Computational Linguistics and Chinese Language Processing
Vol. 9, No.1 , February 2004, pp. 65-82

Mencius: A Chinese Named Entity Recognizer Using the Maximum Entropy-based Hybrid Model

Tzong-Han Tsai[*+], Shih-Hung Wu[+], Cheng-Wei Lee[+],

Cheng-Wei Shih[+], and Wen-Lian Hsu[+]

Abstract

This paper presents a Chinese named entity recognizer (NER): Mencius. It aims to address Chinese NER problems by combining the advantages of rule-based and machine learning (ML) based NER systems. Rule-based NER systems can explicitly encode human comprehension and can be tuned conveniently, while ML-based systems are robust, portable and inexpensive to develop. Our hybrid system incorporates a rule-based knowledge representation and template-matching tool, called InfoMap [Wu *et al.* 2002], into a maximum entropy (ME) framework. Named entities are represented in InfoMap as templates, which serve as ME features in Mencius. These features are edited manually, and their weights are estimated by the ME framework according to the training data. To understand how word segmentation might influence Chinese NER and the differences between a pure template-based method and our hybrid method, we configure Mencius using four distinct settings. The F-Measures of person names (PER), location names (LOC) and organization names (ORG) of the best configuration in our experiment were respectively 94.3%, 77.8% and 75.3%. From comparing the experiment results obtained using these configurations reveals that hybrid NER Systems always perform better performance in identifying person names. On the other hand, they have a little difficulty identifying location and organization names. Furthermore, using a word segmentation module improves the performance of pure Template-based NER Systems, but, it has little effect on hybrid NER systems.

[*] Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.
E-mail: d90013@csie.ntu.edu.tw
[+] Institute of Information Science, Academia Sinica., Taipei, Taiwan, R.O.C.
E-mail: {thtsai, shwu, aska, dapi, hsu}@iis.sinica.edu.tw

1. Introduction

Information Extraction (IE) is the task of extracting information of interest from unconstrained text. IE involves two main tasks: the recognition of named entities, and the recognition of the relationships among these named entities. Named Entity Recognition (NER) involves the identification of proper names in text and classification of them into different types of named entities (e.g., persons, organizations, locations). NER is important not only in IE [Grishman 2002] but also in lexical acquisition for the development of robust NLP systems [Coates-Stephens 1992]. Moreover, NER has proven useful for tasks such as document indexing and the maintenance of databases containing identified named entities.

During the last decade, NER has drawn much attention at Message Understanding Conferences (MUC) [Chinchor 1995a][Chinchor 1998a]. Both rule-based and machine learning NER systems have had some success. Traditional rule-based approaches have used manually constructed finite state patterns, which match text against a sequence of words. Such systems (like the University of Edinburgh's LTG [Mikheev *et al*. 1998]) do not need very much training data and can encode expert human knowledge. However, rule-based approaches lack robustness and portability. Each new source of text requires significant tweaking of the rules to maintain optimal performance, and the maintenance costs can be quite steep.

Another popular approach in NER is machine-learning (ML). ML is attractive in that it is more portable and less expensive to maintain. Representative ML approaches used in NER are HMM (BBN's IdentiFinder in [Miller *et al*. 1998][Bikel *et al*. 1999] and Maximum Entropy (ME) (New York University's MEME in [Borthwick *et al*. 1998][Borthwick 1999]). However, ML systems are relatively inexpensive to develop, and the outputs of these systems are difficult to interpret. In addition, it is difficult to improve the system performance through error analysis. The performance of an ML system can be very poor when the amount of training data is insufficient. Furthermore, the performance of ML systems is worse than that of rule-based ones by about 2%, as revealed at MUC-6 [Chinchor 1995b] and MUC-7 [Chinchor 1998b]. This might be due to the fact that current ML approaches can not capture non-parametric factors as effectively as human experts who handcraft the rules. Nonetheless, ML approaches do provide important statistical information that is unattainable by human experts. Currently, the F-measures of English rule-based and ML NER systems are in the range of 85% ~ 94%, based on MUC-7 data [Chinchor 1998c]. This is higher than the average performance of Chinese NER systems, which ranges from 79% to 86% [Chinchor 1998].

In this paper, we address the problem of Chinese NER. In Chinese sentences, there are no spaces between words, no capital letters to denote proper names, no sentence breaks, and, worst of all, no standard definition of “words.” As a result, word boundaries cannot, at times, be discerned without a context. In addition, the length of a named entity is longer on average than

that of an English one; thus, the complexity of a Chinese NER system is greater.

