Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Biomedical Informatics 37 (2004) 411–422
Journal of Biomedical Informatics
www.elsevier.com/locate/yjbin

Enhancing HMM-based biomedical named entity recognition by studying special phenomena

Jie Zhang[a,b,*], Dan Shen[a,b,1], Guodong Zhou[a], Jian Su[a], Chew-Lim Tan[b]

[a] Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613, Singapore
[b] Department of Computer Science, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore

Received 22 July 2004
Available online 25 September 2004

Abstract

The purpose of this research is to enhance an HMM-based named entity recognizer in the biomedical domain. First, we analyze the characteristics of biomedical named entities. Then, we propose a rich set of features, including orthographic, morphological, part-of-speech, and semantic trigger features. All these features are integrated via a Hidden Markov Model with back-off modeling. Furthermore, we propose a method for biomedical abbreviation recognition and two methods for cascaded named entity recognition. Evaluation on the GENIA V3.02 and V1.1 shows that our system achieves 66.5 and 62.5 *F*-measure, respectively, and outperforms the previous best published system by 8.1 *F*-measure on the same experimental setting. The major contribution of this paper lies in its rich feature set specially designed for biomedical domain and the effective methods for abbreviation and cascaded named entity recognition. To our best knowledge, our system is the first one that copes with the cascaded phenomena.

Keywords: Biomedical named entity recognition; Cascaded named entity recognition; Abbreviation recognition; HMM

1. Introduction

Named entity recognition (NER) automatically identifies names in texts and classifies them into predefined classes. The NER task is defined by the message understanding conferences (MUC), where the names of entities in the newswire domain, such as PERSON, ORGANIZATION, LOCATION, and etc., are recognized. With the exploding amount of biomedical literatures, NER is strongly demanded in the biomedical domain. It is a useful technique for many applications, such as text mining in the biomedical domain, bioinformatics tools, biomedical database building, etc. In the biomedical domain, not only the names of entities, such as protein, gene, and virus, but also the names of some concepts, such as names of biomedical processes, need to be recognized.

In previous research work, many NER systems have been applied successfully in the newswire domain [1–3]. Recently, more and more explorations have been done to port existing NER systems into the biomedical domain [4–10,12]. Since most of the systems are evaluated on different training data, we can hardly make comparison. However, there is still some room for improvement based on the reported results of these systems.

In this paper, we will study how to adapt a general Hidden Markov Model (HMM)-based named entity recognizer [1] to the biomedical domain. We specially explore various features for biomedical named entities and propose methods to cope with abbreviations and

* Corresponding author.
E-mail addresses: zhangjie@i2r.a-star.edu.sg (J. Zhang), shendan@i2r.a-star.edu.sg, dshen@coli.uni-sb.de (D. Shen), zhougd@i2r.a-star.edu.sg (G. Zhou), sujian@i2r.a-star.edu.sg (J. Su), tancl@comp.nus.edu.sg (C.-L. Tan).
[1] Present address: Universität des Saarlandes, Computational Linguistics Department, 66041 Saarbrücken, Germany.

1532-0464/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2004.08.005

cascaded named entity phenomena. As a result, various features (orthographic, morphological, part-of-speech, and head noun trigger features) and methods (an abbreviation recognition algorithm and two cascaded named entity recognition methods) are integrated in our system. The experiment shows that our system significantly outperforms the previous best published system.

The rest of the paper is organized as follows. Section 2 summarizes special characteristics of biomedical named entities. In Section 3, we provide detailed description of our HMM-based named entity recognition model and present various methods for abbreviation recognition and cascaded named entity recognition. In Section 4, the rich feature set is described in detail. In Section 5, we show our experimental configurations and various experimental results. Section 6 presents the related work. Finally, in Chapter 7 we conclude this paper with future work.

2. Characteristics of biomedical named entities

Since named entity recognition in the newswire domain is successful and mature, people may ask what else can be done in the biomedical domain and what difference exists between the two domains. To answer these questions, we study the special characteristics of various biomedical named entities to get a clear understanding before bringing out our solutions. In summary, biomedical named entities have following special characteristics:

1. Biomedical named entities often have pre-modifiers, e.g., *activated B cell lines*, and are sometimes very long, e.g., *47 kDa sterol regulatory element binding factor*. These are the main causes of difficulty of identifying the boundary of a named entity.
2. Two or more biomedical named entities can share one head noun by using the conjunction or disjunction construction, e.g., *91 and 84 kDa proteins*. It is hard to resolve such phenomenon.
3. One biomedical named entity may have various forms, e.g., *N-acetylcysteine*, *N-acetyl-cysteine*, *NAcetyl Cysteine*, etc. Especially, the capitalization information may not be so useful in this domain, since the use of capitalization is casual.
4. Biomedical named entities may be cascaded. One named entity may be embedded in another named entity, e.g., *<PROTEIN><DNA>kappa 3</DNA> binding factor </PROTEIN>*. More efforts have to be made to identify such named entities.
5. Abbreviations are frequently used in the biomedical domain, e.g., *TCEd*, *IFN*, *TPA*, etc. Since abbreviations carry less information than their full forms, it is more difficult to classify them.

These above factors make NER in the biomedical domain difficult. Therefore, it is necessary to explore rich features and effective methods to deal with the special characteristics in the biomedical domain.

3. Feature set

3.1. Orthographic features (F_o)

Orthographic features are designed to capture word formation information, such as capital letters, numeric characters, and their combinations. Orthographic information have been widely used in NER, such as [1,4–6,8,9]. Generally, orthographic features are manually designed and aim to group words by similar formations. In the biomedical domain, orthographic features are likely to be served as indicators of unknown words, such as unknown abbreviations. For example, suppose *IL-2* is in the training data, but *IL-12* is not. Fortunately, we can guess that *IL-12* is similar to *IL-2* based on their orthographic features. In our work, we manually design orthographic features based on the characteristics of biomedical names. Table 1 shows the list of orthographic features by the descending order of priority.

