BIOINFORMATICS

Vol. 20 no. 7 2004, pages 1178–1190
DOI: 10.1093/bioinformatics/bth060

Recognizing names in biomedical texts: a machine learning approach

GuoDong Zhou[1,], Jie Zhang[1,2], Jian Su[1], Dan Shen[1,2] and ChewLim Tan[2]*

[1]Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613 and [2]School of Computing, National University of Singapore, Singapore 119610

Received on August 5, 2003; revised on October 9, 2003; accepted on November 12, 2003
Advance Access publication February 10, 2004

ABSTRACT

Motivation: With an overwhelming amount of textual information in molecular biology and biomedicine, there is a need for effective and efficient literature mining and knowledge discovery that can help biologists to gather and make use of the knowledge encoded in text documents. In order to make organized and structured information available, automatically recognizing biomedical entity names becomes critical and is important for information retrieval, information extraction and automated knowledge acquisition.

Results: In this paper, we present a named entity recognition system in the biomedical domain, called PowerBioNE. In order to deal with the special phenomena of naming conventions in the biomedical domain, we propose various evidential features: (1) word formation pattern; (2) morphological pattern, such as prefix and suffix; (3) part-of-speech; (4) head noun trigger; (5) special verb trigger and (6) name alias feature. All the features are integrated effectively and efficiently through a hidden Markov model (HMM) and a HMM-based named entity recognizer. In addition, a *k*-Nearest Neighbor (*k*-NN) algorithm is proposed to resolve the data sparseness problem in our system. Finally, we present a pattern-based post-processing to automatically extract rules from the training data to deal with the cascaded entity name phenomenon. From our best knowledge, PowerBioNE is the first system which deals with the cascaded entity name phenomenon. Evaluation shows that our system achieves the *F*-measure of 66.6 and 62.2 on the 23 classes of GENIA V3.0 and V1.1, respectively. In particular, our system achieves the *F*-measure of 75.8 on the 'protein' class of GENIA V3.0. For comparison, our system outperforms the best published result by 7.8 on GENIA V1.1, without help of any dictionaries. It also shows that our HMM and the *k*-NN algorithm outperform other models, such as back-off HMM, linear interpolated HMM, support vector machines, C4.5, C4.5 rules and RIPPER, by effectively capturing the local context dependency and resolving the data sparseness problem. Moreover, evaluation on GENIA V3.0 shows that the post-processing for the cascaded entity name phenomenon improves the *F*-measure by 3.9. Finally, error analysis shows that about half of the errors are caused by the strict annotation scheme and the annotation inconsistency in the GENIA corpus. This suggests that our system achieves an acceptable *F*-measure of 83.6 on the 23 classes of GENIA V3.0 and in particular 86.2 on the 'protein' class, without help of any dictionaries. We think that a *F*-measure of 90 on the 23 classes of GENIA V3.0 and in particular 92 on the 'protein' class, can be achieved through refining of the annotation scheme in the GENIA corpus, such as flexible annotation scheme and annotation consistency, and inclusion of a reasonable biomedical dictionary.

Availability: A demo system is available at http://textmining.i2r.a-star.edu.sg/NLS/demo.htm. Technology license is available upon the bilateral agreement.

Contact: zhougd@i2r.a-star.edu.sg

INTRODUCTION

With an overwhelming amount of textual information in molecular biology and biomedicine, there is a need for effective and efficient literature mining and knowledge discovery that can help biologists to gather and make use of the knowledge encoded in text documents. In order to make organized and structured information available, automatically recognizing biomedical entity names becomes critical and is important for information retrieval, information extraction and automated knowledge acquisition.

Such technique, called named entity recognition, has been well developed in the Information Extraction literature (MUC-6, 1995; MUC-7, 1998). In MUC, the task of named entity recognition is to recognize the names of persons, locations, organizations, etc. in the newswire domain. In the biomedical domain, we care about entities like gene, protein, virus, etc. In recent years, many explorations have been performed to port existing named entity recognition systems into the biomedical domain (Proux *et al*., 1998; Nobata *et al*., 1999, 2000; Collier *et al*., 2000; Gaizauskas *et al*., 2000; Fukuda *et al*., 1998; Kazama *et al*., 2002; Takeuchi and Collier, 2002; Lee *et al*., 2003). However, few of them achieve satisfactory

[*] To whom correspondence should be addressed.

performance. Named entity recognition in the biomedical domain remains a challenging task. This may be due to the special naming conventions of entity names in the biomedical domain, compared with those in the newswire domain, as follows.

Descriptive naming convention: Many biomedical entity names are descriptive, e.g. 'normal thymic epithelial cells'. Such phenomenon highlights the difficulty for identifying the left boundaries of biomedical entity names. Moreover, some biomedical entity names are very long, e.g. '47 kDa sterol regulatory element binding factor'. In GENIA V3.0 (Ohta *et al.*, 2002), we find that 18.6% of biomedical entity names consist of at least four words, as shown in Figure 1.

Conjunction and disjunction: Two or more biomedical entity names may share one head noun by using conjunction or disjunction construction. For example, '91 and 84 kDa proteins' consists of two entity names: '91 kDa proteins' and '84 kDa proteins'. In GENIA V3.0, we find that 2.06% of biomedical entity names have such construction.

Non-standardized naming convention: One biomedical entity name may be found with various spelling forms, e.g. 'N-acetylcysteine', 'N-acetyl-cysteine', 'NAcetylCysteine', etc. We find that the use of capitalization or hyphen is more casual in the biomedical domain and most of biomedical entity names are not proper names. In GENIA V3.0, we find that 62.89% of words in biomedical entity names are in lowercase. Moreover, more and more biomedical entity names are created by authors and have not been standardized yet. This also results in the low coverage of existing biomedical dictionaries (Fukuda *et al.*, 1998; Nobata *et al.*, 1999). Therefore, traditional named entity recognition techniques relying on existing dictionaries may not perform well in the biomedical domain.

Abbreviation: Abbreviations are frequently used in the biomedical domain. Chang *et al.* (2002) shows that, in MEDLINE abstracts until the end of 2001, 42.8% of abstracts have at least one abbreviation and 23.7% of abstracts have two or more. It also shows that there is one new abbreviation in every 5–10 abstracts on average and the growth rate of new abbreviations is increasing. Moreover, many abbreviations in the biomedical domain are formed quite irregularly. For example, abbreviations can be from a subset of syllable boundaries, e.g. 'IL2' for 'Interleukin 2' or from contiguous characters, e.g. 'PAL' for 'palate'. Finally, abbreviations in the biomedical domain are highly ambiguous. For example, 'TCF' may refer to 'T cell Factor' or 'Tissue Culture Fluid' in different articles. Liu *et al.* (2002) shows that 81.2% of abbreviations are ambiguous and have an average of 16.6 senses in MEDLINE abstracts. Therefore, the classes of abbreviations are much dependent on the context, but not just decided by existing dictionaries.

Cascaded construction: One biomedical entity name may be embedded in another biomedical entity name, e.g. '<PROTEIN><DNA> kappa 3</DNA> binding factor </PROTEIN>'. We find that 16.57% of biomedical entity names have such cascaded construction in GENIA V3.0.

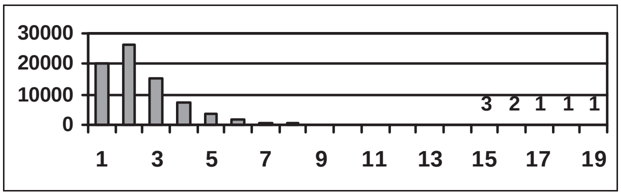

Fig. 1. Distribution of the number of words in biomedical entity names (GENIA V3.0).

On all accounts, we can say that the entity names in the biomedical domain are much more complex than those in the newswire domain. Therefore, it is necessary to explore more evidential features to cope with the special phenomena of naming conventions in the biomedical domain.