Previous works [Chen *et al*. 1998] [Yu *et al*. 1998] [Sun *et al*., 2002] on Chinese NER have relied on the word segmentation module. However, an error in the word segmentation step might lead to errors in NER results. Therefore, we want to compare the results of NER with/without performing word segmentation. Without word segmentation, a character-based tagger is used, which treats each character as a token and combines the tagged outcomes of contiguous characters to form an NER output. With word segmentation, we treat each word or character as a token, and combine the tagged outcomes of contiguous tokens to form an NER output.

Borthwick [1999] used an ME framework to integrate many NLP resources, including previous systems such as Proteus, a POS tagger. Mencius, the Chinese named entity recognizer presented here, incorporates a rule-based knowledge representation and a template-matching tool, called InfoMap [Wu *et al.* 2002], into a maximum entropy (ME) framework. Named entities are represented in InfoMap as templates, which serve as ME features in Mencius. These features are edited manually, and their weights are estimated by means of the ME framework according to the training data.

This paper is organized as follows. Section 2 provides the ME-based framework for NER. Section 3 describes features and how they are represented in our knowledge representation system, InfoMap. The data set and experimental results are discussed in section 4. Section 5 gives our conclusions and possible extensions of the current work.

2. Maximum Entropy-Based NER Framework

For our purpose, we regard each character as a token. Consider a test corpus and a set of n named entity categories. Since a named entity can have more than one token, we associate the following two tags with each category x: *x_begin* and *x_continue*. In addition, we use the tag *unknown* to indicate that a token is not part of a named entity. The NER problem can then be rephrased as the problem of assigning one of $2n + 1$ tags to each token. In Mencius, there are 3 named entity categories and 7 tags: *person_begin*, *person_continue*, *location_begin*, *location_continue*, *organization_begin*, *organization_continue* and *unknown*. For example, the phrase [李 遠 哲 在 高 雄 市] (Lee, Yuan Tseh in Kaohsiung City) could be tagged as *_begin*, [*person person_continue*, *person_continue*, *unknown*, *location_begin*, *location_continue*, *location_continue*].

2.1 Maximum Entropy

ME is a flexible statistical model which assigns an *outcome* for each token based on its *history*

and *features*. Outcome space is comprised of the seven Mencius tags for an ME formulation of NER. ME computes the probability $p(o|h)$ for any o from the space of all possible outcomes O, and for every h from the space of all possible histories H. A *history* is composed of all the conditioning data that enable one to assign probabilities to the space of outcomes. In NER, *history* can be viewed as consisting of the all information derivable from the test corpus relavant to the current token.

The computation of $p(o|h)$ in ME depends on a set of binary-valued *features*, which are helpful in making a prediction about the outcome. For instance, one of our features is as follows: when the current character is a known surname, it is likely to be the leading character of a person name. More formally, we can represent this feature as

$$f(h,o) = \begin{cases} 1: \text{if Current-Char-Surname(h)} = \text{true and } o = person\_begin \\ 0: \text{else} \end{cases} \quad (1)$$

Here, *Current-Char-Surname(h)* is a binary function that returns the value *true* if the *current character* of the history h is in the surname list.

Given a set of features and a training corpus, the ME estimation process produces a model in which every feature f_i has a weight α_i. This allows us to compute the conditional probability as follows [Berger *et al*. 1996]:

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

Intuitively, the probability is the multiplication of the weights of active features (i.e., those $f_i(h,o) = 1$). The weight α_i is estimated by means of a procedure called Generalized Iterative Scaling (GIS) [Darroch *et al*. 1972]. This is an iterative method that improves estimation of the weights at each iteration. The ME estimation technique guarantees that for every feature f_i, the expected value of α_i equals the empirical expectation of α_i in the training corpus.

As Borthwick [1999] remarked, ME allows the modeler to concentrate on finding the features that characterize the problem while letting the ME estimation routine deal with assigning relative weights to the features.

2.2 Decoding

After an ME model has been trained and the proper weight α_i has been assigned to each feature f_i, decoding (i.e., *marking up*) a new piece of text becomes a simple task. First, Mencius tokenizes the text and preprocesses the testing sentence. Then for each token, it checks which

features are active and combines the α_i of the active features according to equation 2. Finally, a Viterbi search is run to find the highest probability path through the lattice of conditional probabilities that does not produce any invalid tag sequences (for instance, the sequence [*person_begin, location_continue*] is invalid). Further details on Viterbi search can be found in [Viterbi 1967].