From Table 1, we can find that the features, such as *GreekLetter*, *RomanDigit*, *ATCGsequence*, are specially designed for the biomedical domain. The features dealing with mixed alphabets and digits, such as *AlphaDigitAlpha*, *CapMixAlpha*, etc., are beneficial for biomedical abbreviations. Moreover, the features, such as *ATCGsequence*, identify the similarity of the words

Table 1
Orthographic features

F_o Name	Example
Comma	,
Dot	.
LeftRoundBracket	(
RightRoundBracket	)
LeftSquareBracket	[
RightSquareBracket	]
RomanDigit	II
GreekLetter	Beta
StopWord	in, at
ATCGsequence	AACAAAG
OneDigit	5
AllDigits	60
DigitCommaDigit	1,25
DigitDotDigit	0.5
OneCap	T
AllCaps	CSF
CapLowAlpha	All
CapMixAlpha	IgM
LowMixAlpha	kDa
AlphaDigitAlpha	H2A
AlphaDigit	T4
DigitAlphaDigit	6C2
DigitAlpha	19D

according to their formations, e.g., *AACAAAG*, *CTCAGGA*, etc. Besides these, some features, such as *comma*, *dot*, *StopWord*, etc., are to provide information to detect the boundaries of named entities. Especially, parentheses, such as *LeftRoundBracket*, *RightRoundBracket*, *LeftSquareBracket*, and *RightSquareBracket*, are often used to indicate the definitions of biomedical abbreviations. In Section 4.2, we will explain how to make use of parentheses to deal with abbreviations in detail.

3.2. Morphological feature (F_m)

Morphological information, such as prefix and suffix, is considered as an important cue for terminology identification. In our system, we use a statistical method to get the most frequent 100 prefixes and suffixes from the training data as candidates. Then, each of these candidates is evaluated according to the Eq. (1).

$$Wt_i = \frac{(\#IN_i - \#OUT_i)}{N_i} \quad (1)$$

in which, $\#INi$ is the number that the prefix/suffix i occurs within NEs; $\#OUTi$ is the number that the prefix/suffix i occurs out of named entities; Ni is the total number of the prefix/suffix i.

Equation (1) assumes that the particular prefix/suffix, which is most likely inside and least likely outside named entities, may be thought useful. The candidates with Wt above a certain threshold (0.7 in our experimentation) are selected. Then, we calculate the frequency of each prefix/suffix in each entity class and group the prefixes/suffixes with the similar distribution among entity classes into one feature. This is because prefixes/suffixes with the similar distributions have similar contributions, and it avoids suffering from data sparseness problem. Some of morphological features are listed in Table 2.

From Table 2, the suffixes *∼cin*, *∼mide*, and *∼zole* are grouped into one feature *sOOC* because they all have high frequencies in the entity class *OTHER-ORGANIC-COMPOUND* and relatively low frequencies in the other entity classes. In our work, totally 37 prefixes and suffixes were selected and grouped to 23 features.

Table 2
Examples of morphological features

F_m Name	Prefix/Suffix	Example
sOOC	∼cin	Actinomycin
	∼mide	Cycloheximide
	∼zole	Sulphamethoxazole
sLPD	∼lipid	Phospholipids
	∼rogen	Estrogen
	∼vitamin	Dihydroxyvitamin
sCTP	∼blast	Erythroblast
	∼cyte	Thymocyte
	∼phil	Eosinophil
sPEPT	∼peptide	Neuropeptide
sMA	∼ma	Hybridoma
sVIR	∼virus	Cytomegalovirus

3.3. Part-of-speech features (F_{pos})

In the previous NER in the newswire domain, part-of-speech (POS) features are proven useless, as POS features may affect the use of some important capitalization information [1]. However, the capitalization information in the biomedical domain is not as useful as it in the newswire domain. Moreover, since many biomedical named entities are descriptive and long, identifying entity boundary is not a trivial task. As a syntactical feature, POS tagging can help to capture the noun phrase region. Therefore, it is useful for biomedical NER, based on the assumption that a named entity is more likely to be a noun phrase.

In our work, we adapt a HMM-based POS tagger to the biomedical domain by using GENIA corpus as the training data. The POS tagger achieves the precision of 97.37 using 80% of GENIA V2.1 corpus (536 abstracts, 123K words) as training data and the rest 20% (134 abstracts, 29K words) as test data. In our NER system, each word is assigned a POS feature by this POS tagger.

3.4. Semantic trigger features

We design semantic trigger features to indicate certain entity classes based on the semantic information. Trigger words are key words inside or outside of named entities. Initially, we collected two types of semantic triggers: head noun triggers and special verb triggers.

3.5. Head noun triggers (F_{hnt})

The head noun is the main noun or noun phrase of some compound words and describes the function or the property, e.g., *B cells* is the head noun for the named entity *activated human B cells*. Compared with the other words in named entities, the head noun is a much more decisive factor for distinguishing entity classes. For instance,

<OTHER-NAME>IFN-gamma treatment</OTHER-NAME>
<DNA>IFN-gamma activation sequence</DNA>

Both of the instances above begin with the *IFN-gamma* with only a difference in head nouns, *treatment* and *sequence*. These two biomedical named entities belong to two different classes: *OTHER-NAME* and *DNA*. This example implies that no matter how many similar expressions are within entities, entity classes are normally determined by head nouns. The usefulness of the head noun is also supported by [11].

Table 3
Examples of head noun triggers

Class	1-gram	2-grams
PROTEIN	Kinase	Binding protein
	Interleukin	Activator protein
	Interferon	Cell receptor
	Ligand	Gene product
VIRUS	Virus	Recombinant virus
	Provirus	Lymphotropic herpesvirus
	Cytomegalovirus	Virus particles
	Adenovirus	Immunodeficiency virus
DNA	DNA	X chromosome
	Breakpoint	α-Promoter
	cDNA	Binding motif
	Chromosome	Promoter element

Table 4
Special verb triggers

Activate	Associate	Bind	Block	Clone
Demonstrate	Express	Identify	Increase	Induce
Inhibit	Investigate	Involve	Isolate	Mediate
Observe	Reduce	Regulate	Reveal	Stimulate

In our work, we extract unigram and bi-grams head nouns automatically from the training data, and rank them by their frequencies. According to the experiment, we select 60% of top ranked head nouns as trigger features for each entity class. Some examples are shown in Table 3.

