

A Maximum Entropy Approach to Biomedical Named Entity Recognition

Yi-Feng Lin, Tzong-Han Tsai, Wen-Chi Chou, Kuen-Pin Wu, Ting-Yi Sung and Wen-Lian Hsu
Institute of Information Science, Academia Sinica
128, Section 2, Academy Road, Taipei, Taiwan
{lego, thtsai, jacky957, kpw, tsung, hsu}@iis.sinica.edu.tw

ABSTRACT

Machine learning approaches are frequently used to solve name entity (NE) recognition (NER). In this paper we propose a hybrid method that uses maximum entropy (ME) as the underlying machine learning method incorporated with dictionary-based and rule-based methods for post-processing. Simply using ME for NER, inaccurate boundary detection of NEs and misclassification may occur. Some NEs are partially recognized by ME. In the post-processing stage, we use dictionary-based and rule-based methods to extend boundary of partially recognized NEs and to adjust classification. We use GENIA corpus 3.01 to conduct 10-fold cross-verification experiments. To evaluate the performance, we consider the longest NE annotations. We evaluate our approach using standard precision (P), recall (R), and F-score, where F-score is defined as $2PR/(P+R)$. The precision, recall and F-score ([P, R, F]) of our ME module for overall 23 categories is [0.512, 0.538, 0.525], and after the post-processing the performance becomes [0.729, 0.711, 0.72] for [P, R, F]. For protein, DNA and RNA classes, our method achieves [P, R, F] of [0.77, 0.80, 0.785], [0.653, 0.748, 0.7], and [0.716, 0.788, 0.752], respectively. The post-processing stage significantly improves the performance of our ME-based NER module.

1. INTRODUCTION

The amount of biomedical literature available on the Web is rapidly increasing. There is a pressing need for biomedical information extraction. To extract useful information from natural language text, we must first recognize biomedical named entities in the text. In fact, named entity (NE) recognition (NER) is a fundamental research topic in natural language processing (NLP), which involves *entity identification* and *classification*.

Unlike NER in the newswire domain, NER in the biomedical domain remains a perplexing challenge. Biomedical NEs in general do not follow any nomenclature, and can be comprised of long compound words or short abbreviations. Some even contain various symbols or spelling variations. In summary, difficulties of NER in the biomedical domain are as follows:

- (1) Unknown word identification:
Unknown words can be acronyms, abbreviations, or words containing hyphens, digits, letters, and Greek

letters. Examples of NEs with unknown words include: *alpha B1*, *GM-CSF*, *Adenylyl cyclase 76E*, and *4'-mycarosyl isovaleryl-CoA transferase*.

- (2) Named entity boundary identification:
The boundary of an NE can be a regular English word, unknown word, Roman numeral, or digit. For example, *MHC Class II*, *latent membrane protein 1*, *NF-kappaB consensus site*, *cyclin-like UDG gene product* all have different types of boundaries. Additionally, nested NEs (an NE embedded in another NE, referred to as *cascaded* NEs by Shen et al. [9]) further complicate this problem. Consider the named entity *kappa 3 binding factor*. Its annotation $\langle \text{PROTEIN} \rangle \langle \text{DNA} \rangle \text{kappa 3} \langle / \text{DNA} \rangle \text{ binding factor} \langle / \text{PROTEIN} \rangle$ has two right boundaries at *3* and *factor*, which correspond to the embedded NE in the DNA category and the nested NE of the Protein category, respectively.
- (3) Named entity classification:
Once an NE is identified, it is then classified into a category such as protein, DNA, RNA, and so on. Ambiguity and inconsistency are often encountered at this stage. NEs with the same orthographical features may fall into different categories. For example, BRIX and SCOP both have the AllCaps feature, but the former is a gene and the latter is a protein. An NE may belong to multiple categories, e.g., *ELK1* is both a DNA and a protein. *p53* is another example. *p53* is a synonym for the gene *TP53* in HUGO nomenclature; but in the GENIA corpus, *p53* is also tagged as a protein. Such ambiguity is intrinsic. Another complication is that a nested NE of one category may contain an NE of another category. For instance, a protein name may contain the gene coding for this protein. For example, *A27L protein* is a protein name containing *A27L* which is the gene coding for this protein. We need to properly distinguish *A27L* from *A27L protein*.