3. Features

We divide features that can be used to recognize named entities into four categories according to whether they are external or not and whether they are category dependent or not. McDonald defined internal and external features in [McDonald 1996]. Internal evidence is found within the entity, while external evidence is gathered from its context. We use category-independent features to distinguish named entities from non-named entities (e.g., first-character-of--a-sentence, capital-letter, out-of-vocabulary), and use category-dependent features to distinguish between different named entity categories (for example, surname and given name lists are used to recognize person names). However, to simplify our design, we only use internal features that are category-dependent in this paper.

3.1 InfoMap – Our Knowledge Representation System

To the calculate values of location features and organization features, Mencius uses InfoMap. InfoMap is our knowledge representation and template matching tool, which represents location or organization names as templates. An input string (sentence) is first matched to one or more location or organization templates by InfoMap and then passed to Mencius; there, it is assigned feature values which further distinguish which named entity category it falls into.

3.1.1 Knowledge Representation Scheme in InfoMap

InfoMap is a hierarchical knowledge representation scheme, consisting of several domains, each with a tree-like taxonomy. The basic units of information in InfoMap are called generic nodes, which represent concepts, and function nodes, which represent the relationships among the generic nodes of one specific domain. In addition, generic nodes can also contain cross references to other nodes to avoid needless repetition.

In Mencius, we apply the geographical taxonomy of InfoMap called GeoMap. Our location and organization templates refer to generic nodes in Geomap. As shown in Figure 1, GeoMap has three sub-domains: World, Mainland China, and Taiwan. Under the sub-domain Taiwan, there are four attributes: Cities, Parks, Counties and City Districts. Moreover, these attributes can be further divided; for example, Counties can be divided into individual counties:

Taipei County, Taoyuan County, etc. In InfoMap, we refer to generic nodes (or concept node) by means of paths. A path of generic nodes consists of all the node names from the root of the domain to the specific generic node, where function nodes are omitted. The node names are separated by periods. For example, the path for the "Taipei County" node is "GeoMap.Counties.Taipei County."

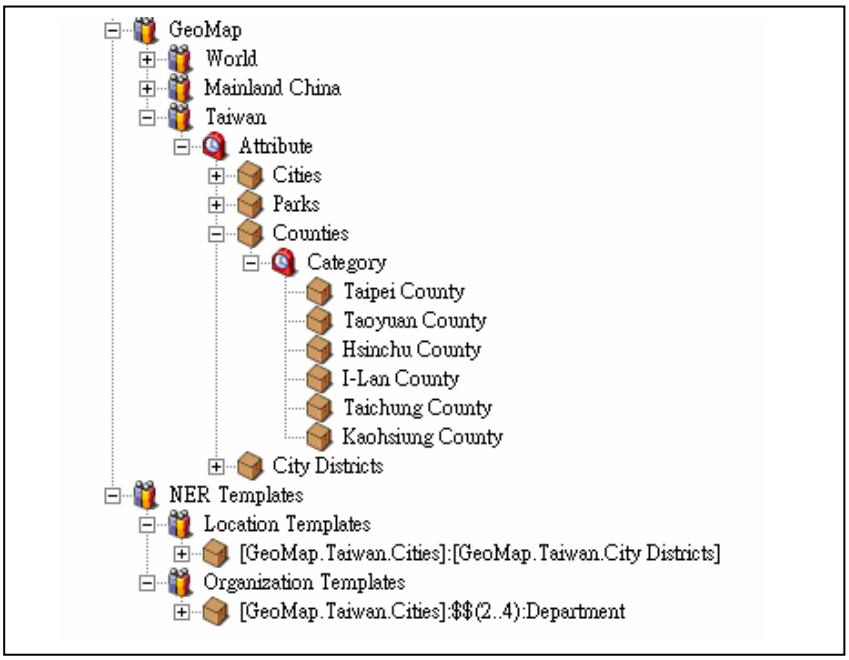

Figure 1. A partial view of GeoMap.

3.1.2 InfoMap Templates

In InfoMap, text templates are stored in generic nodes. Templates can consist of character strings, wildcards (see $$ in Table 1), and references to other generic nodes in InfoMap. For example, the template [通用地理.台灣.縣]:$$(2..4):局 ([GeoMap. Taiwan. Counties]: $$(2..4):Department) can be used to recognize county level governmental departments in Taiwan. The syntax used in InfoMap templates are shown in Table 1. The first part of our sample template shown above (enclosed by "[]") is a path that refers to the generic node "Counties." The second element is a wildcard, ($$) which must be 2 to 4 characters in length. The third element is a specified character "局" (Department).

Table 1. InfoMap template syntax.