In the future application, we may also extract head nouns from public resources.

3.6. Special verb triggers (F_{svt})

Besides collecting trigger words inside named entities, such as head noun triggers, we can also use trigger words from the local context of named entities. Recently, some frequent verbs in MEDLINE have been proven useful for extracting interactions between biomedical entities, e.g., the protein–protein interactions [14,15]. Therefore, we have intuition that particular verbs may also be useful for biomedical NER. For instance, the verb *bind* often indicates the interaction between proteins.

In our work, we selected 20 most frequent verbs which occur adjacent to named entities from the training data automatically as the verb trigger features, which are shown in Table 4.

4. Methods

4.1. HMM-based named entity recognizer

Hidden Markov Model (HMM) is a statistical method. In the past 15 years, HMM has been successfully used in a wide range of applications, such as speech recognition and natural language processing. In HMM, a sequence of output symbols is generated in addition to a 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."

In named entity recognition, the input word sequence, e.g., sentence, can be regarded as the observed sequence and the output tag sequence is the statistically optimal state sequence corresponding to the observed word sequence.

In our work, the name entity recognizer is adapted from the previous work, the HMM-based Named Entity Recognizer on MUC [1]. The core technique is a Hidden Markov Model described as follows:

The named entity recognizer tries to find the most likely tag sequence $T_1^n = t_1 t_2 \cdots t_n$ for a given sequence of tokens $O_1^n = o_1 o_2 \cdots o_n$ that maximizes $P(T_1^n|O_1^n)$. In the token sequence O_1^n, the token o_i is defined as $o_i = \langle f_i, w_i \rangle$, where w_i is the ith word and f_i is the feature set assigned to the word w_i. The feature set is introduced in Section 3. In the tag sequence T_1^n, each tag t_i is structural and consists of three parts: the boundary category, the entity class and the feature set. The boundary category indicates whether the word itself is a named entity, or the word is at the beginning, in the middle, or at the end of a named entity. The entity class consists of a NOT-NAME class and a predefined set of entity classes. The feature set is added in order to represent more accurate models based on the limited number of boundary categories and entity classes.

In the model, $P(T_1^n|O_1^n)$ an be represented as

$$\log P(T_1^n|O_1^n) = \log P(T_1^n) + \log \frac{P(T_1^n, O_1^n)}{P(T_1^n) \cdot P(O_1^n)}. \tag{2}$$

The second term of the right-hand side of Eq. (2) is the mutual information between T_1^n and O_1^n. We assume mutual information independence:

$$\log \frac{P(T_1^n, O_1^n)}{P(T_1^n) \cdot P(O_1^n)} = \sum_{i=1}^{n} \log \frac{P(t_i, O_1^n)}{P(t_i) \cdot P(O_1^n)}. \tag{3}$$

Applying Eq. (3) to (2), we have:

$$\log P(T_1^n|O_1^n) = \log P(T_1^n) - \sum_{i=1}^{n} \log P(t_i) + \sum_{i=1}^{n} \log P(t_i|O_1^n) \tag{4}$$

The first term in the Eq. (4) can be computed by applying chain rules. Each tag is assumed to be probabilistically dependent on the $N-1$ previous tags in the N-gram modeling. The second term is the sum of log probabilities of all the tag instances. Ideally, the third term can be estimated by the forward–backward algorithm recursively [16]. For efficiency, an alternative back-off modeling approach by means of constraint relaxation was applied in our model. This approach enables the decoding process effectively find a near optimal frequently occurred pattern entry in determining the

tag probability distribution of the current word. Details of the back-off modeling can be found in [1].

The Viterbi algorithm [17] is implemented to find the most likely tag sequence in the state space of the possible tag distribution based on the state transition probabilities. Meanwhile, some constraints on the boundary category and entity category between two consecutive tags are applied to filter the invalid name tags.

4.2. Method for abbreviation recognition

Abbreviations are widely used in the biomedical domain. Therefore, it is important to resolve this problem in the biomedical domain.

In our current system, we incorporate a method to classify an abbreviation by mapping the abbreviation to its full form. This approach is based on the assumption that it is easier to classify the full form than its abbreviation. In most cases, this assumption is valid because the full form has more information than its abbreviation to capture its entity class. Moreover, if we can map the abbreviation to its full form, the recognized abbreviation is also helpful for classifying the same forthcoming abbreviations within the current document.

In practice, the abbreviation and its full form often occur simultaneously with parenthesis when an abbreviation first appears in biomedical documents [18,20]. There are often two cases:

1. full form (abbreviation)
2. abbreviation (full form)

Most patterns conform to the first case and if the content inside the parenthesis consists of more than two words, the second case is assumed [18].

In these two cases, the use of parenthesis is both evidential and confusing. On one hand, it is evidential because it indicates the mapping of an abbreviation to its full form. On the other hand, it is confusing because it makes the annotation more complicated and inconsistent. Sometimes, an abbreviation and its full form are annotated separately, as

<CELL-TYPE>human mononuclear leukocytes</CELL-TYPE> (*<CELL-TYPE>hMNL</CELL-TYPE>*),

and sometimes, they are all embedded in a whole entity, such as

<OTHER-NAME>leukotriene B4 (LTB4) generation</OTHER-NAME>.

Therefore, parenthesis needs to be treated specially. In this paper, we develop an abbreviation recognition algorithm described in Fig. 1.

```
for each sentence S_i in the document{
  if exist parenthesis{
    judge the case of {
    "full form (abbr.)";
    "abbr. (full form)";
    }
    store the abbr. A and position P_a to a list;
    record the parenthesis position P_p;
    remove A and parenthesis from sentence;
    apply HMM-based named entity recognizer to S_i;
    restore A and parenthesis into P_a, P_p;
    if P_p within an identified named entity E with the class C_E
      parenthesis is included in E;
    else{
      parenthesis is not included;
      classify A to C_E;
      classify A in the rest part of document to C_E;
    }
  }
  else apply HMM-based named entity recognizer to S_i;
}
```

Fig. 1. Algorithm for abbreviation recognition.