In this paper, we present a named entity recognition system in the biomedical domain, called PowerBioNE. In order to deal with the special phenomena of naming conventions in the biomedical domain, we propose various evidential features: (1) word formation pattern; (2) morphological pattern, such as prefix and suffix; (3) part-of-speech (POS); (4) head noun trigger; (5) special verb trigger and (6) name alias feature. All the features are integrated effectively and efficiently through a hidden Markov model (HMM) and a HMM-based named entity recognizer. In addition, a k-Nearest Neighbor (k-NN) algorithm is proposed to resolve the data sparseness problem in our system. Finally, we present a pattern-based post-processing to automatically extract rules from the training data to deal with the cascaded entity name phenomenon. From our best knowledge, PowerBioNE is the first system which deals with the cascaded entity name phenomenon. Evaluation on the GENIA corpus shows that our HMM and the k-NN algorithm outperform other models, such as back-off HMM, linear interpolated HMM, SVM, C4.5, C4.5 rules and RIPPER, by effectively capturing the local context dependency and resolving the data sparseness problem. It also shows that the post-processing for the cascaded entity name phenomenon can much improve the performance. Finally, it suggests that much better performance can be achieved through refining of the annotation scheme in the GENIA corpus, such as flexible annotation scheme and annotation consistency, and inclusion of a reasonable biomedical dictionary.

GENIA CORPUS

All of our experiments are done on GENIA corpus, which is the largest annotated corpus in the molecular biology domain available to public (Ohta *et al.*, 2002). In our experiments, three versions are used.

GENIA V1.1: This version contains 670 MEDLINE abstracts of 123K words. Since a few previous researches have been done on this version, we use it mainly for comparison of our work with others.

GENIA V2.1: This version incorporates POS to GENIA V1.1. It is mainly used to train our POS tagger and evaluate the usefulness of POS in biomedical named entity recognition.

GENIA V3.0: This version is a superset of GENIA V1.1 and contains 2000 MEDLINE abstracts of 360K words. We use this version to do the great scope of experiments and deep analysis.

The annotation of biomedical entities is based on the GENIA ontology (Ohta *et al*., 2002), which includes 23 distinct classes: multi-cell, mono-cell, virus, body part, tissue, cell type, cell component, organism, cell line, other artificial source, protein, peptide, amino acid monomer, DNA, RNA, poly nucleotide, nucleotide, lipid, carbohydrate, other organic compound, inorganic, atom and other.

FEATURES

In order to deal with the special phenomena of naming conventions in the biomedical domain, various evidential features are explored.

Word formation pattern (F_{WFP})

The purpose of this feature is to capture capitalization, digitalization and other word formation information. This feature has been widely used in both the newswire domain (Bikel *et al*., 1999; Chieu and Ng, 2002; Zhou and Su, 2002) and the biomedical domain (Nobata *et al*., 1999; Gaizauskas *et al*., 2000; Collier *et al*., 2000; Takeuchi and Collier, 2002; Kazama *et al*., 2002). In the newswire domain, this feature is very useful to detect the boundaries of entity names. For example, initial capitalization of a word often indicates an entity name. Although this feature is not so evidential in the biomedical domain as that in the newswire domain, it is still useful to distinguish between biomedical entity names and others. For example, 'H2A' may indicate a biomedical entity name. Table 1 shows a complete list of word formation patterns with the descending order of priority.

Table 1. F_{WFP}:word formation patterns

F_{WFP}	e.g.	F_{WFP}	e.g.
Comma	,	OneCap	T
Dot	.	AllCaps	CSF
Parentheses	( ) [ ]	CapLowAlpha	All
RomanDigit	II	CapMixAlpha	IgM
GreekLetter	Beta	LowMixAlpha	kDa
StopWord	in, at	AlphaDigitAlpha	H2A
ATCGsequence	ACAG	AlphaDigit	T4
OneDigit	5	DigitAlphaDigit	6C2
AllDigits	60	DigitAlpha	19D
DigitCommaDigit	1,25	Others	other
DigitDotDigit	0.5		

Morphological pattern (F_{MP})

Morphological information, such as prefix and suffix, is considered as an important cue for terminology identification and has been widely applied in the biomedical domain (Proux *et al*., 1998; Gaizauskas *et al*., 2000; Kazama *et al*., 2002; Lee *et al*., 2003). Gaizauskas *et al*. (2000) collects 100 common biochemical suffixes using their general English morphological analyzer, while Kazama *et al*. (2002) constructs prefix/suffix lists by sorting the 10 000 prefixes/suffixes from the training data.

Similar to Kazama *et al*. (2002), we use a statistical method to get the most frequent prefixes/suffixes from the training data as candidates. Then, each of these candidates is evaluated using

$$\mathrm{Weight}(\mathrm{Candidate}_t) = \frac{\#\mathrm{IN}_i - \#\mathrm{OUT}_i}{\#\mathrm{IN}_i + \#\mathrm{OUT}_i}, \tag{1}$$

where $\#\mathrm{IN}_i$ is the number of the i-th candidate occurring within entity names and $\#\mathrm{OUT}_i$ is the number of the i-th candidate occurring outside entity names.

Table 2. F_{MP}: examples of morphological patterns

F_{MP}	Prefix/suffix	Example
$\mathrm{MP}_{\mathrm{LIPID}}$	~lipid	Phospholipids
	~rogen	Estrogen
	~vitamin	Dihydroxyvitamin
$\mathrm{MP}_{\mathrm{VIRUS}}$	~virus	Cytomegalovirus

The rationale behind is that a particular prefix/suffix, which occurs most likely within entity names, may be thought as an evidence for distinguishing entity names. The candidates whose weights are above a certain threshold are chosen. Then, we count the frequency of each prefix/suffix in each entity class and group prefixes/suffixes with the similar distribution among the entity classes into one category. This can help resolve the data sparseness problem because prefixes/suffixes with the similar distribution have the similar contribution. Table 2 shows some of morphological patterns.

It shows that suffixes ~lipid, ~rogen and ~vitamin have been grouped into the category $\mathrm{MP}_{\mathrm{LIPID}}$ because they occurs frequently in the class 'Lipid' and much less frequently in other classes. Here, 0.7 is chosen as the threshold based on our experimentation. As a result, average 37 prefixes/suffixes are selected from the training data and further grouped into 23 categories.

Part-of-speech (F_{POS})

In the newswire domain, POS has been proven to be not useful, as POS may affect the use of more reliable capitalization information in determining the entity boundaries

Table 3. Comparison of POS tagger using different training data

Training set	Testing set	Precision
2500 WSJ articles	80 GENIA	85.1
590 GENIA abstracts	V2.1 abstracts	97.4

(Bikel *et al*., 1999; Zhou and Su, 2002). However, since most of the words in biomedical entity names are in lowercase, capitalization information in the biomedical domain is not as evidential as that in the newswire domain. Moreover, many biomedical entity names are descriptive and very long. Therefore, POS may provide useful evidence about the boundaries of biomedical entity names.

Lee *et al*. (2003) makes use of POS to identify the boundaries and find a reasonable performance improvement. However, they apply the existing POS information in the corpus and fail to discuss its contribution on the overall performance. Kazama *et al*. (2002) makes use of an existing POS tagger trained on PENN TreeBank (Marcus *et al*., 1993) and shows that POS can only slightly improve performance while Collier *et al*. (2000), Nobata *et al*. (2000) and Takeuchi and Collier (2002) do not incorporate POS in their systems. From the previous researches, it seems that POS does not help much in biomedical named entity recognition. There may have two reasons. One reason may be the lack of adaptation when porting an existing POS tagger from the newswire domain to the biomedical domain. To demonstrate the effect of adaptation on POS tagging, we adapt an HMM-based POS tagger to the biomedical domain using GENIA V2.1 (670 abstracts, 123K words). Table 3 shows the performance of the POS tagger on a test set of 80 GENIA V2.1 abstracts when trained on the remaining 590 GENIA V2.1 abstracts. For comparison, Table 3 also shows the performance of the POS tagger on the same test set when trained on PENN TreeBank (2500 *Wall Street Journal* articles, 1M words). It shows that the performance of POS tagging in the biomedical domain can be significantly improved via adaptation. In our following experimentations, we will make use of POS after adaptation except special indication. Another reason may be improper modeling of POS. One may ask: does POS really help biomedical named entity recognition? How much does the increase of POS accuracy affect the performance of biomedical named entity recognition? Which is more important, POS accuracy or proper modeling of POS? We will discuss these issues later.