Symbol	Semantics	Example Template	Sample Matching String
:	Concatenate two strings	A:B	AB
$$(m..n)	Wildcards (the number of characters can be from m to n; both m and n have to be non-negative integers)	A:$$(1..2):B	ACB, ADDB, ACDB
[p]	A path to a generic node	[GeoMap.Taiwan.Counties]	Taipei County, Taoyuan County, Hsinchu County, etc.

3.2 Category-Dependent Internal Features

Recall that category-dependent features are used to distinguish among different named entity categories.

3.2.1 Features for Recognizing Person Names

Mencius only deals with a surname plus a first name (usually composed of two characters), for example, 陳水扁 (Chen Shui-bian). There are various other ways to identify a person in a sentence, such as 陳先生 (Mr. Chen) and老陳 (Old Chen), which have not been incorporated into the current system. Furthermore, we do not target transliterated names, such as 布希 (Bush), since they do not follow Chinese name composition rules. We use a table of frequently occurring names to process our candidate test data. If a character and its context (history) correspond to a feature condition, the value of the current character for that feature will be set to 1. Feature conditions, examples and explanations for each feature are shown in Table 2. In the feature condition column, c_{-1}, c_0, and c_1 represent the preceding character, the current character, and the following character, respectively.

Current-Char-Person-Surname: This feature is set to 1 if $c_0c_1c_2$ or c_0c_1 is in the person name database. For example, in the case of $c_0c_1c_2$ = 陳水扁, the feature Current-Char-Person-Surname for 陳 is active since c_0 and its following characters c_1c_2 satisfy the feature condition.

Current-Char-Person-Given-Name: This feature is set to 1 if $c_{-2}c_{-1}c_0$, $c_{-1}c_0$, or $c_{-1}c_0c_1$ is in the person name database.

Current-Char-Surname: This feature is set to 1 if c_0 is in the top 300 popular surname list.

Table 2. Person features.

Feature	Feature Conditions	Example	Explanation
Current-Char-Person-Surname	$c_0c_1c_2$ or c_0c_1 is in the name list	“陳”水扁, “連”戰	Probably the first character of a person name
Current-Char-Person-Given-Name	$c_{-2}c_{-1}c_0$ or $c_{-1}c_0$ or $c_{-1}c_0c_1$ is in the name list	陳“水”扁, 陳水“扁”, 連“戰”	Probably the second or third character of a person name
Current-Char-Surname	c_0 is in the surname list	“陳”, “林”, “李”	Probably a surname
Current-Char-Given-Name	c_0c_1 or $c_{-1}c_0$ is in the given name list	黃“其”聖, 黃其“聖”	Probably part of a popular given name
Current-Char-Freq-Given-Name-Character	Both c_0, c_1 or c_{-1}, c_1 is in the frequent given name character list	羅“方”全, 羅方“全”	Probably a given name character
Current-Char-Speaking-Verb	c_0 or c_0c_1 or $c_{-1}c_0$ is in the list of verbs indicating speech	“說”, “表”示, 表“示”	Probably part of a verb indicating speech (ex: John said he was tired)
Current-Char-Title	c_0 or c_0c_1 or $c_{-1}c_0$ is in the title list	“先”生, 先“生”	Probably part of a title

Current-Char-Given-Name: This feature is set to 1 if c_0c_1 or $c_{-1}c_0$ is in the given name database.

Current-Char-Freq-Given-Name-Character: (c_0 and c_1) or (c_{-1} and c_0) is in the frequently given name character list

Current-Char-Speaking-Verb: c_0 or c_0c_1 or $c_{-1}c_0$ is in the speaking verb list. This feature distinguishes a trigram containing a speaking verb, such as 陳沖說 (Chen Chong said), from a real person name.

Current-Char-Title: c_0 or c_0c_1 or $c_{-1}c_0$ is in the title list. This feature distinguishes a trigram containing a title, such as 陳先生 (Mr. Chen), from a real person name.

3.2.2 Features for Recognizing Location Names

In general, locations are divided into four types: administrative division, public area (park, airport, or port), landmark (road, road section, cross section or address), and landform (mountain, river, sea, or ocean). An administrative division name usually contains one or more

location names in a hierarchical order, such as 安大略省多倫多市 (Toronto, Ontario). A public area name is composed of a Region-Name and a Place-Name. However, the Region-Name is usually omitted from news content if it was previously mentioned. For example, 倫敦海德公園 (Hyde Park, London) contains the Region-Name 倫敦 (London) and the Place-Name 海德公園 (Hyde Park). But "Hyde Park, London" is usually abbreviated as "Hyde Park" within a report. The same rule can be applied to landmark names. A landmark name includes a Region-Name and a Position-Name. In a news article, the Region-Name can be omitted if the Place-Name has been mentioned previously. For example, 溫哥華市羅伯遜街五號 (No. 5, Robson St., Vancouver City) will be stated as 羅伯遜街五號 (No. 5, Robson St.) later in the report.