The main idea of the algorithm in Fig. 1 is described as follows. In the preprocessing stage, we remove the abbreviation and parentheses from the sentence, where an abbreviation is first defined. This measure will make the annotation simpler and the recognizer more effective. Then, we determine which case the abbreviation definition belongs to and record the original positions of the abbreviation and parentheses. After applying our named entity recognizer to the sentence, we restore the abbreviation and parentheses to the recorded positions. Next, the abbreviation is classified based on the two priorities (from high to low): the class of its full form and the class of the abbreviation itself identified by the recognizer. Finally, the same abbreviations in the rest sentences of the current document are assigned the same entity class.

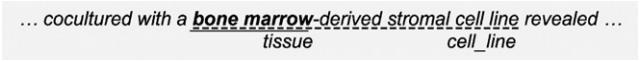
4.3. Methods for cascaded named entity recognition

As mentioned in Section 2, cascaded-annotation is a special problem in biomedical NER. For instance, "*<CELL-LINE><VIRUS>HTLV-I</VIRUS>*-infected cord blood lymphocytes *</CELL-LINE>*" belongs to the class *CELL-LINE* and embeds a virus name *HTLV-I*. In cascaded named entity recognition, we shall recognize both the embedded and the longest named entities. However, people currently care more about the longest named entities for two reasons. First, the longest named entities are more likely to be the subjects that people want to study. Second, they keep all information about the embedded named entities. Therefore, whether tagging the embedded named entities or not depends on user requirements under different circumstances. In our work, we propose two approaches: a post-processing

rule-based cascaded recognition approach and an HMM-based cascaded recognition approach.

4.4. Post-processing rule-based cascaded recognition approach

The post-processing rule-based approach aims to recognize the longest named entities. In the previous non-cascaded recognition, the named entity recognizer may fail to recognize some of the longest named entities, since it sometimes recognizes embedded named entities. For example:



In the above case, the previous system recognizes the embedded *TISSUE* "*bone marrow*," while we prefer to recognize the entity *CELL-LINE*. The cause of errors of this type is that a substring embedded in a named entity itself is another named entity and is recognized as such to prevent the whole string from being correctly recognized. In the above example, the word "*marrow*" is a head noun for *TISSUE*, thus "*bone marrow*" is likely to be recognized as the entity *TISSUE*. Although "*cell line*" is a head noun for *CELL-LINE*, the context is too long to capture.

We propose a post-processing rule-based approach to deal with these cases. The main idea is that we develop a set of patterns which help recognize the longest named entities based on the embedded ones. From GENIA corpus annotation, we collect four basic patterns of cascaded named entities. In addition, we also extend the patterns by combining the basic ones iteratively as shown in Table 5.

Based on these patterns, we can construct a rule set automatically from the training corpus. Table 6 shows some post-processing rules.

After named entity recognition by our recognizer, we get an initial result. Then, we develop a post-processing procedure by applying the rules above to the initial result. For example, given the initial result "*a <PROTEIN>Myc-associated zinc finger protein</PROTEIN> binding site is one of ...*," the post-processing procedure finds that it matches the rule "*<DNA> = <PROTEIN> binding site*." After post-processing, the final result turns to "*a <DNA> Myc-associated zinc finger protein binding site </DNA> is one of ...*" In addition, the post-processing rules are applied iteratively until no new match can be found. For example, given the initial result "*...<AMINO-ACID-MONOMER>tyrosine </AMINO-ACID-MONOMER>kinase inhibitor ...*," the post-processing procedure finds that it matches the rule "*<PROTEIN> = <AMINO-ACID-MONOMER> kinase*" and changes it to "*...<PROTEIN> tyrosine kinase </PROTEIN>* inhibitor ..." in the first iteration. In the next iteration, the post-processing procedure finds that the intermediate result matches the rule "*<OTHER-ORGANIC-COMPOUND> = <PROTEIN> inhibitor*" and updates it again. Since no more matches will occur in the following iterations, the final result is "*...<OTHER-ORGANIC-COMPOUND>* tyrosine kinase inhibitor *</OTHER-ORGANIC-COMPOUND> ...*" In this way, we are able to recognize the longest named entity.

Table 5
Patterns of cascaded named entities

Basic patterns
<NAME'> = <NAME> [head nouns]
<NAME'> = [modifier] <NAME>
<NAME'> = <NAME^1> <NAME^2>
<NAME'> = <NAME^1> [words] <NAME^2>
Extended patterns
<NAME'> = [modifier] <NAME> [head nouns]
<NAME'> = [modifier] <NAME^1> <NAME^2>
<NAME'> = <NAME^1> <NAME^2> [head nouns]
...

Table 6
Examples of post-processing rules for cascaded named entity recognition

Rule instance	<DNA> = <PROTEIN> binding site
From pattern	<NAME'> = <NAME> [head nouns]
Example	A Myc-associated zinc finger protein binding site is one of ...
Rule instance	<PROTEIN> = <VIRUS> <PROTEIN>
From pattern	<NAME'> = <NAME^1> <NAME^2>
Example	Nevertheless, the simian EBV LMP1s retain most functions in ...
Rule instance	<CELL-TYPE> = human <CELL-TYPE>
From pattern	<NAME'> = [modifier] <NAME>
Example	...suggests that human NK cells provide an effective ...
Rule instance	<CELL-TYPE> = <VIRUS>-infected <CELL-TYPE>
From pattern	<NAME'> = <NAME^1> [words] <NAME^2>
Example	... p24 production by HIV-infected human macrophages when ...

4.5. HMM-based cascaded recognition approach

Besides the post-processing rule-based approach, we also propose an HMM-based cascaded recognition approach to recognize named entities with cascaded-annotation. The HMM-based cascaded recognition approach starts from the shortest embedded named entity and extends to the longer named entity iteratively.