To tackle these challenges, researchers use NLP techniques such as machine learning, dictionary-based methods and rule-based methods. Tsuruoka et al. [11] and Hanisch et al. [3] present dictionary-based approaches. Since new biomedical NEs keep being generated in literature, the machine learning approach prevails. After the release of GENIA corpus [6], machine learning approaches using GENIA corpus as training corpus are reported [10; 5; 13; 9; 14]. GENIA corpus provides a benchmark for evaluating different methods. The overall F-scores on 23 categories in GENIA corpus reported by these systems were at most 0.67.

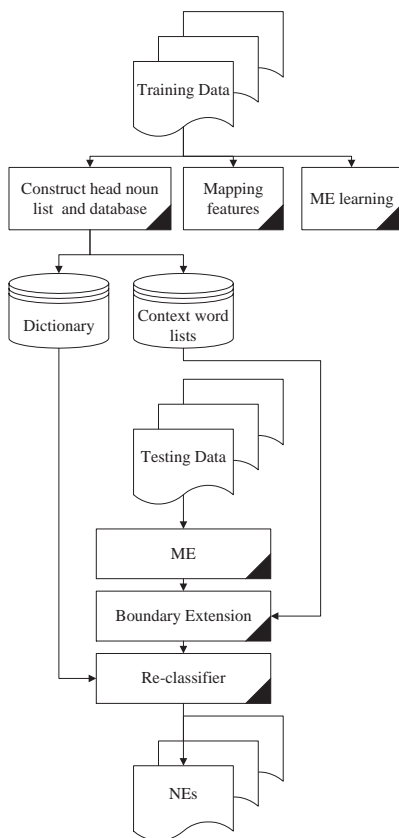


Figure 1: Method overview

The performance of machine learning approaches has big room for improvement. This fact can be attributed to small size of training corpora. Though GENIA corpus is the largest corpus for NER, it is rather small in comparison with the size of biomedical NEs. Various strategies are proposed to enhance the performance. In this paper, we use maximum entropy (ME) as our underlying machine learning method. Unexceptionally, the F-score of pure ME is 0.525 over the 23 categories of GENIA corpus. Our post-processing of ME output aims to resolve boundary detection problems and correct misclassification problems. Dictionary-based and rule-based methods are used, which significantly improves the performance.

2. ME-BASED BIOMEDICAL NER FRAMEWORK

Our recognition method consists of two stages: (1) ME-based recognition, (2) post-processing including boundary extension and reclassification. We first use ME for NER. Then we use a dictionary and rules to correct boundary identification errors by boundary extension. After boundary error correction is performed, the results are reclassified. Our method is depicted in Figure 1.

2.1 Maximum Entropy

We regard each word as a token. Since a named entity can have more than one 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, x_begin , $x_continue$, x_end , x_unique where x is a category. The first three tags denote respectively the beginning, the middle and the end of an NE in category x . The fourth tag denotes that a token itself is an NE of category x . In addition, we use the tag *unknown* to indicate that a token is not part of an NE. The NER problem can then be rephrased as the problem of assigning one of $4n + 1$ tags to each token, where n is the number of NE categories. In our ME module, there are 23 named entity categories and 93 tags. For example, one way to tag the phrase *IL-2 gene expression, CD28, and NF-kappa B* in a paper is “*othername_begin, othername_continue, othername_end, unknown, protein_unique, unknown, unknown, protein_begin, protein_end*.”