Semantic triggers

Currently, two kinds of semantic triggers are explored in our system: head noun triggers (F_{HEAD}) and special verb trigger (F_{VERB}).

Table 4. Examples of auto-generated head nouns

Entity class	Unigram head nouns	Bigram head nouns
PROTEIN	Interleukin	Activator protein
	Interferon	Binding protein
	Kinase	Cell receptor
	Ligand	Gene product
DNA	DNA	X chromosome
	Breakpoint	Alpha promoter
	cDNA	Binding motif
	Chromosome	Promoter element

Table 5. Examples of special verb triggers

Activate	Express
Bind	Induce
Inhibit	Interact
Regulate	Stimulate

Head noun trigger (F_{HEAD})*:* The head noun, which is the major noun of a noun phrase, often describes the function or the property of the noun phrase, e.g. ‘B cells’ is the head noun for the biomedical entity name ‘activated human B cells’. Nobata *et al*. (1999) argues that head nouns in biomedical entity names can provide significant clues about the classes of the biomedical entity names.

In order to evaluate the effect of head noun triggers in the biomedical entity names, we automatically extract unigram and bigram head nouns from the training data, and rank them by frequency. For each entity class, we select 60% of top ranked head nouns as head noun triggers. Table 4 shows some of the examples. In future work, head nouns may also be extracted from public resources to further enhance their usability.

Special verb trigger (F_{VERB})*:* Besides collecting head noun triggers from biomedical entity names themselves, we also extract other semantic triggers from their local context. Recently, some frequently occurring verbs in MEDLINE have been proven useful for extracting the interaction between biomedical entity names (Thomas *et al*., 2000; Sekimizu *et al*., 1998). For example, the verb ‘bind’ is often used to indicate the protein–protein interaction. For named entity recognition in the biomedical domain, it is straightforward to think that particular verbs may also provide the evidence on the boundaries and the classes of biomedical entity names. Table 5 shows eight of them, which are explored in our initial experiments.

Name alias feature (F_{ALIAS})

The intuition behind this feature is the name alias phenomenon that relevant entities will be referred to in many ways throughout a given text and thus success of named entity recognition is conditional on success at determining when one noun phrase refers to the very same entity as another noun phrase.

During decoding, the entity names already recognized from the previous sentences of the document are stored in a list. When the system encounters an entity name candidate (e.g. a word with a special word formation pattern), a name alias algorithm is invoked to first dynamically determine whether the entity name candidate might be alias for a previously recognized name in the recognized list. The name alias feature F_{ALIAS} is represented as ENTITYnLm (L indicates the locality of the name alias phenomenon). Here, ENTITY indicates the class of the recognized entity name and n indicates the number of the words in the recognized entity name, while m indicates the number of the words in the recognized entity name from which the name alias candidate is formed. For example, when the decoding process encounters the word 'TCF', the word 'TCF' is proposed as an entity name candidate and the name alias algorithm is invoked to check if the word 'TCF' is an alias of a recognized named entity. If 'T cell Factor' is a 'Protein' name recognized earlier in the document, the word 'TCF' is determined as an alias of 'T cell Factor' with the name alias feature Protein3L3 by taking the three initial letters of the three-word 'protein' name 'T cell Factor'.

While the above method is useful to detect the inter-sentential name alias phenomenon, it is unable to identify the inner-sentential name alias phenomenon: the inner-sentential abbreviation. Such abbreviations widely occur in the biomedical domain.

In our system, we present an effective and efficient algorithm to recognize the inner-sentential abbreviations more accurately by mapping them to their full expanded forms. In GENIA corpus, we observe that the expanded form and its abbreviation often occur together via parentheses. Generally, there are two patterns: 'expanded form (abbreviation)' and 'abbreviation (expanded form)'. We also observe that the first pattern dominates except the case that the expression inside the parentheses includes at least two words, since an abbreviation normally includes only one word.

Our algorithm is based on the fact that it is much harder to classify an abbreviation than its expanded form. Generally, the expanded form is more evidential than its abbreviation to determine its class. The algorithm works as follows: when an abbreviation with parentheses is detected in a sentence, we remove the abbreviation and the parentheses from the sentence. After applying the HMM-based named entity recognizer to the sentence, we restore the abbreviation with parentheses to its original position in the sentence. Then, the abbreviation is classified as the same class of the expanded form, if the expanded form is recognized as an entity name. Finally, the expanded form and its abbreviation are stored in the recognized list of biomedical entity names from the document to help the resolution of forthcoming occurrences of the same abbreviation in the document.

METHODS

HMM-based biomedical named entity recognition

Given above various features, the key problem is how to effectively and efficiently integrate them together and find the optimal resolution to biomedical named entity recognition. Here, we use the HMM and the HMM-based named entity recognizer as described in Zhou and Su (2002). A HMM is a model where a sequence of outputs is generated in addition to the Markov state sequence. It is a latent variable model in the sense that only the output sequence is observed while the state sequence remains 'hidden'.

Given an output sequence $O_1^n = o_1 o_2 \cdots o_n$, the purpose of a HMM is to find the most likely tag (state) sequence $S_1^n = s_1 s_2 \cdots s_n$ that maximizes $P(S_1^n|O_1^n)$. Here, $o_i = \langle f_i, w_i \rangle$, where w_i is the word and $f_i = \langle F_{\mathrm{WFP}}^i, F_{\mathrm{MP}}^i, F_{\mathrm{POS}}^i, F_{\mathrm{HEAD}}^i, F_{\mathrm{VERB}}^i, F_{\mathrm{ALIAS}}^i \rangle$ is the feature set of the word w_i, and s_i = BOUNDARY$_i$_ENTITY$_i$_FEATURE$_i$, where BOUNDARY$_i$ denotes the position of the current word in the entity; ENTITY$_i$ indicates the class of the entity; and FEATURE$_i$ is the feature set.

By rewriting $\log P(S_1^n|O_1^n)$, we have:

$$\log P(S_1^n|O_1^n) = \log P(S_1^n) + \log \frac{P(S_1^n, O_1^n)}{P(S_1^n) \cdot P(O_1^n)}. \tag{2}$$

The second term in Equation (2) is the mutual information between S_1^n and O_1^n. In order to simplify the computation of this term, we assume mutual information independence:

$$MI(S_1^n, O_1^n) = \sum_{i=1}^{n} MI(s_i, O_1^n) \quad \text{or}$$

$$\log \frac{P(S_1^n, O_1^n)}{P(S_1^n) \cdot P(O_1^n)} = \sum_{i-1}^{n} \log \frac{P(s_i, O_1^n)}{P(s_i) \cdot P(O_1^n)}. \tag{3}$$

That is, an individual tag is only dependent on the output sequence O_1^n and independent on other tags in the tag sequence S_1^n. This assumption is reasonable because the dependence among the tags in the tag sequence S_1^n has already been captured by the first term in Equation (2). Applying the assumption (3) to Equation (2), we have:

$$\log P(S_1^n|O_1^n) = \log P(S_1^n) - \sum_{i=1}^{n} \log P(s_i) + \sum_{i=1}^{n} \log P(s_i|O_1^n). \tag{4}$$

From Equation (4), we can see that:

- The first term can be computed by applying chain rules. In ngram modeling (Chen and Goodman, 1996;

Jelinek, 1989; Katz, 1987), each tag is assumed to be probabilistically dependent on the $N-1$ previous tags.

- The second term is the summation of log probabilities of all the individual tags.
- The third term corresponds to the 'lexical' component (dictionary) of the tagger.

The idea behind our model is that we try to assign each output an appropriate tag, which contains boundary and class information. For example, 'TCF 1 binds stronger than NF kB to TCEd DNA'. The tag assigned to token 'TCF' should indicate that it is at the beginning of an entity name and it belongs to the 'Protein' class; and the tag assigned to token 'binds' should indicate that it does not belong to an entity name. Here, the Viterbi algorithm (Viterbi, 1967) is implemented to find the most likely tag sequence.