In Mencius, we build templates to recognize three types of location names. Our administrative division templates contain more than one set of location names in a hierarchical order. For example, the template, [通用地理.台灣.市]:[通用地理.台灣.各市行政區] ([GeoMap.Taiwan.Cities]:[GeoMap.Taiwan.City Districts]) can be used to recognize all city districts in Taiwan. In addition, public area templates contain one set of location names and a set of Place-Name. For example, [通用地理.台灣.市]:[通用地理.台灣.公園] ([GeoMap.Taiwan.Cities]:[GeoMap.Taiwan.Parks]) can be used to recognize all city parks in Taiwan. Landmark templates are built in the same way.For example, [通用地理.台灣.市]:$$(2..4):路 ([GeoMap.Taiwan.Cities]:$$(2..4):Road) can be used to recognize roads in Taiwan.

Two features are associated with each InfoMap template category x (e.g., location and organization). The first is Current-Char-InfoMap-x-Begin, which is set to 1 for the first character of a matched string and set to 0 for the remaining characters. The other is Current-Char-InfoMap-x-Continue, which is set to 1 for all the characters of matched string except for the first character and set to 0 for the first character. The intuition behind this is as follows: InfoMap can be used to help ME detect which character in a sentence is the first character of the location name and which characters are the remaining characters of a location name. That is, Current-Char-InfoMap-x-Begin is helpful for determining which character should be tagged as *x_begin*, while Current-Char-InfoMap-x-Continue is helpful for determining which character should be tagged as *x_continue* if we build an InfoMap template for that category x. The two features associated with x category are shown below:

$$f(h,o)=\begin{cases}1: \text{if Current-Char-InfoMap-x-Begin = true and } o = x\_begin \\ 0: \text{else}\end{cases} \quad (3)$$

$$f(h,o)=\begin{cases}1: \text{if Current-Char-InfoMap-x-Continue= true and } o = x\_continue \\ 0: \text{else}\end{cases} \quad (4)$$

When recognizing a location name in a sentence, we test if any location templates match the sentence. If several matched templates overlap, we select the longest matched one. As mentioned above, the feature Current-Character-InfoMap-Location-Begin of the first character of the matched string is set to 1 while the feature Current-Character-InfoMap-Location--Continue of the remaining characters of the matched string is set to 1. Table 3 shows the necessary conditions for each organization feature and gives examples of matched data.

Table 3. Location features.

Feature	Feature Conditions	Example	Explanations
Current-Char-InfoMap-Location-Begin	c_0~c_{n-1} matches an InfoMap location template, where the character length of the template is n	“台”北縣板橋市	Probably the leading character of a location name.
Current-Char-InfoMap-Location-Continue	c_a…c_0….c_b matches an InfoMap location template, where a is a negative integer and b is a non-negative integer	台”北”縣板橋市	Probably a continuing character of a location name.

3.2.3 Features for Recognizing Organization Names

Organizations include named corporate, governmental, or other organizational entities. The difficulty in recognizing an organization name is that it usually begins with a location name, such as 台北市地檢署 (Taipei District Public Prosecutors Office). Therefore, traditional machine learning NER systems can only identify the location part rather than the full organization name. For example, the system only extracts 台北市 (Taipei City) from 台北市 SOGO 百貨週年慶 (Taipei SOGO Department Store Anniversary) rather than 台北市 SOGO 百貨 (Taipei SOGO Department Store). According to our analysis of the structure of Chinese organization names, they mostly end with a specific keyword or begin with a location name. Therefore, we use those keywords and location names as the boundary markers of organization names. Based on our observation, we categorize organization names into four types according to their boundary markers.

Type I: With left and right boundary markers

The organization names in this category begin with by one or more geographical names and

ended by an organization keyword. For example, 台北市 (Taipei City) is the left boundary marker of 台北市捷運公司 (Taipei City Rapid Transit Corporation), while an organization keyword, 公司 (Corporation), is the right boundary marker.

Type II: With a left boundary marker

The organization names in this category begin with by one or more than one geographical names, but the organization keyword (e.g., 公司 (Corporation)) is omitted. For example, 台灣捷安特 (Giant Taiwan) only contains the left boundary 台灣 (Taiwan).