We train two HMM models in this approach. The first model is our named entity recognizer, which is mainly to recognize short embedded named entities. Besides this, we also train another HMM model to iteratively extend the short entities. To train this iterative

model, we use the cascaded-annotations of the GENIA corpus and transform them into a new training data set. For example:

<DNA><PROTEIN>*Myc-associated zinc finger protein*<PROTEIN> *binding site*</DNA>

<DNA> [PROTEIN] *binding site*</DNA>

We substitute a class-representing token "[PROTEIN]" for the embedded name of protein. After this transformation, all cascaded-annotated entities in the training data become non-cascaded. We train a HMM model on this training data as a cascaded recognition model. Intuitively, the HMM model captures local context information more easily than long context information. Some long cascaded named entities may be difficult to be recognized in one pass as shown in the previous section. We hope that they can be recognized by two or more iterations if they are missed in the first pass. Therefore, we can use the same HMM method iteratively and do not need any post-processing step. One limitation of this approach may be that the following iterations rely on the first recognition pass. In an ideal situation, if the performance is high in the first pass, the longer named entities are likely to be recognized. In our work, we just concern about the performance of the longest named entities, so that the evaluation is conducted on them. The algorithm of the HMM-based cascaded recognition approach is shown in Fig. 2.

In addition, we also generalize the model to a recursive process which recognizes all levels of the cascaded named entities, i.e., not only the longest named entities but also the embedded ones. The algorithm for this generalized method is shown in Fig. 3.

```
for each sentence Si in the document{
  apply the first pass HMM-based NER model to Si ;
  for each recognized named entity Nj {
    record Nj to a stored-list;
    substitute a class-label token CT(Nj) for Nj in Si ;
  }
  loop until no named entity can be recognized in Si{
    apply the iterative recognition HMM model to Si ;
    for each Nj in the stored-list {
      if CT(Nj) is embedded in newly recognized named entity N'k {
        restore content of Nj to original position CT(Nj) in Si ;
        remove Nj from the stored-list;
      }
    }
    for each newly recognized named entity N'k {
      record N'k to a stored-list;
      substitute a class-label token CT(N'k) for N'k in Si ;
    }
  }
}
```

Fig. 2. Algorithm for the HMM-based cascaded recognition approach.

```
for each sentence Si in the document{
  apply first pass HMM named entity recognition model to Si
  for each recognized named entity Nj {
    record Nj to a stored-list;
    substitute a class-label token CT(Nj) for Nj in Si
  }
  recursive-recognize-cascaded-named-entity(Si);
  for each Nj in the stored-list {
    restore Nj to original position CT(Nj) in Si ;
  }
}

function recursive-recognize-cascaded-named-entity(sentence S){
  apply the iterative recognition HMM model to S ;
  if no named entity is recognized then return ;
  for each recognized named entity Ni {
    record Ni to a local stored-list;
    substitute a class-label token CT(Ni) for Ni in S ;
  }
  recursive-recognize-cascaded-named-entity(S);
  for each Ni in the stored-list {
    restore Ni to original position CT(Ni) in S ;
  }
}
```

Fig. 3. Algorithm of a generalized recursive method for all-level cascaded named entity recognition.

5. Experiments

5.1. GENIA corpus

Currently, the GENIA corpus[2] is the largest annotated text resource in the biomedical domain available to public [19]. The annotation of the biomedical named entities is based on the GENIA ontology. In our task, we recognize 22 distinct entity classes[3] defined in the GENIA ontology, including *MULTI-CELL*, *MONO-CELL*, *VIRUS*, *BODY-PART*, *TISSUE*, *CELL-TYPE*, *CELL-COMPONENT*, *CELL-LINE*, *OTHER-ARTIFICIAL-SOURCE*, *PROTEIN*, *PEPTIDE*, *AMINO-ACID-MONOMER*, *DNA*, *RNA*, *POLYNUCLEOTIDE*, *NUCLEOTIDE*, *LIPID*, *CARBOHYDRATE*, *OTHER-ORGANIC-COMPOUND*, *INORGANIC*, *ATOM*, and *OTHER*. In our experiment, three versions are used, which are V1.1, V2.1, and V3.02.

GENIA Version 1.1 (V1.1)—It consists of 670 MEDLINE abstracts. Since a lot of previous related works are based on this version, we use it to compare our result with the others'.

GENIA Version 2.1 (V2.1)—It consists of the same 670 abstracts as V1.1 with additional part-of-speech tagging. We use this version to adapt the part-of-speech tagger to the biomedical domain as mentioned in Section 3.3.

[2] Downloaded from http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/.

[3] In previous work on GENIA V1.1, there are 23 name classes due to inconsistent annotations of class *ORGANISM*. According to GENIA ontology, *ORGANISM* is not a name class in V3.02. We do not differentiate the subclasses of *PROTEIN*, *DNA*, and *RNA*.

GENIA Version 3.02 (V3.02)—It consists of 2000 MEDLINE abstracts, which is a superset of the GENIA Version 1.1. We use this version to get the latest result and find out the effect of training data size.

5.2. Experimental results

The performance of our system is evaluated using "precision/recall/*F*-measure," in which "precision" is calculated as the ratio of the number of correctly found named entities to the total number of named entities found by our model; "recall" is calculated as the ratio of the number of correctly found named entities to the number of true named entities; and "*F*-measure" is defined by formula (5).

$$F\text{-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (5)$$

We conduct experiments for the biomedical NER on the both GENIA V1.1 and V3.02. For the GENIA V1.1, we split the corpus into a training set of 590 abstracts and a test set of 80 abstracts. We keep the same training/test ratio as [4] in order to make comparisons. For the GENIA V3.02, the 2000 abstracts are split to a training set of 1920 abstracts and a test set of 80 abstracts. The test set is the same as the test set for the GENIA V1.1. In summary, the setting for the biomedical NER is shown in Table 7.

On the GENIA V1.1, our system (62.5 *F*-measure) outperforms [4] (54.4 *F*-measure) by 8.1 *F*-measure. It probably benefits from the rich features and the effective methods proposed. Furthermore, as expected, the performance on the GENIA V3.02 (66.5 *F*-measure) is better than that on the V1.1 (62.5 *F*-measure).