The problem with the above HMM lies in the data sparseness problem raised by $P(s_i|O_1^n)$ in the third term of Equation (4). Ideally, we would have sufficient training data for every event whose conditional probability we wish to calculate. Unfortunately, there is rarely enough training data to compute accurate probabilities when decoding on new data. Generally, two smoothing approaches (Chen and Goodman, 1996) are applied to resolve this problem: linear interpolation (Jelinek, 1989; Collier *et al.*, 2000) and back-off (Katz, 1987; Bikel *et al.*, 1999). However, these two approaches only work well when the number of different information sources is limited. When a few features and/or a long context are considered, the number of different information sources is exponential. In order to resolve the data sparseness problem in our system, we propose a k-NN algorithm using a constraint relaxation principle.

k-NN algorithm for computing $P(s_i|O_1^n)$

As stated above, the main challenge for the above HMM is how to reliably estimate $P(s_i|O_1^n)$. For efficiency, we can always assume $P(s_i|O_1^n) \approx P(s_i|E_i)$, where the pattern entry $E_i = o_{i-N}\cdots o_i \cdots o_{i+N}$. That is, we only consider the output context dependence in a window of $2N+1$ outputs (e.g. we only consider the current output, the previous output and the next output when $N=1$). For convenience, we denote $P(\bullet|E_i)$ as the conditional state probability distribution of the 'hidden' states given E_i and $P(s_i|E_i)$ as the conditional state probability of S_i given E_i.

The k-NN algorithm estimates $P(\bullet|E_i)$ by first finding the K-nearest neighbors of frequently occurring pattern entries $E_i^1, E_i^2 \cdots E_i^K$ to the initial pattern entry E_i and then aggregating them to make a proper estimation of $P(\bullet|E_i)$. Here, the conditional state probability distribution is estimated instead of the classification in a traditional k-NN classifier. To do so, all the frequently occurring pattern entries are extracted from the training corpus and stored in a dictionary *FrequentEntryDictionary*. In order to limit the dictionary size and keep efficiency, we can also constrain a valid set of pattern entry forms *ValidEntryForm* to consider only the most informative information sources.

Here, a constraint relaxation principle is used in the k-NN algorithm to find the K-nearest neighbors $E_i^1, E_i^2, \ldots, E_i^K$ to the initial pattern entry E_i. Considering the possible large number of features in a pattern entry as the constraints, there may have a large number of ways in which the constraints could be relaxed. Therefore, three restrictions are applied to keep the relaxation process tractable and manageable:

- Relaxation is done through iteratively dropping a constraint from the pattern entry.
- The pattern entry after relaxation should also have a valid form as defined in *ValidEntryForm*.
- Only top N (e.g. 5) pattern entries are kept during relaxation according to their likelihoods. Here, *likelihood*(E, E_i) for the pattern entry E is determined by its similarity with the initial pattern entry E_i.

The k-NN algorithm finds the K-nearest neighbors by iteratively relaxing a constraint in the initial pattern entry E_i until a set *kNN*(E_i) of at least K-nearest neighbors is found. At every iteration, we have a set of entries *InputEntrySet* as the input and return a set of entries *OutputEntrySet* as the output. The entries in *InputEntrySet* and *OutputEntrySet* are ranked according to their likelihoods. Initially, *InputEntrySet* is set as $\{\langle E_i,\ \mathit{likelihood}(E_i, E_i)\rangle\}$. Then, *OutputEntrySet* can be generated by relaxing any constraint (validated by *ValidEntryForm*) in every pattern entry of *InputEntrySet* (the pattern entry is deleted from *InputEntrySet* after relaxation). The frequently occurring pattern entries (validated by *FrequentEntryDictionary*) in *OutputEntrySet* are added to *kNN*(E_i) while the remaining are fed back to *InputEntrySet* for the next iteration. The relaxation process continues until *kNN*(E_i) contains at least K frequently occurring pattern entries. In order to remain efficient, only the top N (e.g. 5) pattern entries are kept in *InputEntrySet* and *OutputEntrySet* according to their likelihoods. Finally, the K-nearest neighbors $E_i^1, E_i^2, \ldots, E_i^K$ are extracted from *kNN*(E_i) according to their likelihoods.

After the K-nearest neighbors have been found, they are linearly interpolated according to their likelihoods:

$$P(\bullet|E_i) = \frac{\sum_{k=1}^{K} \mathit{likelihood}(E_i^k, E_i) \cdot P(\bullet|E_i^k)}{\sum_{k=1}^{K} \mathit{likelihood}(E_i^k, E_i)}.$$

In summary, the k-NN algorithm works as follows:

k-NN algorithm for computing $P(\bullet|E_i)$

Assume an initial pattern entry $E_i = o_{i-2}o_{i-1}o_io_{i+1}o_{i+2}$.
Assume a valid set of pattern entry forms *ValidEntryForm*.
Assume *FrequentEntryDictionary* is the dictionary which stores all the frequently occurring pattern entries.
Assume *likelihood*$(E,\ E_i)$ the likelihood of a pattern entry E as one of the K nearest neighbors to the initial pattern entry E_i.

```
Assume $kNN(E_i)$ stores the nearest neighbors to the initial
  pattern entry $E_i$.
BEGIN-of-algorithm
Compute $likelihood(E_i, E_i) = 1$
Initialize $InputEntrySet = \{<E_i, likelihood(E_i,E_i)>\}$,
  $OutputEntrySet = \{\}$ and $kNN(E_i) = \{\}$
LOOP
  DO for every entry $<E_j, likelihood(E_j,E_i)> \in InputEntrySet$
    DO for every constraint in $E_j$
      Assume $E'_j$ the pattern entry after relaxation
        of the constraint in $E_j$
      IF $E'_j$ conforms to ValidEntryForm THEN
      BEGIN
        Compute $likelihood(E'_j, E_i)$
        Add $OutputEntrySet = OutputEntrySet \cup \{<E'_j,likelihood(E'_j,E_i)>\}$
      END IF-THEN
    END DO
  END DO
  Add $kNN(E_i) = kNN(E_i) \cup \{<E, likelihood(E,E_i)>$
    $|<E, likelihood(E, E_i)> \in OutputEntrySet$ &
  $E \in FrequentEntryDictionary\}$
  Adjust $OutputEntrySet = OutputEntrySet$
   $-\{<E, likelihood(E, E_i)>|<E, likelihood(E, E_i)>$
     $\in OutputEntrySet$ & $E \in FrequentEntryDictionary\}$
  IF $kNN(E_i)$ contains at least K frequently occurring
    pattern entries
  THEN Extract the top K nearest neighbors from
    $kNN(E_i)$ according to their likelihoods and exit loop
  ELSE Set $InputEntrySet = OutputEntrySet$ and
    $OutputEntrySet = \{\}$
END LOOP
Compute $P(\bullet|E_i)$ by aggregating the K nearest neighbors
RETURN $P(\bullet|E_i)$
END-of-algorithm
```

Post-processing: cascaded entity name resolution

As shown in Table 1, 16.57% of entities in GENIA V3.0 have cascaded constructions, e.g.

<RNA><DNA>CIITA</DNA>mRNA</RNA>.

Therefore, it is important to resolve such phenomenon. Here, a pattern-based post-processing is proposed to resolve the cascaded entity names while the above HMM-based named entity recognition is applied to recognize embedded entity names and non-cascaded entity names.

In GENIA corpus, we find that there are six useful patterns of cascaded entity name constructions:

- <ENTITY> := <ENTITY> + head noun, e.g. <PROTEIN> binding motif→<DNA>
- <ENTITY> := <ENTITY> + <ENTITY>, e.g. <LIPID> <PROTEIN>→<PROTEIN>
- <ENTITY> := modifier + <ENTITY>, e.g. anti <Protein>→<Protein>
- <ENTITY> := <ENTITY> + word + <ENTITY>, e.g. <VIRUS> infected <MULTICELL>→<MULTICELL>
- <ENTITY> := modifier + <ENTITY> + head noun
- <ENTITY> := <ENTITY> + <ENTITY> + head noun

In our experiments, all the rules of above six patterns are extracted from the cascaded entity names in the training data to deal with the cascaded entity name phenomenon.

