# Using Maximum Entropy to Extract Biomedical Named Entities without Dictionaries

**Tzong-Han Tsai, Chia-Wei Wu, and Wen-Lian Hsu**
Institute of Information Science, Academia Sinica
Nankang, Taipei, Taiwan 115
{thtsai, cwwu, hsu}@iis.sinica.edu.tw

## Abstract

Current NER approaches include: dictionary-based, rule-based, or machine learning. Since there is no consolidated nomenclature for most biomedical NEs, most NER systems relying on limited dictionaries or rules do not perform satisfactorily. In this paper, we apply Maximum Entropy (ME) to construct our NER framework. We represent shallow linguistic information as linguistic features in our ME model. On the GENIA 3.02 corpus, our system achieves satisfactory F-scores of 74.3% in protein and 70.0% overall without using any dictionary. Our system performs significantly better than dictionary-based systems. Using partial match criteria, our system achieves an F-score of 81.3%. Using appropriate domain knowledge to modify the boundaries, our system has the potential to achieve an F-score of over 80%.

## 1 Introduction

Biomedical literature available on the web has experienced unprecedented growth in recent years. Therefore, demand for efficiently processing these documents is increasing rapidly. There has been a surge of interest in mining biomedical literature. 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 much more.

Critical tasks for biomedical literature mining include named entity recognition (NER), tokenization, relation extraction, indexing and categorization/clustering (Cohen and Hunter, 2005). Among these technologies, 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 NEs do not follow any nomenclature (Shatkay and Feldman, 2003) and can comprise long compound words and short abbreviations (Pakhomov, 2002). Some NEs contain various symbols and other spelling variations. 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 tackle these challenges, researchers use three main approaches: dictionary-based, rule-based, and machine learning. In biomedical domain, there are more and more well-curated resources, including lexical resources such as LocusLink (Maglott, 2002) and ontologies such as MeSH (NLM, 2003). One might think that dictionary-based systems relying solely on these resources could achieve satisfactory performance. However, according to (Pakhomov, 2002), 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. Therefore, we chose the machine learning approach.

Various machine learning approaches such as ME (Kazama et al., 2002), SVM (Kazama et al., 2002; Song et al., 2004), HMM (Zhao, 2004) are applied to NER. In this paper, we chose ME as our framework since it is much easier to represent various features in such a framework. In addition, ME models 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. ME-based tagger, in particular, excel at solving sequence tagging problems such as POS tagging (Ratnaparkhi, 1997), general English NER (Borthwick, 1999), and Chunking (Koeling, 2000).

In this paper, we describe how to construct a ME-based framework that can exploit shallow linguistic information in the recognition of biomedical named entities. Hopefully, our experience in integrating these features may prove useful for those interested in constructing machine learning based NER system.

## 2 Maximum Entropy Based Tagger

### 2.1 Formulation

In the Biomedical 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, and the 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 2*n* + 1 tags to each token, where *n* is the number of NE categories. 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*].

### 2.2 Maximum Entropy Modeling

ME is a flexible statistical model which assigns an outcome for each token based on its history and features. ME computes the probability $p(o|h)$ for any $o$ from the space of all possible outcomes $O$, and for every $h$ from the space of all possible histories $H$. A history is all the conditioning data that enables one to assign probabilities to the space of outcomes. In NER, history can be viewed as all information derivable from the training corpus relative to the current token. The computation of $p(o|h)$ in ME depends on a set of binary-valued features, which are helpful in making predictions about the outcome. For instance, one of our features is: when all alphabets of the current token are capitalized, it is likely to be part of a biomedical NE. Formally, we can represent this feature as follows:

$$f(h,o) = \begin{cases} 1: & \text{if } \textit{W0-AllCaps(h)=true} \\ & \text{and o=}\textit{B-protein} \\ 0: & \text{otherwise} \end{cases} \tag{1}$$