ME is a flexible statistical model which assigns an outcome for each token based on its *history* and *features*. Outcome space is comprised of the 93 tags for an ME formulation of NER. 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 characters of the current token are capitalized, it is likely to be part of a biomedical NE. Formally, we can represent this feature as follows:

$$f(h, o) = \begin{cases} 1 & \text{if } Current-Token-AllCaps(h) = \text{true} \\ & \text{and } o = \text{protein_begin}; \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Here, $Current-Token-AllCaps(h)$ is a binary function that returns the value *true* if all characters 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 α_i . From [1], we can compute the conditional probability as:

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

The probability is given by multiplying the weights of active features (i.e., those $f_i(h, o) = 1$). The weight α_i is estimated by a procedure called Generalized Iterative Scaling. This method improves estimation of weights at each iteration. The ME estimation technique guarantees that, for every feature f_i , the expected value of α_i equals the empirical expectation of α_i in the training corpus.

As noted in Borthwick [2], ME allows users to focus on finding features that characterizes the problem while leaving feature weight assignment to the ME estimation routine.

2.2 Decoding

After having trained an ME model and assigned the proper weights α_i to 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 α_i of the active features according to Equation 2. Finally, a Viterbi search is run

Table 1: Orthographical features

Orthographical features	Example	Orthographical features	Example
AllCaps	EBNA, NFAT, LMP	AlphaDigit	p50, p65
AlphaDigitAlpha	IL23R, E1A	ATGCSequence	CCGCCC, ATGAT
CapLowAlpha	Src, Ras, Epo	CapMixAlpha	NFkappaB, EpoR
CapsAndDigits	IL2, STAT4, SH2	DigitAlpha	2xNFkappaB, 2A
DigitAlphaDigit	32Dc13, 2D3	DigitCommaDigit	1,25
Digits	1, 2, 3, 1.1	Greek Letter	alpha, beta
Hyphen	-	LowMixAlpha	mRNA, mAb
Roman Numeral	I, II, III	SingleCap	A-Z
Stop word	at, in	Other	“, ”, “.”, “(”, “)”

Table 2: Head nouns

	Head nouns
Unigram	factor, protein, receptor, alpha, NF-kappaB, IL-2, cytokine, AP-1, kinase, IL-4, transcription, domain, complex, TNF-alpha, IFN-gamma, Nuclear, p50, p65, beta, NFAT, CD28, TNF, PKC, calcineurin, molecules, GM-CSF, GATA-1, IL-12, subunit, cell, STAT3, family, antibody, TCR, CIITA, chain, tumor, gamma, factor-alpha, expression, interleukin, IkappaBalpha
Bigram	NF-kappa B, transcription factor, I kappa, kappa B, nuclear factor, protein kinase, B alpha, kinase C, tumor necrosis, T cell, glucocorticoid receptor, colony-stimulating factor, binding protein, factor alpha, necrosis factor-alpha, adhesion molecule, monoclonal antibody, necrosis factor, T lymphocyte, cytoplasmic domain, gene product, binding domain

Table 3: Morphological features

~ase	~blast	~cin	~cyte
~kin	~lin	~lipid	~ma
~mide	~peptide	~phil	~rin
~rogen	~sor	~tin	~tor
~virus	~vitamin	~zole	anti~
cyto~	dehydr~	erytho~	hemo~

to find the highest probability path through the lattice of conditional probabilities that does not produce any invalid tag sequences. For instance, the sequence [*protein_begin*, *othername_continue*] is invalid because it does not contain an ending token and these two tokens are not in the same category. Further details on the Viterbi search can be found in [12].

2.3 Related Studies of NER Using ME

Raychaudhuri et al. [8] uses ME to assign Gene Ontology tags to genes appearing in biomedical literature. They report that ME outperforms the Naive Bayes method and the nearest-neighbor method. ME is also used for acronym and abbreviation normalization in medical texts. Pakhomov [7]

and Kazama et al. [4] report that SVM outperforms ME for biological NER. In Kazama et al. [4], the comparison is made using GENIA corpus version 1.0. The precision, recall and F-score ([P, R, F]) of the SVM-based system was [0.562, 0.528, 0.544] for overall categories and [0.492, 0.664, 0.565] for protein. The ME-based system reports [P, R, F] of [0.534, 0.530, 0.532] for overall performance and [0.491, 0.621, 0.548] for protein. Nevertheless, the authors also state that one advantage of the ME model is that it allows flexible feature selection. When new features, e.g., syntax features are added to ME, users do not need to reformulate the model like in the HMM model and ME estimation routine can automatically calculate new weight assignment. Thus we choose ME as the underlying machine learning model.