EXPERIMENTS AND EVALUATION

We evaluate our PowerBioNE system on GENIA V3.0 and GENIA V1.1 using precision/recall/F-measure (van Rijsbergen, 1979). For GENIA V1.1, we select 80 abstracts as the held-out test data and the remaining 590 abstracts as the training data (same as Kazama *et al*., 2002). For GENIA V3.0, we select 200 abstracts as the held-out test data and the remaining 1800 abstract as the training data. All the experimentations are done 10 times and the evaluations are averaged over the held-out test data. As a result, average 63 rules are extracted from the cascaded entity names in the training data of GENIA V1.1 for cascaded entity name resolution while average 102 rules are extracted from the cascaded entity names in the training data of GENIA V3.0.

Table 6 shows the best performance of our system on GENIA V3.0 and GENIA V1.1, and the comparison with that of the best reported system on GENIA V1.1 (Kazama *et al.*, 2002). It shows that our system achieves the F-measure of 62.2 on GENIA V1.1 and the F-measure of 66.6 on GENIA V3.0, respectively, without help of any dictionaries. It also shows that our system outperforms Kazama *et al.* (2002) by 7.8 in F-measure on GENIA V1.1. This may be due to the various evidential features, the effective HMM and k-NN algorithm and the post-processing for the cascaded entity name phenomenon in our system.

One important question is about the performance of different entity classes. Table 7 shows the number of entity name instances and the detailed performance for each biomedical entity class on GENIA V3.0. Of particular interest, our system achieves the F-measure of 75.8 on the class ‘Protein’. For comparison, our system achieves 5.6 higher F-measure without help of any dictionaries than Tsuruoka and Tsujii (2003) with help of a large dictionary. It shows that the performance varies a lot among different entity classes. One reason may be due to different difficulties in recognizing different entity classes. For example, the ‘Body Part’ class is among the easiest to recognize and has best performance due to the fact that the number of body parts is quite limited,

Table 6. Performance of our PowerBioNE system and comparison with that of Kazama *et al*. (2002)

Performance	*P*	*R*	*F*
Our system on GENIA V3.0	66.5	66.6	66.6
Our system on GENIA V1.1	63.1	61.2	62.2
Kazama *et al*. (2002) on GENIA V1.1	56.2	52.8	54.4

Table 7. Performance of different entity classes on GENIA V3.0

Entity class	Number of instances	*F*-measure
Body part	418	80.5
Cell type	6751	79.1
Lipid	1814	75.5
Multi-cell	1757	77.8
Protein	26 482	75.8
Cell component	690	69.7
Polynucleotide	271	69.5
Other organic compound	3754	67.5
Amino acid monomer	423	65.9
DNA	9576	63.3
Cell line	3992	61.4
RNA	871	61.2
Virus	1048	61.2
Other	20 879	60.5
Mono-cell	181	50.5
Tissue	663	48.8
Inorganic	214	40.5
Nucleotide	141	22.7
Peptide	420	17.2
Atom	171	15.3
Other artificial source	202	7.8
Carbohydrate	70	4.9
Organic	2	00.0
Overall	81 190	66.6

compared with the much worse performance of the ‘Peptide’ class due to the much large number of peptides and their variations. Another reason may be due to the different numbers of instances in different entity classes. Though GENIA V3.0 provides a good basis for named entity recognition in the biomedical domain and probably the best available, it has clear bias. It shows that GENIA V3.0 is much unbalanced for different entity classes. For example, the ‘Protein’ class occupies about 30% while the class ‘Organic’ only has two instances. Those entity classes with a much larger number of instances, such as ‘Protein’ and ‘Cell Type’, normally have much higher performance than those with much less instances, such as ‘Peptide’ and ‘Polynucleotide’. Table 7 also shows that, while GENIA V3.0 is of enough size for recognizing the major classes, such as ‘Protein’, ‘Cell Type’, ‘Cell Line’, ‘Lipid’, etc. it is of limited size and fails in recognizing the minor classes, such as ‘Organic’, ‘Carbohydrate’, ‘Atom’, ‘Peptide’, ‘Nucleotide’, ‘Inorganic’, etc. Finally, GENIA V3.0 does not differentiate the ‘Drug’ class, which is found very important in many real applications.

Table 8. Effects of different features and post-processing for cascaded entity name resolution on GENIA V3.0

F_{WFP}	F_{MP}	F_{POS}	F_{HEAD}	F_{VERB}	F_{ALIAS}	Cascaded entity name resolution	*P*	*R*	*F*
√							42.0	22.0	28.9
√	√						44.5	24.5	31.6
√	√	√					58.2	51.0	54.4
√	√	√	√				62.2	62.1	62.1
√	√	√	√	√			62.1	61.6	61.8
√	√	√	√		√		62.5	62.9	62.7
√	√	√	√			√	66.2	65.8	66.0
√	√	√	√		√	√	66.5	66.6	66.6

Another important question is about the effect of different features and the pattern-based post-processing for cascaded entity name resolution. Table 8 answers the question on GENIA V3.0:

- The contribution of the word formation pattern feature in the biomedical domain is very limited compared with that in the newswire domain. It achieves the *F*-measure of 28.9 on GENIA V3.0 compared with 77.6 on MUC-7 (Zhou and Su, 2002).
- The morphological pattern feature further increases the *F*-measure by 2.7.
- POS after adaptation is proven to be very useful in the biomedical domain. It increases the *F*-measure by 22.8. On contrary, POS is proven to be of little use in the newswire domain (Bikel *et al*., 1999; Zhou and Su, 2002). It is because that POS affects the use of more reliable capitalization information in determining the boundaries of entity names in the newswire domain, e.g. MUC-6 and MUC-7. However, since most of the words in biomedical entity names are in lowercase, capitalization information in the biomedical domain is not as evidential as that in the newswire domain. Moreover, many biomedical entity names are descriptive and very long. Therefore, POS provides useful evidence about the boundaries of biomedical entity names.
- Within our expectation, the head noun trigger feature is very useful. It significantly increases the *F*-measure by 7.7.
- Out of our expectation, the use of the special verb trigger feature decreases the recall rate while keeping the precision.
- The name alias feature only slightly improves the *F*-measure by 0.6. This may be due to the complexity of

Table 9. Effect of different number of special verb triggers on GENIA V3.0

Number of verb triggers	*P*	*R*	*F*
0	66.5	66.6	66.6
2	66.5	66.5	66.5
4	66.5	66.4	66.5
8	66.4	66.3	66.4
16	66.4	66.0	66.2
32	66.4	65.4	65.9
64	66.3	65.1	65.7
128	66.1	64.6	65.4

Table 10. Effect of different POS accuracies and different models on GENIA V3.0 (without post-processing for cascaded entity name resolution)

Method	Accuracy of POS tagging[a]					
	No POS	80.0	85.1	90.0	95.0	97.4
Our HMM	55.3	57.6	60.2	61.5	62.3	62.7
Back-off HMM (Bikel *et al.*, 1999)	54.8	55.4	57.9	59.1	59.8	60.1
Linear interpolated HMM (Collier *et al.*, 2000)	54.3	54.9	57.7	58.7	59.5	59.8
SVM (Lee *et al.*, 2003)[b]	53.7	54.2	55.2	56.1	56.6	56.8
C4.5 (Lee *et al.*, 2003)[b]	53.5	54.0	54.9	55.6	56.0	56.2
C4.5 rules (Lee *et al.*, 2003)[b]	53.9	54.6	55.7	56.4	56.8	57.1
RIPPER (Lee *et al.*, 2003)[b]	53.8	54.2	55.4	56.3	56.8	57.0
Naïve Bayes classifier (Tsuruoka and Tsujii, 2003)[c]	48.5	48.8	49.5	49.9	50.1	50.2

[a] The POS tagger with accuracy of 85.1 are trained on 2500 WSJ articles; the POS tagger with accuracy of 97.4 are adapted on GENIA V2.1; other POS taggers are trained on the subsets of 2500 WSJ articles.

