imran Mansuri for answering my questions about the CRF package. We are grateful for the support of National Science Council under GRANT NSC94-2752-E-001-001.

References

- Andrade, M. A. & Valencia, A. (1997). Automatic annotation for biological sequences by extraction of keywords from

- MEDLINE abstracts. Development of a prototype system. *Proceedings of the ISMB' 97*.
- Ashburner, M., Ball, C. A., & Blake, J. A. (2000). Gene ontology: Tool for the unification of biology. *Nature Genet.* 25: 25-29.
- Berger, A., Pietra, S. A. D., & Pietra, V. J. D. (1996). A Maximum Entropy Approach to Natural Language Processing. *Computer Linguistics* 22: 39-71.
- Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M. C., Estreicher, A., Gasteiger, E., Martin, M. J., Michoud, K., O'Donovan, C., Phan, I., Pilboud, S., & Schneider, M. (2003). The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.* 31(1): 365-370.
- Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2(2): 121-167.
- Chinchor, N. (1998). Message Understanding Conference Proceedings. *Proceedings of the Message Understanding Conference*, http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/proceedings_index.html.
- Cohen, K. B. & Hunter, L. (2005). Natural Language Processing and Systems Biology. *Artificial Intelligence and Systems Biology*. W. Dubitzky and F. Azuaje, Springer.
- Cohen, W. W. & Ravikumar, P. (2003). Secondstring: An open-source Java toolkit of approximate string-matching techniques. <http://secondstring.sourceforge.net>.
- Cohen, W. W. & Sarawagi, S. (2004). Semi-Markov Conditional Random Fields for Information Extraction. *Proceedings of the NIPS' 04*.
- Daelemans, W., Zavrel, J., Berck, P., & Gillis, S. (1996). MBT: A Memory-Based Part of Speech Tagger-Generator. *Proceedings of the Fourth Workshop on Very Large Corpora*.
- Demetriou, G. & Gaizauskas, R. (2003). Corpus resources for development and evaluation of a biological text mining system. *Proceedings of the Third Meeting of the Special Interest Group on Text Mining*.
- Finkel, J., Dingare, S., Nguyen, H., Nissim, M., Manning, C., & Sinclair, G. (2004). Exploiting Context for Biomedical Entity Recognition: From Syntax to the Web. *Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)*.
- Fukuda, K., Tsunoda, T., Tamura, A., & Takagi, T. (1998). Toward information extraction: identifying protein names from biological papers. *Proceedings of the Pacific Symposium on Biocomputing '98*.
- Fukuda, K. I. (1998). KEX download site. <http://www.hgc.jp/service/tool/doc/KeX/intro.html>.
- Giménez, J. & Márquez, L. (2004). SVMTool: A general POS tagger generator based on Support Vector Machines. *Proceedings of the In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal.
- Hammersley, J. & Clifford, P. (1971). Markov fields on finite graphs and lattices. *Unpublished manuscript*.
- Hanisch, D., Fluck, J., Mevissen, H., & Zimmer, R. (2003). Playing biology's name game: identifying protein names in scientific text. *Proceedings of the Pacific Symposium on Biocomputing '03*.
- Hirschman, L. (2003). Using biological resources to bootstrap text mining.
- Jaakkola, T., Diekhansz, M., & Hausslerz, D. (2000). A discriminative framework for detecting remote protein homologies. *J. Comp. Biol.* 7(1/2): 95-114.
- Jenssen, T. K., Laegreid, A., Komorowski, J., & Hovig, E. (2001). A literature network of human genes for high-throughput analysis of gene expression. *Nature Genet.* 28: 21-28.
- Kazama, J., Makino, T., Ohta, Y., & J. Tsujii (2002). Tuning support vector machines for biomedical named entity recognition. *Proceedings of the ACL-02 Workshop on Natural Language Processing in Biomedical Applications*.
- Kim, J.-D., Ohta, T., Teteisi, Y., & Tsujii, J. i. (2003). GENIA corpus - a semantically annotated corpus for biotextmining. *Bioinformatics* 19(suppl. 1).
- Kim, J.-D., Ohta, T., Tsuruoka, Y., Tateisi, Y., & Collier, N. (2004). Introduction to the Bio-Entity Task at JNLPBA. *Proceedings of the the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*.
- Korf, I., Flicek, P., Duan, D., & Brent, M. R. (2001). Integrating genomic homology into gene structure prediction. *Proceedings of the ISMB 2001*.
- Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data. *Proceedings of the ICML' 01*.
- Lee, K.-J., Hwang, Y.-S., & Rim, H.-C. (2003). Two phase biomedical NE Recognition based on SVMs. *Proceedings of the ACL-03 Workshop on Natural Language Processing in Biomedicine*.