3. FEATURES

Feature selection is critical to the success of machine learning approaches. Orthographical features, head noun features, morphological features, and part-of-speech (POS) features are frequently used for token identification. We use POS features annotated in the GENIA corpus and report the remaining features below.

3.1 Orthographical Features

Table 1 lists some orthographical features used in our system. In our experience, AllCaps, CapMixAlpha, LowMixAlpha, SingleCap are more useful than others.

3.2 Head Nouns

The head noun is usually the major noun or noun phrase of an NE that describes its function or the property, e.g., *transcription factor* is the head noun for the NE *NF-kappa B transcription factor*. Compared with the other words in NE, head noun is a decisive factor for distinguishing the NE class. For instance, the classifications of <Protein> NF-kappa B transcription factor </Protein> and <DNA> IFN-gamma activation sequence </DNA> are determined by the head nouns *transcription factor* and *sequence*. In this work, only unigram and bigram head nouns are considered. We use training corpus to obtain 960 frequently used head nouns, and some are listed in Table 2.

3.3 Morphological Features

We consider morphological features of at least three characters in length. Some are listed in Table 3.

4. POST-PROCESSING AND RECLASSIFICATION FOR ERROR CORRECTION

Using ME, we find some NEs are partially recognized or mistakenly classified. In the post-processing stage, we aim to resolve boundary detection problems of partially recognized NEs by a boundary extension method. Afterwards, we use a re-classifier to resolve NE misclassification. Dictionary-based and rule-based methods are used for post-processing. The dictionary is constructed from the training corpus.

4.1 Boundary Extension

For those partially recognized NEs, we deal with two types of boundary detection problems that arise from (1) nested NEs and (2) brackets for name alias and slash for concatenated names.

Nested NEs may cause boundary detection problems. Consider the example “[E1A]_{/protein} gene” → “[E1A gene]_{/DNA}.” A straightforward right(R)-boundary extension rule is to extend the boundary if the NE is followed by NEs and/or head nouns. In the example “[GATA-1]_{/protein} activity” → “[GATA-1 activity]_{/othername},” the word *activity* is not a head noun. How do we determine whether the right boundary should be extended to *activity*? Consider another example: “type [I receptor]_{/protein}” → “[type I receptor]_{/protein}.” Should the left boundary extend to the word *type*? For the left(L)-boundary extension, we consider extension to include a modifier. What modifiers are allowed?

To resolve the abovementioned problems, we compile two lists of the *leftmost* (*L*) and the *rightmost* (*R*) context words of NEs in the training corpus. To construct these lists, we calculate the frequency of each context word candidate and determine a cutoff threshold to include candidates into the lists. The threshold is expected to affect the content of the lists and thus, the performance of post-processing. However, in our experiments, we have tried different threshold values and found that the threshold does not significantly affect the performance. We thus include all the candidates in the lists. Note that these context words may not be head nouns, but unigram head nouns surely belong to the lists.

In the previous example, *activity* is in the R-context word list and thus the right boundary can be extended to *activity*. We use context word lists to examine un-tagged tokens that are adjacent to ME-recognized NEs. If these tokens appear in the L- & R-context word lists, then they are concatenated with ME’s output. But simply using context word lists to determine boundary extension may fail in some cases. For example, *binding* is in the R-context word list. But *binding* can be tagged as a verb, an adjective or a noun. If *binding* is tagged as a verb, it is unlikely to be a part of an NE. Only few tokens tagged as a verb are included in NEs of GENIA corpus. We thus consider only adjective and noun as valid POS tags for the token in consideration. To further improve boundary extension accuracy, we examine the validity of the POS tag of the token. If this token appears in a context word list and its POS is valid, we will concatenate this token with the NE.

In summary, our boundary extension algorithm to resolve nested NEs goes as follows:

Step 1. Check R-boundary extension: Extend the boundary of an NE recognized by ME repeatedly if the NE is followed by another NE or a token in R-context word list with valid POS tag.