[b] SVM, C4.5, C4.5 rules and RIPPER use the same two-phase recognition method as described in Lee *et al.* (2003) except the different models.

[c] The naïve Bayes classifier is used as a baseline.

the name alias phenomenon in the biomedical domain and the simple strategy applied in our system. Given its special characteristics, we find it very difficult to deal with the name alias phenomenon in the biomedical domain. Many of name aliases do not take any lexical, morphology or semantic evidence for a certain class. In the future work, we will explore more useful patterns to the name alias phenomenon.

- The pattern-based post-processing for cascaded entity name resolution is proven to be very useful. It improves the F-measure further by 3.9. From our best knowledge, our system is the first system which deals with the cascaded entity name resolution.

Two questions arise from Table 8. One question is about the effect of the special verb trigger feature. In the above experimentation, only eight special verb triggers (as shown in Table 5) are explored. One may ask whether more verb triggers can improve the performance. Table 5 answers the question. Here, all the verb triggers are gathered from inside the entity names and their local context (the preceding two words and the following two words). It shows that more verb triggers only decrease the performance more. We find that, although a verb trigger often strongly indicates a possible occurrence of an entity name in its local context, it is very difficult to determine its functionality, e.g.

- in '. . . the <PROTEIN>NF-kappa B</PROTEIN> **activated** in . . .', where 'activated' occurs after a 'Protein' entity name.
- in '. . . of <CELLTYPE>**activated** T-cells </CELLTYPE>) transcriptional activity . . .', where 'activated' acts as a modifier in a 'Cell Type' entity name.
- in '. . . blockade of the <PROTEIN>Ca(2+)-**activated** K+ channel</PROTEIN> is not associated . . .', where 'activated' occurs inside a 'Protein' entity name.
- in '<PROTEIN>Tax</PROTEIN> co-transfected with <DNA>reporter constructs</DNA> into <CELLLINE> Jurkat cells</CELLLINE> maximally **activated** <PROTEIN>HTLV-I-LTR-CAT</PROTEIN> and <PROTEIN>kappa B-fos-CAT</PROTEIN> and also **activated** <DNA>LT-293</DNA> to a lesser extent', where both 'activated' occur before 'protein'/'DNA' entity names and act as the predicates of the sentence.

Another question arisen from Table 8 is about the usefulness of POS. Our experimentation shows that POS is very useful in biomedical named entity recognition. This contradicts with previous researches (e.g. Kazama *et al*., 2002). There may be two reasons: different POS accuracies and different models. Table 10 shows the effect of different POS accuracies and different models on GENIA V3.0. For comparison, all the models use the same features described in this paper excluding the special verb trigger feature and the pattern-based post-processing for cascaded entity name resolution. In particular, the back-off HMM and the linear interpolated HMM further smooth the corresponding probabilities by orderly dropping the features F_{ALIAS}, F_{MP}, F_{HEAD}, F_{POS} and F_{WFP}, according to their contributions as shown in Table 8. It shows that inclusion of POS with different accuracies performs quite differently on these models. Inclusion of POS with accuracy of 80% only slightly improves the performance and there is a dramatic performance improvement for POS accuracies between 80 and 90%, while increasing the POS accuracy above 90% only slightly improves the performance. It suggests that stable and significant performance improvement can only be achieved for inclusion of POS with enough accuracy (e.g. 90% for GENIA V3.0). It also shows that our HMM benefits most from the inclusion of POS and different models have much

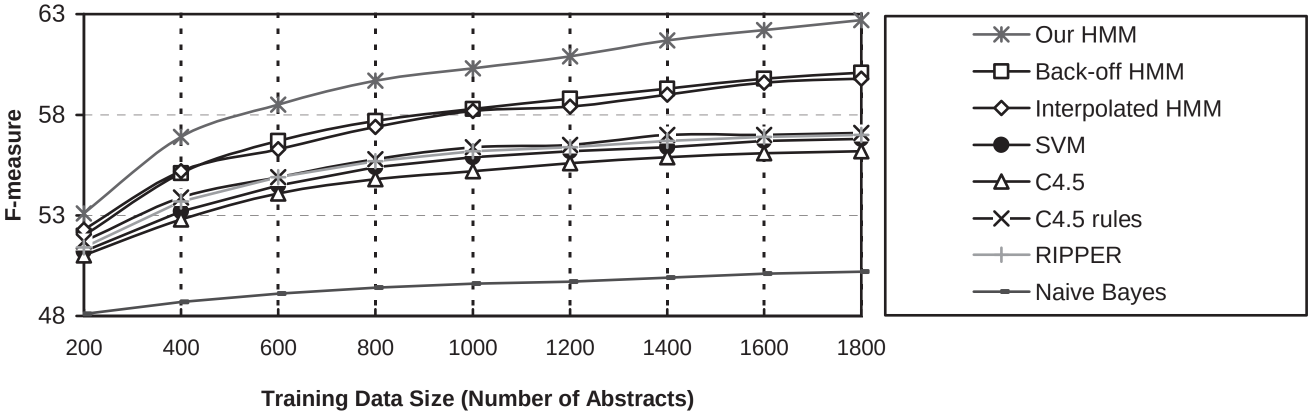

Fig. 2. Effect of different training data size on different models (without post-processing for cascaded entity name resolution).

different characteristics on the usefulness of POS although all the models benefit from inclusion of POS. For comparison of inclusion of POS after adaptation over inclusion of POS before adaptation, it shows that our HMM performs best with performance improvement of 2.5; the other two HMMs perform second with that of 2.1–2.2, the four feature vector-based models (SVM, C4.5, C4.5 rules and RIPPER) perform third with that of 1.3–1.6 and the baseline naïve Bayes classifier performs worst with that of 0.7. For comparison of inclusion of POS after adaptation over no inclusion of POS, it shows that our HMM perform best with performance improvement of 7.4, the other two HMMs perform second with that of 5.3–5.5, the four feature vector-based models (SVM, C4.5, C4.5 rules and RIPPER) perform third with that of 2.7–3.2 and the baseline naïve Bayes classifier performs worst with that of 1.7. It also shows that for all the models, about 40% of performance improvement is achieved by adaptation of POS.

Table 10 also shows that among these models, the performance of HMMs is higher than that of others:

- Our HMM performs best with the F-measure of 62.7.
- The back-off HMM and the linear interpolated HMM perform second with 2.6–2.9 behind.
- SVM, C4.5, C4.5 rules and RIPPER achieve the F-measure of 56.2–57.1 with 5.6–6.5 behind.
- The baseline naïve Bayes classifier performs worst with 12.5 behind.

The main reason may be due to modeling of local context dependence. HMMs have the better ability of capturing the locality of various phenomena, which indicate biomedical entity names in a text document. The feature vector-based classifiers, such as SVM, C4.5, C4.5 rules and RIPPER, cannot effectively capture the local context dependence by assuming the independence between the features while the baseline naïve Bayes classifier fails to capture local context dependence by assuming the conditional probability independence among the local context. It also shows that among the three HMMs, our HMM performs much better than the back-off HMM and the linear interpolated HMM. This is due to the different resolutions of the data sparseness problem in the three HMMs: our k-NN algorithm can dynamically smooth the probability estimation according to its local context and thus resolve the data sparseness problem much better than the linear interpolated method and the back-off method which smooth in a predefined way.

The final question is the effect of different training data sizes on different models. Figure 2 answers the question on GENIA V3.0. For comparison, all models are tested without post-processing for biomedical entity name resolution. It shows that all the models have a turning point at about 400–600 abstracts: as the training data size increases, all the models improves the performance much more before the turning point than after the turning point. It also indicates that all the three HMMs have the much potential of benefiting from further increasing the size of GENIA V3.0 while further increasing the size of GENIA V3.0 does not help much for the other models. This is due to that the HMMs are much more data-driven than the other models due to their modeling of local context dependency. Among them, our HMM is the most data-driven due to its dynamic modeling of local context dependence while the baseline naïve Bayes classifier is the least data-driven due to its assumption of conditional probability independence among the local context.

