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

Tzong-Han Tsai[*†], Shih-Hung Wu[†], Cheng-Wei Lee[†],
Cheng-Wei Shih[†], and Wen-Lian Hsu[†]

[*]Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan, R.O.C.
d90013@csie.ntu.edu.tw

[†]Institute of Information Science, Academia Sinica.
Taipei, Taiwan, R.O.C.
{thtsai, shwu, aska, dapi, hsu}@iis.sinica.edu.tw

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, InfoMap [1], 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 avoid the errors caused by word segmentation, we model the NER problem as a character-based tagging problem. In our experiments, Mencius outperforms both pure rule-based NER systems. The F-Measures of person names (PER), location names (LOC) and organization names (ORG) in the experiment are respectively 94.3%, 77.8% and 75.3%. We also compared the NER results with/without word segmentation and found slight differences.

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 their classification into different types of named entities (e.g., persons, organizations, locations). NER is not only important in IE [3] but also in lexical acquisition for the development of robust NLP systems [4]. Moreover, NER has proven fruitful for tasks such as documents indexing, and maintenance of databases containing identified named entities.

During the last decade, NER has drawn much attention at Message Understanding Conferences (MUC) [5] [6]. Both rule-based and machine learning NER systems have had some success. Previous rule-based approaches have used manually constructed finite state patterns, which match text against a sequence of words. Such system (like University of Edinburgh's LTG [7]) do not need too much training data and can encode expert human knowledge. However, rule-based approaches lack robustness and portability. Each new source of text requires a significant tweaking of the rules to maintain optimal performance; the maintenance costs can be quite steep.

Another popular approach in NER is machine-learning (ML). ML is more attractive in that it is more portable and less expensive to maintain. The representative ML approaches used in NER are HMM (BBN's IdentiFinder in [8, 9] and Maximum Entropy (ME) (New York Univ.'s MEME in [10] [11]). Although ML systems are relatively inexpensive to develop, the outputs of these systems are difficult to interpret. As well, it is difficult to improve the system performance through error analysis. The performance of a ML system can be very poor when training data is insufficient. Furthermore, the performance of ML systems is worse than that of rule-based ones by about 2% as witnessed in MUC-6 [12] and MUC-7 [13]. This might be due to the fact that current ML approaches can capture non-parametric factors less effectively than human experts who handcraft the rules. Nonetheless, ML approaches do provide important statistical information that is unattainable by human experts. Currently, the F-measure in English rule-based and ML NER systems are 85% ~ 94% on MUC-7 data [14]. This is higher than the average performance of Chinese NER systems, which ranges from 79% to 86% [14].

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 or sentence breaks, and, worst of all, no standard definition of “words”. As a result, word boundaries cannot, at times, be discerned without context. As well, the length of a named entity is longer on average than an English one, thus, the complexity of a Chinese NER system is greater.

Previous works [15] [16] [2] on Chinese NER rely 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 result with/without word segmentation. Without word segmentation, we use a character-based tagger, treat each character as a token, and combine the tagged outcomes of continuing characters to form an NER output. With word segmentation, we treat each word or character as a token, and combine the tagged outcomes of continuing tokens to form an NER output.

Borthwick [11] uses an ME framework to integrate many NLP resources, including previous systems such as Proteus, a POS tagger. In this paper, Mencius incorporates a rule-based knowledge representation and template-matching tool, InfoMap [1], 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.

This paper is organized as follows. Section 2 provides the ME-based framework for NER. Section 3 describes features and how to represent them 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 two tags to 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 [*person_begin*, *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 all the conditioning data that enables one to assign probabilities to the space of outcomes. In NER, *history* could be viewed as all information derivable from the test corpus relative to the current token.

The computation of $p(o|h)$ in ME depends on a set of binary-valued *features*, which are helpful in making a prediction about the outcome. For instance, one of our features is: 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

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

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 [17].

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

Intuitively, the probability is the multiplication of weights of active features (i.e. those f_i $(h,o) = 1$). The weight α_i is estimated by a procedure called Generalized Iterative Scaling (GIS) [18]. This is an iterative method that improves the 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 [11] 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 having trained an ME model and assigned the proper weight α_i 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 we check which features are active and combine 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 the Viterbi search can be found in [19].

3 Features

We divide features that can be used to recognize named entities into four categories according to whether they are external and whether they are category dependent. McDonald defined internal and external features in [20]. The internal evidence is found within the entity, while the 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 category-dependent features to distinguish between different named entity categories (for example, surname and given name lists are used for recognizing 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 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 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. 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 separates into individual counties: Taipei County, Taoyuan County, etc. In InfoMap, we refer to generic nodes (or concept node) by paths. A path of generic nodes consists of all node names from the root of the domain to the specific generic node, in which 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."