Step 2. Check L-boundary extension: Repeat similar procedure as in Step 1.

Step 3. Repeat Step 1 and 2 until no extension occurs.

Our algorithm can handle six patterns of nested NE construction presented in Zhou et al. [14].

The second type of boundary detection problem occurs when NEs contain brackets for name alias and slash for concatenated names which are not well handled by maximum entropy. For example, *basic helix-loop-helix (bHLH) motif* is an NE. Our ME module recognizes both *basic helix-loop-helix* and *bHLH* as protein. Since “(” and “)” are not valid context words, the previous algorithm cannot extend the boundary of ME’s output. Our solution is to detect whether *motif* is a valid context word. If yes, *basic helix-loop-helix (bHLH) motif* will be concatenated as one named entity.

After performing boundary extension for nested NEs, we use rule-based approach to extend boundary of the second type problem. The rules are given as follows:

1. NE := NE (+ NE) + R-context word;
2. NE := NE + / + NE (+ / + NE) + R-context word.

Inspecting the results generated by ME, we found that some human names were identified as NEs. A special module developed by our laboratory was introduced to filter these errors. This module is originally designed to extract authors, paper titles and journal names from citations.

4.2 Re-classifier

In boundary extension stage, we do not change the classification. Our re-classifier aims to resolve two types of classification errors. The first type is associated with boundary extension, for example, “[GATA-1]_{/protein} activity” → “[GATA-1 activity]_{/othername}.” The other type is intrinsic ambiguity caused by abbreviations. Orthographical features of AllCaps and CapsAndDigits are sometimes insufficient to distinguish between abbreviations of protein and DNA. For example, CD28 is a protein, and PS1 a DNA.

The re-classifier performs two steps. The first step is dictionary lookup. If the named entity is in the dictionary, we assign new class according to the dictionary. If the NE is not in the dictionary, we take the second step to adjust the classification according to R context word. We assign the class according to the context word.

5. EXPERIMENTS

5.1 GENIA Corpus

We use GENIA corpus version 3.01 to evaluate our system. The GENIA corpus contains 2,000 abstracts extracted from the Medline database and these abstracts are annotated with Penn Treebank part-of-speech tags. The annotation of the NEs is based on GENIA ontology. In our experiments, we use 23 distinct NE categories of GENIA corpus.

5.2 Experimental Results

We conduct 10-fold cross validation experiments and divide 2000 abstracts into 10 collections. Each collection contains not only abstracts but also paper titles. We evaluate our approach using standard precision (P), recall (R), and F-score, where F-score is defined as $2PR/(P+R)$. To evaluate our method, we consider the longest word annotation, since these NEs are useful for relation extraction.

Table 5: NE recognition performance

Config	Boundary Extension			Reclassify		NE Recognition P/R/F
	BE-1	BE-2	BE-3	RC-1	RC-2	
Baseline						0.512/0.538/0.525
Conf4	✓	✓	✓			0.645/0.634/0.639
Conf5	✓	✓	✓	✓		0.67/0.658/0.664
Conf6	✓	✓	✓		✓	0.707/0.695/0.701
Conf7	✓	✓	✓	✓	✓	0.727/0.715/0.721

Table 7: Partial matching performance

Task	NE Identification			NE Recognition		
	Precision	Recall	F-Score	Precision	Recall	F-Score
Exact Match	0.776	0.763	0.769	0.727	0.715	0.721
LD=1, ER > CR	0.802	0.788	0.795	0.74	0.728	0.734
LD=2, ER > CR	0.818	0.804	0.811	0.754	0.741	0.747
LD=1, CR > ER	0.804	0.791	0.797	0.744	0.731	0.737
LD=2, CR > ER	0.809	0.795	0.802	0.748	0.735	0.741
RD=1, ER > CR	0.805	0.79	0.797	0.733	0.72	0.726
RD=2, ER > CR	0.813	0.798	0.805	0.737	0.724	0.73
RD=1, CR > ER	0.808	0.791	0.799	0.735	0.721	0.728
RD=2, CR > ER	0.811	0.802	0.806	0.736	0.723	0.729