Here, *W0-AllCaps(h)* is a binary function that returns the value true if all alphabets of the current token in the history *h* are capitalized. Given a set of features and a training corpus, the ME estimation process produces a model in which every feature $f_i$ has a weight $\alpha_i$. From (Berger et al., 1996), we can compute the conditional probability as:

$$p(o|h) = \frac{1}{Z(h)} \prod_i \alpha_i^{f_i(h,o)} \tag{2}$$

$$Z(h) = \sum_o \prod_i \alpha_i^{f_i(h,o)} \tag{3}$$

The probability is given by multiplying the weights of active features (i.e., those $f_i(h,o) = 1$). The weight $\alpha_i$ is estimated by a procedure called Generalized Iterative Scaling (GIS) (Darroch and Ratcliff, 1972). This method improves the estimation of weights iteratively. The ME estimation technique guarantees that, for every feature $f_i$, the expected value of ıequals the empirical expectation of ıin the training corpus.

As noted in (Borthwick, 1999), ME allows users to focus on finding features that characterizes the problem while leaving feature weight assignment to the ME estimation routine. When new features, e.g., syntax features, are added to ME, users do not need to reformulate the model as in the HMM model. The ME estimation routine can automatically calculate new weight assignments. More complete discussions of ME including a description of the MEs estimation procedure and references to some of the many successful computational linguistics systems using ME can be found in the following introduction (Ratnaparkhi, 1997).

### 2.3 Decoding

After having trained an ME model and assigned the proper weights ıto each feature $f_i$, decoding (i.e., marking up) a new piece of text becomes simple. First, the ME module tokenizes the text. Then, for each token, we check which features are active and combine $\alpha_i$ of the active features according to Equation 2. Finally, the probability of a tag sequence $y_1 ... y_n$ given a sentence $w_1 ... w_n$ is approximated as follows:

$$p(o_1 ... o_n | w_1 ... w_n) \approx \prod_{j=1}^{n} p(o_j | h_j) \qquad (4)$$

where $h_j$ is the context for word $w_j$. The tagger uses beam search to find the most probable sequence given the sentence. Sequences containing invalid subsequences are filtered out. For instance, the sequence [B-protein, I-DNA] is invalid because it does not contain an ending token and these two tokens are not in the same category. Further details on the beam search can be found in `http://www-jcsu.jesus.cam.ac.uk/~tdk22/project/beam.html`.

## 3 Linguistic Features

### 3.1 Orthographical Features

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

Table 1: Orthographical features

| Feature name | Regular Expression |
|---|---|
| INITCAP | [A-Z].* |
| CAPITALIZED | [A-Z][a-z]+ |
| ALLCAPS | [A-Z]+ |
| CAPSMIX | .*[A-Z][a-z].* \|<br>.*[a-z][A-Z].* |
| ALPHANUMERIC | .*[A-Za-z].*[0-9].* \|<br>.*[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]+ |
| HASDASH | .*-.* |
| INITDASH | -.* |
| ENDDASH | .*- |
| PUNCTUATION | [,.;:?!-+] |
| QUOTE | [“’] |

### 3.2 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 ME 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.

### 3.3 Part-of-speech Features

Part of speech information is quite useful for identifying NEs. Verbs and prepositions usually indicate an NEs boundaries, whereas nouns not found in the dictionary are usually good candidates for named entities. Our experience indicates that five is also a suitable window size. The MBT POS tagger (Daelemans et al., 1996) is used to provide POS information. We trained it on GENIA 3.02p and achieves 97.85% accuracy.

### 3.4 Word Shape Features

NEs in the same category may look similar (e.g., 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’,

Table 2: Basic statistics for the data set

| Data | # abs | # sen | # words |
|---|---|---|---|
| GENIA 3.02 | 2,000 | 18,546 | 472,006 (236.00/abs) |

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".

### 3.5 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. In our experience, the acceptable length for prefixes and suffixes is 3-5 characters.