Besides the overall performance, we also evaluate performances of all the entity classes, which are shown in Fig. 4. From Fig. 4, we can find that the performances vary a lot among the different entity classes. It is probably due to two reasons. First, different entity classes have different difficulties for the named entity recognition. For example, *BODY-PART* is one of the easiest entity classes since the number of instances for body part is limited. Second, the numbers of the training and test instances are not evenly distributed among all the entity classes. Some minor classes, such as *NUCLEOTIDE*, *ATOM*, *INORGANIC*, *CARBOHYDRATE* and, etc., lack enough data to achieve acceptable performances.

Table 7
Overall performance of biomedical named entity recognition on GENIA corpus V3.02 and V1.1, comparing to Kazama's [4] on GENIA corpus V1.1.

	Precision	Recall	*F*-measure
Our model on V3.02	67.7	65.3	66.5
Our model on V1.1	63.8	61.3	62.5
Kazama's [4] on V1.1	56.2	52.8	54.4

Furthermore, in order to evaluate the contributions of the different features, we evaluate our system using the different combinations of the features. The results are shown in Table 8.

From Table 8, several findings are concluded:

(1) Based on the orthographic feature (F_o), our system achieves a basic level performance of 28.7 *F*-measure. In MUC-7 task, performance can reach 77.6 *F*-measure by using the orthographic feature only [1]. It suggests that in the biomedical domain the orthographic feature is not so informative.
(2) The head noun trigger feature (F_{hnt}) is proven very useful. It greatly improves the *F*-measure (+21.1 based on F_o; +18.3 based on $F_o + F_m$; +9.4 based on $F_o + F_{pos}$; +7.3 based on $F_o + F_m + F_{pos}$).
(3) The part-of-speech feature (F_{pos}) also makes significant improvement on *F*-measure (+23.6 based on F_o; +23.1 based on $F_o + F_m$; +11.9 based on $F_o + F_{hnt}$; +12.1 based on $F_o + F_m + F_{hnt}$). It greatly benefits from the effective adaptation of the part-of-speech tagger to the biomedical domain.
(4) The morphological feature (F_m) leads to the positive effect by +2.7 *F*-measure improvement based on F_o and +2.2 *F*-measure improvement based on $F_o + F_{pos}$. However, it cannot make improvement based on $F_o + F_{hnt}$ and can only slightly improve the recall by +0.2 based on $F_o + F_{pos} + F_{hnt}$. The probable reason is that F_m and F_{hnt} provide some overlapping information. The information captured by F_m may also be captured by F_{hnt}. Moreover, the information captured by F_{hnt} is more accurate than that captured by F_m. The contribution made by F_m may come from where there is no indication of F_{hnt}.
(5) Out of our expectation, the special verb trigger feature (F_{svt}) decreases both precision and recall and degrades the *F*-measure by 1.8 based on $F_o + F_m + F_{pos} + F_{hnt}$.

To evaluate our proposed methods for abbreviation and cascaded named entity recognition, we make further experiments based on the four features which lead to the best performance as shown above. The results are summarized in Table 9.

First, we evaluate the contribution of the abbreviation recognition method. The result shows that the method leads to an improvement on *F*-measure by 1.2 based on the best combination of features $F_o + F_m + F_{pos} + F_{hnt}$ (4F). The reason why the improvement is not so significant is that our abbreviation recognition method mainly relies on the recognition of its full form. Once the full form is wrongly recognized, all abbreviations can be wrong altogether. However, the principle of the method is rea-

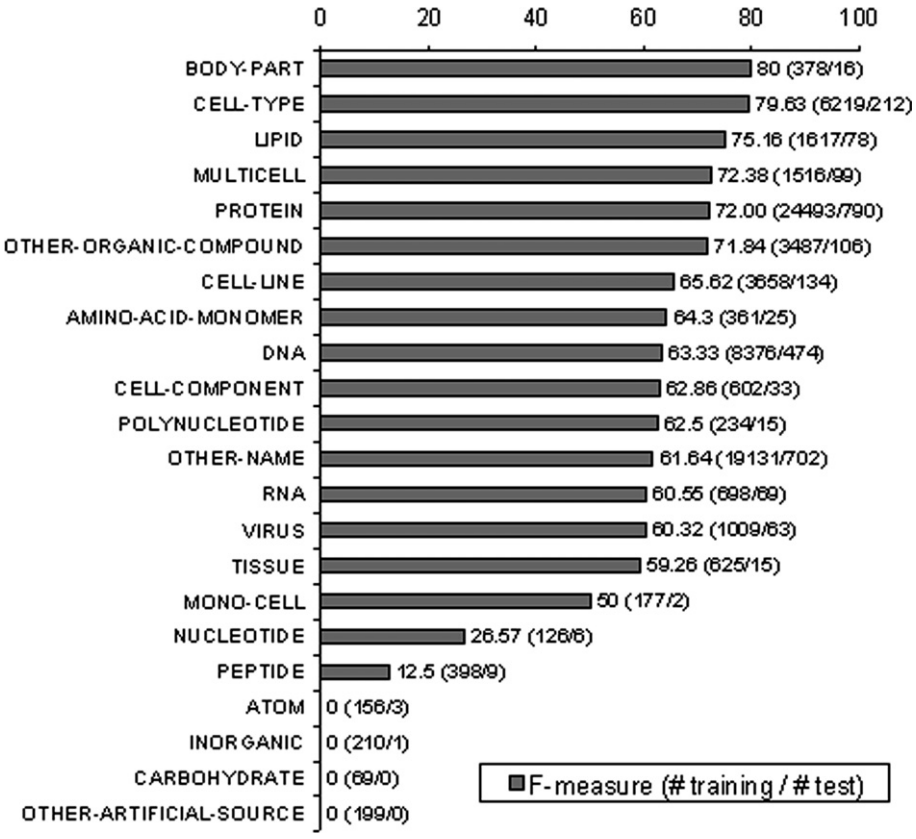


Fig. 4. Performance of each named entity class.