ERROR ANALYSIS

Our system achieves the F-measure of 66.6 on GENIA V3.0. In particular, our system achieves the F-measure of 75.8 on the ‘protein’ class. In order to further evaluate our system and explore possible improvement, we have implemented an error analysis. This is done by randomly choose 100 errors from our recognition results. During the error analysis, we find many errors are due to the strict annotation scheme and the annotation inconsistence in the GENIA corpus, and can be considered acceptable. Therefore, we will also examine the acceptable F-measure of our system, in particular, the acceptable F-measure on the 'protein' class.

All the 100 errors are classified into following sources:

- *Left boundary errors* (15): It includes the errors with correct class identification, correct right boundary detection and only wrong left boundary detection. We find that most of such errors come from the descriptive naming convention. We also find that 12 out of 15 errors are acceptable and ignorance of the descriptive words often does not make much difference for the entity names. In fact, it is even hard for biologists to decide whether the descriptive words should be a part of the entity names, such as 'normal', 'activated', etc. In particular, 3 of 15 errors belong to the 'protein' class. Among them, two errors are acceptable, e.g. 'classical <PROTEIN>1,25 (OH) 2D3 receptor</PROTEIN>'=>'<PROTEIN>classical 1,25 (OH) 2D3 receptor</PROTEIN>' (with format of 'annotation in the corpus => identification made by our system'), while the other one is unacceptable, e.g. '<PROTEIN>viral transcription factor</PROTEIN>=> viral <PROTEIN>transcription factor </PROTEIN>'.
- *Cascaded entity name errors* (17): It includes the errors caused by the cascaded entity name phenomenon. We find that most of such errors come from the annotation inconsistence in the GENIA corpus: in some cases, only the embedded entity names are annotated while in other cases, the embedded entity names are not annotated. Our system tends to annotate both the embedded entity names and the whole entity names. Among them, we find that 13 of 17 errors are acceptable. In particular, 2 of 17 errors belong to the 'protein' class. Among them, all the two errors are acceptable, e.g. '<DNA>NF kappa B binding site</DNA>' => '<DNA><PROTEIN>NF kappa B</PROTEIN> binding site</DNA>'.
- *Misclassification errors* (16): It includes the errors with wrong class identification, correct right boundary detection and correct left boundary detection. We find that this kind of errors mainly comes from the sense ambiguity of biomedical entity names and is very difficult to disambiguate. Among them, seven errors are related with the 'DNA' class and six errors are related with the 'Cell Line' and 'Cell Type' classes. Among them, we find that only 3 out of 16 errors are acceptable. In particular, there are five errors related to the 'protein' class. We also find that all the five errors are caused by misclassification of the 'DNA' class to the 'protein' class and all of them are unacceptable, e.g. '<DNA>type I IFN<DNA>' => '<PROTEIN>type I IFN</PROTEIN>'.
- *True negative* (29): It includes the errors by missing the identification of biomedical entity names. We find that 14 errors come from the 'other' class and 11 errors from the 'protein' class. We also find that GENIA corpus annotates some general noun phrases as biomedical entity names, e.g. 'protein' in 'the protein' and 'cofactor' in 'a cofactor'. We also find that 12 of 29 errors are acceptable. In particular, 11 of 29 errors related to the 'protein' class. Among them, four errors are acceptable, e.g. 'the <PROTEIN>protein</PROTEIN>' => 'the protein', while the other seven are unacceptable, e.g. '<PROTEIN>80 kDa</PROTEIN>' => '80 kDa'.
- *False positive* (18): It includes the errors by wrongly identifying biomedical entity names which are not annotated in the GENIA corpus. We find that 11 out of 18 errors come from the 'other' class. This suggests that the annotation of the 'other' class is much lack of consistency and most problematic in the GENIA corpus. We also find that 10 out of 18 errors are acceptable. In particular, 2 out of 18 errors are related to the 'protein' class and both of them are all acceptable, e.g. 'affinity sites' => '<PROTEIN>affinity sites</PROTEIN>'.
- *Miscellaneous* (11): It includes all the other errors, e.g. combination of the above errors and the errors caused by parentheses. We find that only 1 out of 11 errors is acceptable. We also find that, among them, two errors are related with the 'protein' class and both are unacceptable, e.g. '<PROTEIN>17 amino acid epitope</PROTEIN>' => '17 <RNA>amino acid epitope</RNA>'.

From above error analysis, we find that about 50% (51/100) of errors are acceptable and can be avoided by flexible annotation scheme (e.g. regarding the modifiers in the left boundaries) and consistent annotation (e.g. in the annotation of the 'other' class and the cascaded entity name phenomenon). In particular, about 40% (12/28) of errors are accepted on the 'protein' class. This means that the acceptable F-measure is about 83.6 on the 23 classes of GENIA V3.0. In particular, the acceptable F-measure on the 'protein' class is about 86.2. In addition, this performance is achieved without using any dictionaries. If we can have a reasonable dictionary with coverage of 50%, an acceptable F-measure of 90 on the 23 classes of GENIA V3.0, in particular 92 on the 'protein' class can be achieved.

RELATED WORK

Previous approaches in biomedical named entity recognition typically use domain specific heuristic rules and heavily rely on existing dictionaries. Such representative researches include Fukuda *et al*. (1998), Proux *et al*. (1998), and Gaizauskas *et al*. (2000). Fukuda *et al*. (1998) uses some heuristic rules to identify 'protein' names in SH3 protein domain. Evaluation on 30 annotated MEDLINE abstracts shows precision of 95.22% and recall of 97.40%. Proux *et al.* (1998) uses finite state technology to detect 'gene' names by using lexical and morphological knowledge. It achieves precision of 91.4% and recall of 94.4% on a small corpus of 1200 sentences from Flybase (The Flybase Consortium, 1998), and precision of about 70% on a larger corpus of 25 000 abstracts from MEDLINE. Gaizauskas *et al.* (2000) derives their system from an existing system in the newswire domain (MUC) and applies it in two projects: extraction of enzymes and metabolic pathways (EMPathIE) and extraction of protein structure (PASTA). Their system mainly uses lexicons, morphological cues, such as suffixes and hand-coded rules. Evaluation of biomedical named entity recognition on six full journal articles in EMPathIE task achieves precision of 86% and recall of 68% on 10 named entity classes, such as 'compound', 'element', 'enzyme', etc. while evaluation on 52 MEDLINE abstracts in PASTA task achieves precision of 94% and recall of 88% on 13 named entity classes, such as 'protein', 'species', 'residue', etc. Although these rule-based systems seem quite promising, they lack the ability of adaptation to new named entity classes in the biomedical domain. Once a new class is required to identify, a set of rules for this class has to be generated manually. In addition, the more classes, the more ambiguous and difficult to construct consistent rules. Finally, these systems heavily depend on well-developed dictionaries.

The current trend is to apply machine learning approaches in biomedical named entity recognition, largely due to the development of the GENIA corpus. The typical explorations include Nobata *et al.* (1999), Collier *et al.* (2000), Takeuchi and Collier (2002), Kazama *et al.* (2002), Lee *et al.* (2003) and Tsuruoka and Tsujii (2003). Nobata *et al.* (1999), Collier *et al.* (2000) and Takeuchi and Collier (2002) use a preliminary version of the GENIA corpus which contains 100 abstracts and identifies 10 named entity classes, such as 'protein', 'DNA', 'RNA', 'cell line', 'cell type', etc. Nobata *et al.* (1999) uses the decision tree and incorporates POS, character information and domain-specific word lists. The experiment shows that it achieves the F-measure of about 56. Collier *et al.* (2000) applies a linear interpolated HMM and incorporates surface word itself and character information. It achieves the F-measure of 72.8. Takeuchi and Collier (2002) applies SVM and incorporates surface word itself, character information and pervious word class tags (-3 to $+3$). It achieves the F-measure of 71.8. Kazama *et al.* (2002) also applies SVM and incorporates a rich feature set, including word feature, POS, prefix feature, suffix feature, previous class feature, word cache feature and HMM state feature. The experiment on GENIA V1.1 of 23 classes shows the F-measure of 54.4, compared with our 62.2 on the same GENIA V1.1. Tsuruoka and Tsujii (2003) applies a dictionary-based approach and a naïve Bayes classifier to filter out false positives. It only evaluates against the 'protein' class in GENIA V3.0, and receives the F-measure of 70.2 with the help of a large dictionary, compared with our 75.8 on the same class without help of any dictionaries. Lee *et al.* (2003) uses a two phase SVM-based recognition approach and incorporates word formation pattern and POS. The evaluation on GENIA V3.0 shows the F-measure of 66.5 over 22 classes except the 'other' class with help of an entity name dictionary, compared with our 68.7 over the same classes without help of any dictionaries.