## 4 Experiment

### 4.1 Datasets

In our experiment, we use the GENIA version 3.02 corpus (Kim et al., 2003). Its basic statistics is summarized in Table 2. Frequencies for all NE classes in it are showed in Table 3.

### 4.2 Results

In Table 4, one can see that F-scores for protein and cell-type are comparably high. We believe this is because protein and cell type are among the top three most frequent categories in the training set (as shown in Table 3). 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. RNAs performance is comparably low because its training set is much smaller than those of other categories. Cell lines performance is the lowest since it overlaps heavily with cell type and its training set is also very small.

In Table 5, one can see that, using the partial matching criterion, the precision rates, recall rates, and F-scores of protein names are all over 85%. The overall F-Score is 81.3%. The table also shows that 83.9% of our systems suggestions correctly identify at least one part of an NE, and that our system tags at least one part of 78.9% of all NEs in the test corpus. The precision rate in all categories is over 80%, showing that , by using appropriate post-processing methods, our system can achieve high precision in all NE categories.

Table 4: NER performance of each NE category on the GENIA 3.02 data (10-fold CV)

| NE category | Precision | Recall | F-score |
|---|---|---|---|
| protein | 74.1 | 74.5 | 74.3 |
| DNA | 65.9 | 54.4 | 59.6 |
| RNA | 75.3 | 48.0 | 58.6 |
| cell line | 65.4 | 51.4 | 57.6 |
| cell type | 72.3 | 69.1 | 70.7 |
| Overall | 72.0 | 67.9 | 70.0 |

Table 5: Partial matching performance on the GENIA 3.02 corpus (10-fold CV)

| NE category | Precision | Recall | F-score |
|---|---|---|---|
| protein | 85.3 | 85.5 | 85.4 |
| DNA | 80.3 | 66.3 | 72.7 |
| RNA | 84.0 | 53.0 | 65.0 |
| cell line | 80.9 | 63.3 | 71.1 |
| cell type | 83.1 | 79.4 | 81.2 |
| Overall | 83.9 | 78.9 | 81.3 |

In Table 6, we compare our system with two dictionary-based systems. One exploits handcrafted rules based on heuristics and protein name dictionaries (Seki and Mostafa, 2003). We denote this system as "rule + dictionary". The other system (Tsuruoka and Tsujii, 2004) has two configurations: the first one exploits patterns to detect protein names and their fragments, which is denoted as "dictionary expansion"; the second one further applies naive Bayes filters to exclude erroneous detections, which is denoted as "dictionary expansion + filters". One can see that our system performs better than these dictionary/heuristic systems by a wide margin. The basic "rule + dictionary" system achieves only 54.4% recall. By expanding the original dictionary ("dictionary expansion"), they improve the recall rate to 68.1%. After applying post processing filters ("dictionary expansion + filters"), the recall rate dropped slightly, but precision increased by 25.7%. Still, our system performs better than the best dictionary-based system by 7.6%.

Table 3: Frequencies for NEs in each data set

| Data | protein | DNA | RNA | cell type | cell line | All |
|---|---|---|---|---|---|---|
| GENIA 3.02 | 30,269 | 9,533 | 951 | 6,718 | 3,830 | 51,301 |

Table 6: Performance comparison between systems with and w/o dictionaries in extracting protein names on the GENIA 3.02 data

| System | Precision | Recall | F-score |
|---|---|---|---|
| our system | 74.1 | 74.5 | 74.3 |
| rule + dictionary | 42.6 | 54.4 | 47.8 |
| dictionary expansion | 46.0 | 68.1 | 54.8 |
| dictionary expansion + filters | 71.7 | 62.3 | 66.6 |

## 5 Analysis and discussion

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

1. Preceding adjective problem
   Some descriptive adjectives are annotated as parts of the following NE, but some are not.

2. Nested NEs
   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 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.

3. Cell-line/cell-type confusion
   NEs in the cell line class are from certain cell types. It is difficult even for an expert to distinguish them.