Table 8
Experimental results for biomedical named entity recognition by using different combinations of features

F_o	F_m	F_{pos}	F_{hnt}	F_{svt}	Precision	Recall	F-measure
✓					41.8	21.8	28.7
✓	✓				44.4	24.3	31.4
✓		✓			55.7	49.4	52.3
✓			✓		55.9	44.9	49.8
✓	✓	✓			58.0	51.3	54.5
✓	✓		✓		55.8	44.8	49.7
✓		✓	✓		61.9	61.5	61.7
✓	✓	✓	✓		**61.9**	**61.7**	**61.8**
✓	✓	✓	✓	✓	60.6	59.3	60.0

Table 9
Effectiveness of abbreviation recognition method and two cascaded named entity recognition methods

V3.02	Precision	Recall	F-measure
$F_o + F_m + F_{pos} + F_{hnt}$ (4F)	61.9	61.7	61.8
4F + abbreviation recognition	63.4	62.7	63.0
4F + abbr. + post-processing rule-based appr.	**67.7**	**65.3**	**66.5**
4F + abbr. + HMM-based appr.	65.5	63.0	64.2

sonable and the result is positive. Our abbreviation recognition method provides an effective and reasonable solution when domain-specific abbreviation dictionaries are not available.

Furthermore, we evaluate the two approaches for cascaded named entity recognition proposed in section 4.3. Using post-processing rule-based cascaded recognition approach, we get a significant improvement by 3.5 F-measure. Another approach, the HMM-based cascaded recognition approach, also leads to a positive effect of +1.2 F-measure. We can find that the post-processing rule-based approach outperforms the HMM-based approach. It is probably because we do not have enough training data with the cascaded annotation for the HMM-based approach to get a reliable performance. However, the HMM-based approach is more general and can be enhanced when we have more training instances with the cascaded annotation.

6. Related work

This section presents a review of the recent literatures on the biomedical named entity recognition. We group them into rule-based and machine learning-based approaches.

6.1. Rule-based approaches

As for rule-based approaches, the representative research efforts include [10,12,9].

Fukuda et al. [10] proposes a method called PROPER (Protein Proper-noun phrase Extracting Rules), which attempts to identify protein names from biomedical documents based on surface clues of character strings, such as the presence of upper cases and special characters. They summarize the nomenclature of the protein names into three categories based on the surface characteristics of word. Their system is evaluated on 30 annotated MEDLINE abstracts on SH3 domain and achieves the precision of 91.90% and the recall of 93.32%.

Proux et al. [12] detects gene names in biomedical documents based on lexical and morphological information. They make use of a finite state-based tagger to conduct the lexical and morphological analysis of each word in the first level. The tagger tokenizes the sentences, conducts a lexical lookup to process the morphological analysis and performs the part-of-speech tagging. Each word in the sentence is given various tags and a special flag. The tags include noun, proper noun and abbreviation, etc. The special flag indicates whether the word matches a known word or is "guessed." Based on the tags and the special flag, they build a series of rules including recovery rules, algorithmic rules, and contextual rules. Their system achieves the precision of 91.4% and the recall of 94.4% on a small corpus (1200 sentences) from FlyBase. However, they find that when they apply the system to a larger corpus (25,000 MEDLINE abstracts) and evaluate the performance by sampling, the precision is reduced to around 70%.

Gaizauskas et al. [9] derives their system from a developed Information Extraction system in the MUC. Their system consists of five processing stages: text processing, morphological analysis, term lookup, terminology parsing, and term matching. The main information resources include case-insensitive terminology lexicons (the component term of various categories) such as the resources from the public databases (SWISS-PROT, CATH, and SCOP), morphological cues (standard biochemical suffixes) and hand-constructed grammar rules for each terminology class. Their system is applied in two projects: extraction of information about enzymes and metabolic pathways (EMPathIE) and extraction of information about protein structure (PASTA). The EMPathIE system is designed for 10 named entity classes, such as compound, element, enzyme, etc., and achieves the precision of 86% and the recall of 68% on 6 full journal articles. The PASTA system is designed for 13 named entity classes, such as protein, species, residue, etc., and achieves the precision of 94% and the recall of 88% on 52 MEDLINE abstracts.

Tanabe and Wilbur [13] proposes a method using a combination of statistical and knowledge-based strategies. They use a transformation-based part-of-speech tagger to generate rules automatically, as well as manually generated rules concerning morphological and part-of-speech information, low frequency trigrams, suffixes and indicator terms. They conduct experiments to detect protein/gene names (class GENE). The test set consists of 56469 MEDLINE abstracts. They randomly check 100 sentences out of every 50K sentences in the test set. Their method achieves the precision of 85.7% and the recall of 66.7%.

Although these rule-based systems seem quite promising, it is costly to adapt them to new entity classes in the biomedical domain. Once a new entity class is defined, a set of new rules has to be prepared manually. Consequently, the more classes are, the more difficult to construct consistent rules. Moreover, up to now, the evaluations of these systems are only based on small corpora. Proux et al. [12] reports their system fails in a larger corpus. Rule-based systems seem not to be robust and flexible.

6.2. Machine-learning approaches

Currently, the machine learning-based approaches become more and more popular in the biomedical named entity recognition. The typical works include [6,8,5,4].

Nobata et al. [6] tries two classification methods and three identification methods for the biomedical named entity recognition. The first classification method induces a Naïve Bayes classifier using conditional probabilities between word and class from the distribution of words in pre-classified domain-specific word lists. The second classification method uses a decision tree approach which incorporates the feature sets of part-of-speech information, character type information and domain specific word lists. The three identification methods include shallow parsing, decision trees and statistical identification. The system recognizes 10 entity classes, such as protein, DNA, RNA, cell line, cell type, etc. They conduct a series of experiments by combination of each classification and identification method. The experiments show that by using both decision tree methods for classification and identification, they achieve the best *F*-measure of 56.98–66.24 on 100 manually annotated MEDLINE abstracts by 5-fold cross validation. The corpus is a preliminary version of the GENIA corpus.

Collier et al. [8] applies linear interpolating HMM for gene name recognition. They train the HMM entirely based on surface word and character information. The classes and the corpus are the same as those in [6]. The system achieves the *F*-measure of 72.8.

Takeuchi and Collier [5] uses SVM. The model incorporates surface word, orthographic feature and the class assignments of context words. The window size of context is −3 to +3. In their experiment, they find that part-of-speech features degrade the performance in their

model. The evaluation is also conducted on the same corpus as used in [6] and the *F*-measure is 71.78.