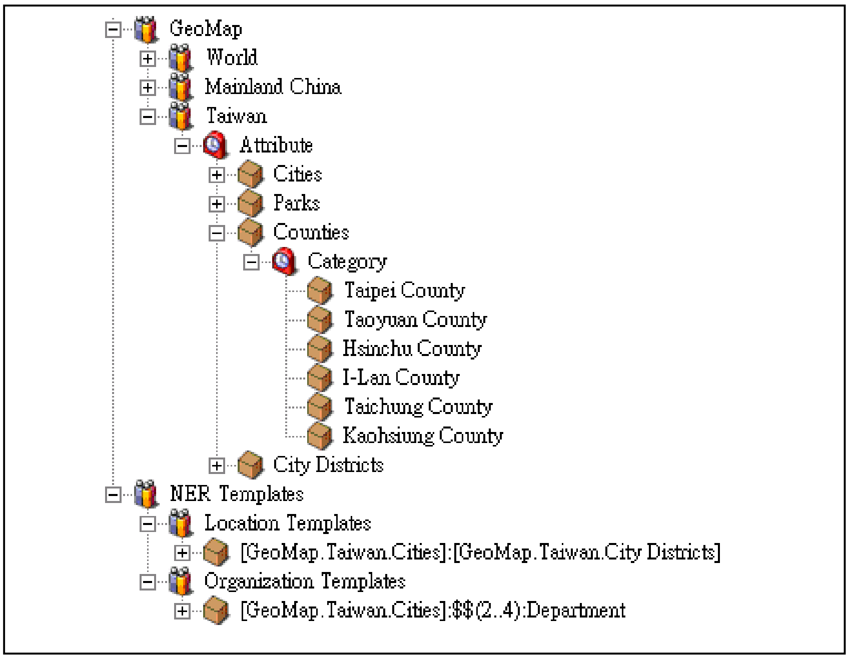

Figure 1. A partial view of GeoMap

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]:$$(2..4):Department), can be used to recognize county level governmental departments in Taiwan. The syntax used in InfoMap templates are shown in Table 1. The first part of our sample template above (enclosed by "[]") is a path that refers to the generic node "Counties". The second element is a wildcard ($$) 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
$$(m..n)	Wildcards (number of characters can be from m to n; both m and n have to be non-negative integers)	A:$$(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 surname plus first name (usually with two characters), for example, 陳水扁 (Chen Shui-bian). There are various way to express 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 conditions column, c_{-1}, c_0, and c_1 represent the preceding character, the current character, and the following character respectively.

Table 2. Person Features

Feature	Feature Conditions	Example	Explanation
Current-Char-Person-Surname	$c_0c_1c_2$ or c_0c_1 are 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 $c_{-1}c_0c_1$ are 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 are in the surname list	“陳”, “林”, “李”	Probably a surname
Current-Char-Given-Name	c_0c_1 or $c_{-1}c_0$ are 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 are 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 $c_{-1}c_0$ are 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 $c_{-1}c_0$ are in the title list	“先”生, 先 “生”	Probably part of a title

Current-Char-Person-Surname: This feature is set to 1 if $c_0c_1c_2$ or c_0c_1 are in the person name database. For example, in the case $c_0c_1c_2$ = 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 $c_{-1}c_0c_1$ are 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.

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

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

Current-Char-Speaking-Verb: c_0 or c_0c_1 or $c_{-1}c_0$ are 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 $c_{-1}c_0$ are 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 than one location names in 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 in news content if it was previously mentioned. For example, 倫敦海德公園 (Hyde Park, London) contains a Region-Name 倫敦 (London) and a Place-Name 海德公園 (Hyde Park). But “Hyde Park, London” is usually abbreviated as “Hyde Park” within the 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.) in the report later.

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 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]), is for recognizing all Taiwanese city districts. 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]) is for recognizing all Taiwanese city parks. Landmark templates are built in the same way. E.g., [通用地理.台灣.市]:$$(2..4):路 ([GeoMap.Taiwan.Cities]:$$(2..4):Road), is for recognizing roads in Taiwan.

For each InfoMap template category x (e.g., location and organization), there are two features associated with it. The first is Current-Char-InfoMap-x-Begin, which is set to 1 for the first character of 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 is: using InfoMap to help ME detect which character in the sentence is the first character of a 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 is tagged as *x_begin* while Current-Char-InfoMap-x-Continue is helpful for determining which character is tagged as *x_continue* if we build InfoMap template for that category x. The two features associated with x category are showed below.

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

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

In 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 we 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	“台”北縣板	Probably the

	InfoMap location template, where the character length of the template is n	橋市	leading character of a location
Current-Char-InfoMap-Location-Continue	$c_a \ldots c_0 \ldots c_b$ 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 the continuing character of a location

3.2.3 Features for Recognizing Organization Names

Organizations include named corporate, governmental, or other organizational entity. The difficulty of recognizing an organization name is that an organization name is usually led by location names, such as 台北市地檢署 (Taipei District Public Prosecutors Office). Therefore, traditional machine learning NER systems 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, we found that organization names are mostly ended with a specific keyword or led by 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 by boundary markers:

Type I: With left and right boundary markers:

The organization name in this category is led by one or more than one 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 left boundary markers:

The organization name in this category is led 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 right boundary marker:

The organization name in this category is ended by an organization keyword. For example, 捷安特公司 (Giant Corporation) only contains the right boundary 公司 (Corporation).