Direct comparison of different biomedical named entity recognition systems is difficult because of different methods, entity classes, evaluation corpus and the use of dictionaries. In general, systems on specified evaluation corpus with help of dictionaries tend to perform much better than those on general ones without help of any dictionaries.

CONCLUSION

In this paper, we describe our exploration on porting an existing HMM-based named entity recognizer to the biomedical domain. Various lexical, morphological, syntactic, semantic and discourse features are incorporated to cope with the special phenomena in biomedical named entity recognition. In addition, a k-NN algorithm is proposed to effectively resolve the data sparseness problem. Finally, we present a pattern-based post-processing to deal with the cascaded entity name phenomenon. Evaluation shows that our HMM-based system and the k-NN algorithm can effectively integrate various features in biomedical named entity recognition. It also shows that our HMM and the k-NN algorithm outperform other models, such as back-off HMM, linear interpolated HMM, SVM, C4.5, C4.5 rules and RIPPER, by effectively capturing the local context dependency and resolving the data sparseness problem.

The main contribution of our work lies on the proposal and appropriate integration of various evidential features, including word formation pattern, morphological pattern, POS, head noun trigger, special verb trigger and name alias feature. The second contribution is the k-NN algorithm in the effective resolution of the data sparseness problem in our system. The final contribution is the pattern-based post-processing in dealing with the cascaded entity name phenomenon. From our best knowledge, our system is the first system which deals with the cascaded entity name phenomenon.

In the near future, we will further improve the performance by investigating more on conjunction and disjunction construction, the synonym phenomenon, the cascaded entity name phenomenon and inclusion of a reasonable biomedical dictionary. In the meanwhile, we will explore our system in real applications.

REFERENCES

Bikel,M.D., Schwartz,R. and Weischedel,M.R. (1999) An algorithm that learns what's in a name. *Mach. Learn.: Special Issue on Natural Language Learning*, **34**, 211–231.

Chang,J.T., Schutze,H. and Altman,R.B. (2002) Creating an online dictionary of abbreviations from MEDLINE. *J. Am. Med. Inform. Assoc.*, **9**, 612–620.

Chen,S.F. and Goodman,J. (1996) An empirical study of smoothing technniques for language modeling. *Proceedings of the 34th Annual Meeting of the Association of Computational Linguistics (ACL'1996)*. Santa Cruz, California, USA, pp. 310–318.

Chieu,H.L. and Ng,H.T. (2002) Named entity recognition: a maximum entropy approach using global information. *Proceedings of the 19th International Conference on Computational Linguistics (COLING'2002)*. Taipei, Taiwan, pp. 190–196.

Collier,N., Nobata,C. and Tsujii,J. (2000) Extracting the names of genes and gene products with a hidden Markov model. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*. Saarbrucken, Germany, 31 July– 4 August, 201–207.

Fukuda,K., Tsunoda,T., Tamura,A. and Takagi,T. (1998) Toward information extraction: identifying protein names from biological papers. In *Proceedings of the Pacific Symposium on Biocomputing'98 (PSB'98)*. Hawaii, January, pp. 707–718.

Gaizauskas,R., Demetriou,G. and Humphreys,K. (2000) Term recognition and classification in biological science journal articles. *Proceedings of the Computational Terminology for Medical and Biological Applications Workshop of the 2nd International Conference on Natural Language Processing (NLP'2000)*, pp. 37–44.

Jelinek,F. (1989) Self-organizing language modeling for speech recognition. In Waibel,A. and Lee,K.-F. (eds), *Readings in Speech Recognition*. Morgan Kaufmann, Publishers, Palo Alto, CA, pp. 450–506.

Katz,S.M. (1987) Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. Acoust. Speech Signal Proc.*, **35**, 400–401.

Kazama,J., Makino,T., Ohta,Y. and Tsujii,J. (2002) Tuning support vector machines for biomedical named entity recognition. *Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain (ACL'2002)*. Philadelphia, PA, USA, July, pp. 1–8.

Lee,K.J. Hwang,Y.S. and Rim,H.C. (2003) Two-phase biomedical NE recognition based on SVMs. *Proceedings of the ACL'2003 Workshop on Natural Language Processing in Biomedicine*. Sapporo, Japan, pp. 33–40.

Liu,H., Aronson,A.R. and Friedman,C. (2002) A study of abbreviations in MEDLINE abstracts. *Proceedings of the American Medical Informatics Association Symposium 2002*. PA, USA, pp. 327–332.

Marcus,M.P., Marcinkiewicz,M.A. and Santorini,B. (1993) Building a large annotated corpus of English: the Penn Treebank. *Comput. Linguist.*, **19**, 310–330.

MUC6 (1995) Morgan Kaufmann Publishers, Inc. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Columbia, MD, USA.

MUC7 (1998) Morgan Kaufmann Publishers, Inc. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Fairfax, VA, USA.

Nobata,C., Collier,N. and Tsujii,J. (1999) Automatic term identification and classification in biology texts. *Proceedings of the 5th Natural Language Rim Symposium (NLPRS'99)*. Beijing, China, November, pp. 369–374.

Nobata,C., Collier,N. and Tsujii,J. (2000) Comparison between tagged corpora for the named entity task. In Kilgarriff,A. and Berber Sardinha,T. (eds), *Proceedings of the Workshop on Comparing Corpora (ACL'2000)*. Hong Kong University of Science and Technology, pp. 20–27.

Ohta,T., Tateisi,Y., Kim,J., Mima,H. and Tsujii,J. (2002) The GENIA corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of Human Language Technology (HLT' 2002)*. San Diego, pp. 489–493.

Proux,D., Rechenmann,F., Julliard,L., Pillet,V. and Jacq,B. (1998) Detecting gene symbols and names in biological texts: a first step toward pertinent information extraction. *Genome Inform.*, **9**, 72–80.

Sekimizu,T., Park,H. and Tsujii,J. (1998) Identifying the interaction between genes and gene products based on frequently seen verbs in medline abstracts. Genome Inform Ser. Workshop *Genome Inform.*, **9**, 62–71.

Takeuchi,K. and Collier,N. (2002) Use of support vector machines in extended named entity recognition. *Proceedings of the Sixth Conference on Natural Language Learning (CONLL' 2002)*. Taipei, Taiwan, August, pp. 119–125.

The Flybase Consortium (1998) Flybase—a *Drosophila* database. *Nucleic Acids Res.*, **26**, 85–88.

Thomas,J., Milward,D., Ouzounis,C., Pulman,S. and Carroll,M. (2000) Automatic extraction of protein interactions from scientific abstracts. *Proceedings of the Pacific Symposium on Biocomputing'2000 (PSB'2000)*. Honolulu, Hawaii, 4–9 January, pp. 541–551.

Tsuruoka,Y. and Tsujii,J. (2003) Boosting precision and recall of dictionary-based protein name recognition. *Proceedings of the ACL'2003 Workshop on Natural Language Processing in Biomedicine*. Sapporo, Japan, 11 July, pp. 41–48.

van Rijsbergen,C.J. (1979) *Information Retrieval*, 2nd edn. Butterworth, London.

Viterbi,A.J. (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theory*, **13**, 260–269.

Zhou,G.D. and Su,J. (2002) Named entity recognition using an HMM-based chunk tagger. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'2002)*. Philadelphia, July, pp. 473–480.