Table 4: NE identification performance

Config	Boundary Extension			NE Identification P/R/F
	BE-1	BE-2	BE-3	
Baseline				0.56/0.589/0.574
Conf1			✓	0.582/0.597/0.594
Conf2		✓		0.591/0.6/0.595
Conf3	✓			0.757/0.746/0.751
Conf4	✓	✓	✓	0.776/0.763/0.769

Table 6: System performance comparison (measured in F-Score)

Category	Overall	Protein	DNA	RNA
Our system	0.721	0.785	0.700	0.752
Zhou et al, 2004	0.666	0.758	0.633	0.612

In Table 4, we report the named entity identification (regardless of classification) performance. We use BE-1 to denote the nested boundary extension algorithm, BE-2 to denote the boundary extension for brackets and slashes, and BE-3 to denote the module to remove human names. From the figures, we can see that each method yields different degree of improvement in NE identification (boundary detection) performance. BE-1, which improves the NE identification performance by 0.177, is the most effective boundary extension method among the three methods.

In Table 5, we report the named entity recognition (including classification) performance. We use RC-1 to denote the re-classifier using dictionary lookup and RC-2 to denote the re-classifier using R context word. In Table 6, we show the performance of our system in overall 23 categories and in protein, DNA and RNA classes, and compare them with those reported in Zhou et al. [14]. We can see that our system has advantage over Zhou’s system in each main NE category and in overall performance. In Table 7, we report the partial matching results. We use LD = i (RD = i) to

mean that the recognized NE differs from the annotation by only i words at the left (right) boundary. ER and CR denote the length of the recognized NE (the experiment result) and the length of the annotation (the correct result).

6. CONCLUDING REMARKS

In this paper, we propose a hybrid method using maximum entropy and dictionary/rule-based methods. Currently, dictionary is only used in the post-processing stage. In the future, we shall improve our system by also using dictionary in the preprocessing stage. However, we need to overcome the difficulty arising from integration of dictionary preprocessing with ME. In the post-processing stage, we shall explore more extensively on determining rules for boundary extension and entity concatenation. In addition, we shall try to automatically generate good rules to enhance our system.

7. REFERENCES

- [1] A. Berger, S. A. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71, 1996.
- [2] A. Borthwick. *A maximum entropy approach to named entity recognition*. New York University, 1999.
- [3] D. Hanisch, J. Fluck, H. Mevissen, and R. Zimmer. Playing biology’s name game: identifying protein names in scientific text. In *PSB ’03*, 2003.
- [4] J. Kazama, T. Makino, Y. Ohta, and J. Tsujii. Tuning support vector machines for biomedical named entity recognition. In *ACL 2002*, 2002.
- [5] K.-J. Lee, Y.-S. Hwang, and H.-C. Rim. Two-phase biomedical ner recognition based on svms. In *ACL 2003*, 2003.

- [6] T. Ohta, Y. Tateisi, H. Mima, and J. Tsujii. Genia corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of the Human Language Technology Conference (HLT)*, 2002.
- [7] S. Pakhomov. Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical text. In *ACL 2002*, 2002.
- [8] S. Raychaudhuri, J. Chang, P. Sutphin, and R. Altman. Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. *Genome Research*, 12, 2002.
- [9] D. Shen, J. Zhang, G. Zhou, J. Su, and C. Tan. Effective adaptation of a hidden markov model-based named entity recognizer for biomedical domain. In *ACL 2003*, 2003.
- [10] K. Takeuchi and N. Collier. Bio-medical entity extraction using support vector machines. In *ACL 2003*, 2003.
- [11] Y. Tsuruoka and J. Tsujii. Boosting precision and recall of dictionary-based protein name recognition. In *ACL 2003*, 2003.
- [12] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, IT:260–269, 1967.
- [13] K. Yamamoto, T. Kudo, A. Konagaya, and Y. Matsumoto. Protein name tagging for biomedical annotation in text. In *ACL 2003*, 2003.
- [14] G. Zhou, J. Zhang, J. Su, D. Shen, and C. Tan. Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, 20:1178–1190, 2004.