Type IV: No boundary marker:

In this category, both left and right boundaries as above mentioned are omitted, such as 捷安特 (Giant). The organization names in this category are usually in the 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 build [通用地理.台灣.市]:$$(2..4):局 ([GeoMap.Taiwan.Cities]:$$(2..4):Department) for recognizing county level government departments in Taiwan. However, in Type II, III, 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.

Once 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 in recognizing organization names.

Table 4. Organization Features

Feature	Feature Conditions	Example	Explanations
Current-Char-InfoMap-Organization-Begin	c_0~c_{n-1} is 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
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
Current-Char-Organization-Keyword	c_0 or c_0c_1 or $c_{-1}c_0$ are in the organization keyword list	“公”司, 公“司”	Probably part of an organization keyword

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 provides a greater variety of organization names than single domain corpus does. The full location names and organization names are comparatively longer and our corpus contains more location names and addresses at 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 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 configured the following four settings of our NER system to analyze the effects of using Maximum Entropy-based Framework and 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). Following the standard 10-fold cross-validation method, we tested Mencius in each configuration using the data set mentioned in Section 4.1. The following subsections show details of each configuration and the results of it.

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

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

Table 6. Performance of 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 with 100,000 words CKIP Traditional Chinese dictionary to split a sentence into tokens. This module combined from the results of forward and backward longest matching algorithm in the following way: in the agreement part (the segmentation result of forward longest matching and the result of backward longest matching are the same), we regard the word as a token. While in the distinct part (the segmentation result of forward longest matching and the result of backward longest matching are not the same), we split the substring into characters. The accuracy of the agreement part is 98%. Then, we use a person name list and InfoMap templates to recognize all named entities. The number of lexicons in person name lists and gazetteers is 32000. As shown in Table 6, the results indicate the F-Measures of PER, LOC and ORG are 89.0%, 74.1% and 71.6%, respectively.

Table 7. Performance of 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 (HSCT)

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

Table 8. Performance of 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 (HSWT)

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

Table 9. Performance of 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 observe that TW achieves much better precision than TC in PER. With word segmentation, if some trigram, or 4 gram is not a personal name, the string consists of characters before (or after) this gram and some characters in this gram may form a Chinese word. Take 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 leading trigram template. However, in TW, due to the word segmentation result sentence “新 古典 主義”, the last character “主” and the following character form a Chinese word “主義”, so “古典” plus “主義” cannot match the family-name leading trigram template.

HC versus HW

We observe that HW achieves almost the same precision as HC in all three NE categories. HW also achieves similar precision with HC in PER and ORG. For NE in PER, this is because the length of person names is 2 to 4. Therefore, the 5 characters long window (-2 to +2) is enough to recognize a person name. For NE in LOC, HW recall is worse than HC. This is because the word segmentation module split an occupation name as a word, for example, “台北市長”. Since for HW, the basic token is word, so it cannot extract the LOC NE “台北市” from the token “台北市長”. For NE in LOC and ORG, we need more high-level features and external features. Since Mencius lacks those kind of features, HW doesn’t get significant better performance than HC.

TC versus HC

We observe that in PER, HC archieves much better precision than TC, but in LOC and ORG, HC performs slightly better than TC. This is because for NE in PER, most key features for identifying person names are close to the personal name, or inside the personal name. Take the sentence “立即連絡海鷗直” for example, while determining whether “連絡海” is a person name, we can see that “立即” seldom appears before a person name, and “鷗” seldom appears after a personal name. So ME can use this information to recognize that “連絡海” is not a personal name, but, to recognize a location name and a 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” for example, the two preceding characters are “美”and “、”, and the two following characters are “、”and “蘭”. ME cannot use these information to an identify location name.

TW versus HW

We observe that HW achieves better precision than TW in identifying person names. This is because 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” for example, it matches the leading person name template of double family name, because “王” and “金” are both family names. However, “王金平” is the right person name. ME can use the information that “王金平” has appeared in the training corpus to identify the personal name “王金平” from a sentence. We also observe that HW achieves better recall than TW in identifying personal names. This is because HW can find some bigram personal name from training data, but TW cannot because we don’t have bigram person name templates. In addition, some personal names are in the dictionary, so some tokens are personal names. Take “陳建仁 的 作爲” for example. Although the token “陳建仁” cannot match any person name template, ME can use context information and training data to recognize “陳建仁”. In identifying location names, ME needs a wider context to detect location names, so HW’s recall is worse than TW. 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 Chinese. Therefore, ME achieves better precision in identifying location names.

5 Conclusions

In this paper, we have developed a Chinese NER system, Mencius. We configured the following settings of Mencius 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 experiment result showed that whether taking a character or a word as a token, hybrid NER System always performed better performance in identifying person names. However, it has little affect in identifying location and organization names. This is because the context information around a location name or an organization name is more complex than around a personal name. Besides, using a word segmentation module improves the performance of a pure Template-based NER System. But, it has little affect in 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

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

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

[3] R. Grishman, "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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[20] D. McDonald, "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.