Type III: With a right boundary marker

The organization names in this category end with an organization keyword. For example, 捷安特公司 (Giant Corporation) only contains the right boundary 公司 (Corporation).

Table 4. Organization features.

Feature	Feature Conditions	Example	Explanations
Current-Char-InfoMap-Organization-Begin	c_0~c_{n-1} matches an InfoMap organization template, where the character length of the template is n	“台”北市捷運公司	Probably the leading character of an organization name
Current-Char-InfoMap-Organization-Continue	c_a…c_0….c_b matches an InfoMap organization template, where a is a negative integer and b is a non-negative integer	台”北”市捷運公司	Probably the leading character of an organization name
Current-Char-Organization-Keyword	c_0 or c_0c_1 or $c_{-1}c_0$ is in the organization keyword list	“公”司, 公 “司”	Probably part of an organization keyword

Type IV: No boundary marker

In this category, both left and right boundaries as above mentioned are omitted, for example, 捷安特 (Giant). The organization names in this category are usually in abbreviated form.

In Mencius, we build templates for recognizing Type I organization names. Each organization template begins with a location name in GeoMap and ends with an organization keyword. For example, we can build [通用地理.台灣.市]:$$(2..4):局([GeoMap.Taiwan.Cities]:

$$(2..4):Department) to recognize county level government departments in Taiwan. However, in Types II, III, and IV, organization names cannot be recognized by templates. Therefore, the maximum entropy model uses features of characters (from c_{-2} to $c_{2)}$, tags (from t_{-2} to t_2), and organization keywords, e.g., 公司 (Corporation), to find the most likely tag sequences and recognize them.

When a string matches an organization template, the feature Current-Character- InfoMap--Organization-Start of the first character is set to 1. In addition, the feature Current--Character-InfoMap-Organization-Continue of the remaining characters is set to 1. The necessary conditions for each organization feature and examples of matched data are shown in Table 4. These features are helpful for recognizing organization names.

4. Experiments

4.1 Data Sets

For Chinese NER, the most famous corpus is MET-2 [6]. There are two main differences between our corpus and MET-2: the number of domains and the amount of data. First, MET-2 contains only one domain (Accident), while our corpus, which was collected from the online United Daily News in December 2002 (http://www.udn.com.tw), contains six domains: Local News, Social Affairs, Investment, Politics, Headline News and Business, which provide a greater variety of organization names than a single domain corpus can. The full location names and organization names are comparatively longer, and our corpus contains more location names and addresses at the county level. Therefore, the patterns of location names and organization names are more complex in our corpus.

Secondly, our corpus is much larger than MET2, which contains 174 Chinese PER, 750 LOC, and 377 ORG. Our corpus contains 1,242 Chinese PER, 954 LOC, and 1,147 ORG in 10,000 sentences (about 126,872 Chinese characters). The statistics of our data are shown in Table 5.

Table 5. Statistics of the data Set

Domain	Number of Named Entities			Size (in characters)
	PER	LOC	ORG	
Local News	84	139	97	11835
Social Affairs	310	287	354	37719
Investment	20	63	33	14397
Politics	419	209	233	17168
Headline News	267	70	243	19938
Business	142	186	187	25815
Total	1242	954	1147	126872

4.2 Experimental Results

To understand how word segmentation might influence Chinese NER and the differences between a pure template-based method and our hybrid method, we configure Mencius using the following four settings: (1) Template-based with Char-based Tokenization (TC), (2) Template-based with Word-based Tokenization (TW), (3) Hybrid with Char-based Tokenization (HC), and (4) Hybrid with Word-based Tokenization (HW). Following the standard 10-fold cross-validation method, we tested Mencius with each configuration using the data set mentioned in section 4.1. The following subsections provide details about each configuration and the results obtained.

4.2.1 Template-based with Char-based Tokenization (TC)

In this experiment, we regarded each character as a token, and used a person name list and InfoMap templates to recognize all named entities. The number of lexicons in the person name lists and gazetteers was 32000. As shown in Table 6, the obtained F-Measures of PER, LOC and ORG were 76.2%, 75.4% and 75.1%, respectively.

Table 6. Performance of the Template-based System with Char-based Tokenization.