- Lee, S.-Z., Tsujii, J.-i., & Rim, H.-C. (2000). Lexicalized Hidden Markov Models for Part-of-Speech Tagging. *Proceedings of the COLING 2000*.
- Leek, T. R. (1997). Information Extraction Using Hidden Markov Models. Master's thesis, Department of Computer Science, University of California, San Deigo.
- Marquez, L., Padr'o, L., & Rodriguez, H. (2000). A Machine Learning Approach to POS Tagging. *Machine Learning* 39: 59-91.
- Maglott, D. (2002). Locuslink: a directory of genes. *NCBI Handbook*: 19-1 to 19-16.
- Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. *Proceedings of the CoNLL-02*.
- McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. *Proceedings of the 17th International Conf. on Machine Learning*.
- McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. *Proceedings of the ICML' 00*.
- Narayanaswamy, M., Ravikumar, K. E., & Vijay-Shanker, K. (2003). A biological named entity recognizer. *Proceedings of the PSB' 03*.
- NLM (2003). Mesh: Medical subject headings. <http://www.nlm.nih.gov/mesh/>.
- Nocedal, J. & Wright, S. J. (1999). *Numerical Optimization*, Springer.
- Pakhomov, S. (2002). Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical text. *Proceedings of the the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Pinto, D., McCallum, A., Wei, X., & Croft, W. B. (2003). Table extraction using conditional random fields. *Proceedings of the ACM SIGIR' 03*.
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2): 257--286.
- Ratnaparkhi, A. (1996). A Maximum Entropy Model for Part-of-Speech Tagging. *Proceedings of the In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-1)*.
- Sarawagi, S. & Imran, M. (2004). the CRF package. <http://crf.sourceforge.net>.
- Settles, B. (2004). Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets. *Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)*.
- Sha, F. & Pereira, F. (2003). Shallow parsing with conditional random fields. *Proceedings of the HLT & NAACL' 03*.
- Shatkay, H., Edwards, S., Wilbur, W. J., & Boguski, M. (2000). Genes, themes and microarrays: using information retrieval for large-scale gene analysis. *Proceedings of the ISMB' 00*.
- Shatkay, H. & Feldman, R. (2003). Mining the biomedical literature in the genomic era: an overview. *Journal of Computational Biology* 10(6): 821-855.
- Shen, D., Zhang, J., Zhou, G., Su, J., & Tan, C.-L. (2003). Effective Adaptation of a Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain. *Proceedings of the ACL-03 Workshop on Natural Language Processing in Biomedicine*.
- Tateisi, Y. (2001). GENIA Resources Quick Start Manual. <http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/GPML/>.
- Valencia, A. & Blaschke, C. (2004). BioCreAtIvE - Critical Assessment of Information Extraction systems in Biology. *Proceedings of the BioCreAtIvE' 04*, <http://www.mitre.org/public/biocreative/>.
- Venter, J. C. (2001). The sequence of the human genome. *Science* 291: 1304-1351.
- Wallach, H. (2004). Conditional Random Fields: An Introduction. http://www.inference.phy.cam.ac.uk/hmw26/papers/crf_intro.pdf.
- Wallach, H. M. (2004). Conditional Random Fields: An Introduction. http://www.inference.phy.cam.ac.uk/hmw26/papers/crf_intro.ps.
- Winkler, W. E. (1995). *Business Survey methods*, Wiley.
- Zhao, S. (2004). Named Entity Recognition in Biomedical Texts using an HMM Model. *Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)*.
- Zhou, G. (2004). Recognizing Names in Biomedical Texts using Hidden Markov and SVM plus Sigmoid. *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA) held in conjunction with COLING 2004*.
- Zhou, G., Zhang, J., Su, J., Shen, D., & Tan, C. (2004). Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics* 20: 1178-1190.