Kazama et al. [4] also develops a system using SVM. To our knowledge, it is the earliest published work on the GENIA V1.1, which contains 670 MEDLINE abstracts and 24 named entity classes. Compared with [6], they make use of richer features, such as word feature, part-of-speech feature, prefix feature, suffix feature, previous class feature, word cache feature and HMM state feature. They use a BIO (beginning/in/out of entity) representation to classify a word. Since SVM is a binary classifier, they use the pair-wise strategy to construct a multi-class classifier. In addition, they use a class splitting technique to balance the class distribution. Since there are too many samples of the class "O," they split class "O" into several subclasses by combining the class "O" and the part-of-speech tags, such as "O-NN," "O-JJ" and etc. Their system achieves *F*-measure of 54.4 on the GENIA V1.1.

Certainly, it is difficult to compare the various models because of the different experimental settings. Since [6,8,5] use the same class and corpus, we make a rough comparison among them. The results show that the HMM and the SVM outperform the decision tree and the performance of the HMM and the SVM are almost equivalent. The results also show that how to capture the useful information for domain-specific named entities and how to integrate them effectively in the model is crucial. In this respect, [8] only use the surface word and the character information, which may not be adequate for coping with the complicated biomedical named entities.

7. Conclusion

In the paper, we introduce the enhancement of a HMM-based biomedical named entity recognizer by studying various special phenomena, such as abbreviations, cascaded named entities and etc. We integrate rich features, such as the orthographic, morphological, part of speech and semantic information. In addition, we present an abbreviation recognition method to recognize the abbreviation according to its full form. We also present a post-processing rule-based cascaded recognition approach and an HMM-based cascaded recognition approach to extend the biomedical named entity recognition. To our best knowledge, our work is the first research work to cope with the cascaded phenomena in the biomedical domain. Based on the rich features and the methods, our system is successfully adapted to the biomedical domain and achieves significantly better performance than the previous best published system. The limitation of our method is that some complicated constructions, such as "and/or," may not be effectively handled. In the near future, further explorations can be made on the complicated constructions in the biomedical documents. One possible way is to develop some effective patterns for the conjunction and disjunction construction. In addition, existing public resources and databases can be integrated in our system.

References

[1] Zhou GD, Su J. Named Entity Recognition using an HMM-based Chunk Tagger. In: Proceedings of the 40th annual meeting of the association for computational linguistics (ACL); 2002. p. 473–80.
[2] Bikel M, Schwartz Danie R, Weischedel Ralph M. An algorithm that learns what's in a name. In: Proceedings of machine learning; 1999 [Special Issue on NLP].
[3] Borthwick A. A maximum entropy approach to named entity recognition. Ph.D. thesis. New York University.
[4] Kazama J, Makino T, Ohta Y, Tsujii J. Tuning support vector machines for biomedical named entity recognition. In: Proceedings of the workshop on natural language processing in the biomedical domain; 2002 (at ACL'2002). p. 1–8.
[5] Takeuchi K, Collier N. Use of support vector machines in extended named entity recognition. In: Proceedings of the sixth conference on natural language learning (CONLL 2002); 2002. p. 119–25.
[6] Nobata C, Collier N, Tsujii J. Automatic term identification and classification in biology texts. In: Proceedings of the 5th NLPRS; 1999. p. 369–74.
[7] Nobata C, Collier N, Tsujii J. Comparison between tagged corpora for the named entity task. In: Proceedings of the workshop on comparing corpora (at ACL'2000); 2000. p. 20–7.
[8] Collier N, Nobata C, Tsujii J. Extracting the names of genes and gene products with a hidden Markov model. In: Proceedings of COLING 2000; 2000. p. 201–7.
[9] Gaizauskas R, Demetriou G Humphreys K. Term recognition and classification in biological science journal articles. In: Proceedings of the computational terminology for medical and biological applications workshop of the 2nd international conference on NLP; 2000. p. 37–44.
[10] Fukuda K, Tsunoda T, Tamura A, Takagi T. Toward information extraction: identifying protein names from biological papers. In: Proceedings of the Pacific Symposium on Biocomputing'98 (PSB'98); 1998. p. 707–18.
[11] Rindflesch TC, Tanabe L, Weinstein JW, Hunter L. EDGAR: extraction of drugs, genes and relations from the biomedical literature. In: Proceedings of the pacific symposium on biocomputing 2000 (PSB'2000); 2000. p. 517–28.
[12] Proux D, Rechenmann F, Julliard L, Pillet V, Jacq B. Detecting gene symbols and names in biological texts: a first step toward pertinent information extraction. In: Proceedings of genome inform ser workshop genome inform; 1998. p. 72–80.
[13] Tanabe L, Wilbur WJ. Tagging gene and protein names in biomedical text. Bioinformatics 2002;18(8):1124–32.
[14] Thomas J, Milward D, Ouzounis C, Pulman S, Carroll M. Automatic extraction of protein interactions from scientific abstracts. In: Proceedings of the pacific symposium on biocomputing'2000 (PSB'2000). Hawaii; 2000. p. 541–51.
[15] Sekimizu T, Park H, Tsujii J. Identifying the interaction between genes and gene products based on frequently seen verbs in medline abstracts. In: Proceedings of genome informatics. Universal Academy Press; 1998.
[16] Rabiner Lawrence R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In: Proceedings of the IEEE: vol. 77. No. 2; 1989. p. 257–86.

[17] Viterbi AJ. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. In: Proceedings of IEEE transactions on information theory; 1967. p. 260–9.
[18] Schwartz AS, Hearst MA. A simple algorithm for identifying abbreviation definitions in biomedical text. In: Proceedings of the pacific symposium on biocomputing (PSB 2003) Kauai; 2003.
[19] Ohta T, Tateisi Y, Kim J, Mima H, Tsujii J. The GENIA corpus: an annotated research abstract corpus in molecular biology domain. In: Proceedings of HLT 2002; 2002.
[20] Yu H, Hripcsak G, Friedman C. Mapping abbreviations to full forms in electronic articles. J Am Med Informat Assoc 2002;9(3):262–72.