NE	P(%)	R(%)	F(%)
PER	64.77	92.59	76.22
LOC	76.41	74.42	75.40
ORG	85.60	66.93	75.12
Total	72.95	78.62	75.67

4.2.2 Template-based with Word-based Tokenization (TW)

In this experiment, we used a word segmentation module based on the 100,000-word CKIP Traditional Chinese dictionary to split sentences into tokens. This module combines forward and backward longest matching algorithms in the following way: if the segmentation results of the two algorithms agree in certain substrings, this module outputs tokens in those substrings. While in the part which the segmentation results of the two algorithms differ, this module skips word tokens and only outputs character tokens. In the previous test, 98% of the word tokens were valid words. Then, we used person name lists and InfoMap templates to recognize all the named entities. The number of lexicons in the person name lists and gazetteers was 32,000. As shown in Table 6, the obtained F-Measures of PER, LOC and ORG were 89.0%, 74.1% and 71.6%, respectively.

Table 7. Performance of the Template-based System with Word-based Tokenization.

NE	P(%)	R(%)	F(%)
PER	88.69	89.32	89.00
LOC	76.92	71.44	74.08
ORG	85.66	61.44	71.55
Total	84.14	74.70	79.14

4.2.3 Hybrid with Char-based Tokenization (HC)

In this experiment, we regarded each character as a token without performing any word segmentation. We then integrated person name lists, location templates, and organization templates into a Maximum-Entropy-Based framework. As shown in Table 8, the obtained F-Measures of PER, LOC and ORG were 94.3%, 77.8% and 75.3%, respectively.

Table 8. Performance of the Hybrid System with Char-based Tokenization.

NE	P(%)	R(%)	F(%)
PER	96.97	91.71	94.27
LOC	80.96	74.81	77.76
ORG	87.16	66.22	75.26
Total	89.05	78.18	83.26

4.2.4 Hybrid System with Word-based Tokenization (HW)

In this experiment, we used the same word segmentation module described in section 4.2.2 to split sentences into tokens. Then, we integrated person name lists, location templates, and organization templates into a Maximum-Entropy-Based framework. As shown in Table 9, the obtained F-Measures of PER, LOC and ORG were 95.9%, 73.4% and 76.1%, respectively.

Table 9. Performance of the Hybrid System with Word-based Tokenization.

NE	P(%)	R(%)	F(%)
PER	98.74	93.31	95.94
LOC	81.46	66.73	73.36
ORG	87.54	67.29	76.09
Total	90.33	76.66	82.93

4.2.5 Comparisons

TC versus TW

We observed that TW achieved much higher precision than TC in PER. When word segmentation is not performed, some trigrams and quadgrams may falsely appear to be person names. Take the sentence “新古典主義” for example. TC would extract “古典主” as a person

name since “古典主” matches our family-name trigram template. However, in TW, thanks to word segmentation, “古典” and “主義” would be marked as tokens first and would not match the family-name trigram template.

HC versus HW

We observed that HW achieved similar precision to that of HC in all three NE categories. HW also achieved recall rates similar to those achieved by HC with PER and ORG NEs. In the case of PER NEs, this is because the length of person names is 2 to 4 characters. Therefore, a five-character long window (-2 to +2) is sufficient to recognize a person name. As far as recognizing LOC NEs is concerned, HW’s recall rate was worse than HC’s. This is because the word segmentation module marks occupational titles as tokens, for example: “台北市長”. HW cannot extract the LOC NE “台北市” from “台北市長” because it has already been defined as a token. To recognize LOC and ORG NEs, we need higher-level features and more external features. Since Mencius lacks these kinds of features, HW doesn’t achieve significantly better performance than HC.

TC versus HC

We observed that in PER, HC achieved much higher precision than TC, while in LOC and ORG, HC performed slightly better than TC. This is because most of the key features for identifying a person name are close to the person name, or inside the personal name. Take the sentence “立即連絡海鷗直升機” as an example; when we wish to determine whether “連絡海” is a person name, we can see that “立即” seldom appears before a person name, and that “鷗” seldom appears after a person name. In HC, ME can use this information to determine that “連絡海” is not a person name, but to recognize a location name and an organization name, we need wider context and features, such as sentence analysis or shallow parsing. Take “如馬公、七美、望安、蘭嶼、綠島、馬祖和金門等離島爲管制航線” as an example; the two preceding characters are “美”and “、”, and the two following characters are “、”and “蘭”. ME cannot use this information to identify a location name.