**System recognition errors:**

1. Misclassification
   Some protein molecules or regions are misclassified as DNA molecules or regions. These errors may be solved by exploiting more context information.

2. Coordinated phrases
   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". 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.

3. False positives
   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.

## 6 Conclusion

Our system successfully integrates linguistic features into the ME framework. Without using any biomedical dictionaries, our system achieves a satisfactory F-score of 74.3% in protein and 70.0% overall. Our system performs significantly better than dictionary-based systems. Using partial match criteria, our system achieves an F-score of 81.3%. That means, with appropriate boundary modification algorithms (with domain knowledge), our system has the potential to achieve an F-score of over 80%.

It is still difficult to recognize long, complicated NEs and to distinguish between two overlapping NE classes, such as cell-line and cell-type. This is because biomedical texts have complicated syntax and involve more expert knowledge than general domain news articles. Another serious problem is annotation inconsistency, which confuses machine learning models and makes evaluation difficult. Certain errors, such as those in boundary identification, are more tolerable if the main purpose is to discover relationships between NEs.

In the future, we will exploit more linguistic features such as composite features and external features. Finally, to reduce human annotation effort and to alleviate the scarcity of available annotated corpora, we will develop machine learning techniques to learn from Web corpora in different biomedical domains.

## Acknowledgements

We are grateful for the support of National Science Council under GRANT NSC94-2752-E-001-001.

## References

A. Berger, S. A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computer Linguistics*, 22:39–71.

A. Borthwick. 1999. *A Maximum Entropy Approach to Named Entity Recognition*. Phd thesis, New York University.

K. Bretonnel Cohen and Lawrence Hunter. 2005. Natural language processing and systems biology. In W. Dubitzky and F. Azuaje, editors, *Artificial Intelligence and Systems Biology*, Springer Series on Computational Biology. Springer.

Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. Mbt: A memory-based part of speech tagger-generator. In E. Ejerhed and I. Dagan, editors, *Fourth Workshop on Very Large Corpora*, pages 14–27.

J. N. Darroch and D. Ratcliff. 1972. Generalized iterative scaling for log-linear models. *Annals of Mathematicl Statistics*, 43:1470–1480.

J. Kazama, T. Makino, Y. Ohta, and J. Tsujii. 2002. Tuning support vector machines for biomedical named entity recognition. In *ACL-02 Workshop on Natural Language Processing in Biomedical Applications*.

Jin-Dong Kim, Tomoko Ohta, Yuka Teteisi, and Jun'ichi Tsujii. 2003. Genia corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl. 1).

Rob Koeling. 2000. Chunking with maximum entropy models. In *CoNLL-2000*.

D. Maglott. 2002. Locuslink: a directory of genes. In *NCBI Handbook*, pages 19–1 to 19–16.

NLM. 2003. Mesh: Medical subject headings.

S. Pakhomov. 2002. Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical text. In *the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*.

A. Ratnaparkhi. 1997. A simple introduction to maximum entropy models for natural language processing. Technical Report Techical Report 97-08, Institute for Research in Cognitive Science University of Pennsylvania.

Kazuhiro Seki and Javed Mostafa. 2003. An approach to protein name extraction using heuristics and a dictionary. In *ASIST 2003*.

Hagit Shatkay and Ronen Feldman. 2003. Mining the biomedical literature in the genomic era: an overview. *Journal of Computational Biology*, 10(6):821–855.

Yu Song, Eunju Kim, Gary Geunbae Lee, and Byoung-kee Yi. 2004. Posbiotm-ner in the shared task of bionlp/nlpba 2004. In *the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, Geneva, Switzerland.

Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2004. Improving the performance of dictionary-based approaches in protein name recognition. *Journal of Biomedical Informatics*, 37(6):461–470.

Shaojun Zhao. 2004. Named entity recognition in biomedical texts using an hmm model. In *COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)*.