TW versus HW

We observed that HW achieved better precision than TW in identifying personal names. This is because in HW, ME can use context information to filter some trigrams and 4 grams, which are not personal names. Take “王 金 平 和 其他 委員” as an example; it matches the double-family-name quadgram template because “王” and “金” are both family names. However, “王金平” is the correct person name. In HW, ME can use the information that “王金平” has appeared in the training corpus and been tagged as a PER NE to identify the person name “王金平” in a sentence. We also observed that HW achieved better recall than TW in identifying person names. This is because in HW, ME can use the information that bigram

personal names are tagged as PER NEs from the training data, but TW cannot because we don't have bigram-person-name templates. In addition, some person names are in the dictionary, so some tokens are person names. Take "陳建仁 的 作爲" as an example. Although the token "陳建仁" cannot match any person name template, in HW, ME can use context information and training data to recognize "陳建仁". To identify location names, ME needs a wider context to detect location names, so HW's recall is worse than TW's. However, ME can filter out some unreasonable trigrams, such as "黃榮村", because it matches a location name template $$(2..3): 村, which represents a village in Taiwan. Therefore, ME achieves bigger precision in identifying location names.

5. Conclusions

In this paper, we have presented a Chinese NER system, called Mencius. We configured Mencius according to the following settings of to analyze the effects of using a Maximum Entropy-based Framework and a word segmentation module: (1) Template-based with Char-based Tokenization (TC), (2) Template-based with Word-based Tokenization (TW), (3) Hybrid with Char-based Tokenization (HC), and (4) Hybrid with Word-based Tokenization (HW). The experimental results showed that whether a character or a word was taken as a token, the hybrid NER System always performed better in identifying person names. However,this had little effect on the identification of location and organization names. This is because the context information around a location name or an organization name is more complex than that around a person name. In addition, using a word segmentation module improved the performance of the pure Template-based NER System. However, it had little effect with the hybrid NER systems. The current version of Mencius lacks sentence parsing templates and shallow parsing tools to handle such complex information. We will add these functions in the future.

References

Berger, A., Della Pietra, S. A., and Della Pietra, V. J., "A maximum entropy approach to natural language processing," Computational Linguistics, vol. 22, pp. 39-71, 1996.

Bikel, D., Schwartz, R., and Weischedel, R., "An Algorithm that Learns What's in a Name," Machine Learning, 1999.

Borthwick, A., Sterling J., Agichtein, E., and Grishman, R., "NYU: Description of the MENE Named Entity System as Used in MUC-7," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Borthwick, A., "A Maximum Entropy Approach to Named Entity Recognition," New York University, 1999.

Chinchor, N., "MUC-6 Named Entity Task Definition (Version 2.1)," presented at the 6th Message Understanding Conference, Columbia, Maryland, 1995.

Chinchor, N., "Statistical Significance of MUC-6 Results," presented at the 6th Message Understanding Conference, Columbia, Maryland, 1995.

Chinchor, N., "MUC-7 Named Entity Task Definition (Version 3.5)," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Chinchor, N., "Statistical Significance of MUC-7 Results," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Chinchor, N., "MUC-7 Test Score Reports for all Participants and all Tasks," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Chen, H. H., Ding, Y. W., Tsai, S. C., and Bian, G. W., "Description of the NTU System Used for MET2," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Coates-Stephens, S., "The Analysis and Acquisition of Proper Names for Robust Text Understanding," in Dept. of Computer Science. London: City University, 1992.

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

Grishman, R., "Information Extraction: Techniques and Challenges," in Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, J. G. Carbonell, Ed. Frascati, Italy: Springer, 1997, pp. 10-26.

McDonald, D., "Internal and External Evidence in the Identification and Semantic Categorization of Proper Names," in Corpus Processing for Lexical Acquisition, J. Pustejovsky, Ed. Cambridge, MA: MIT Press, 1996, pp. 21-39.

Mikheev, A., Grover, C., and Moensk, M., "Description of the LTG System Used for MUC-7," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schwartz, R., Stone, R., and Weischedel, R., "BBN: Description of the SIFT System as Used for MUC-7," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.

Sun, J., Gao, J. F., Zhang, L., Zhou, M., and Huang, C. N., "Chinese Named Entity Identification Using Class-based Language Model," presented at the 19th International Conference on Computational Linguistics,, 2002.

Viterbi, A. J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Transactions on Information Theory, vol. IT, pp. 260-269, 1967.

Wu, S. H., Day, M. Y., Tsai, T. H., and Hsu, W. L., "FAQ-centered Organizational Memory," in Knowledge Management and Organizational Memories, R. Dieng-Kuntz, Ed. Boston: Kluwer Academic Publishers, 2002.

Yu, S. H., Bai, S. H., and Wu, P., "Description of the Kent Ridge Digital Labs System Used for MUC-7," presented at the 7th Message Understanding Conference, Fairfax, Virginia, 1998.