This article has been accepted for publication in IEEE Intelligent Systems but has not yet been fully edited.
Some content may change prior to final publication.

A Boundary Assembling Method for Chinese Entity Mention Recognition

Yanping Chen*, Qinghua Zheng† and Ping Chen‡
Email: ypench@gmail.com, qhzheng@mail.xjtu.edu.cn, ping.chen@umb.edu
*†MOEKLINNS Lab, Department of Computer Science and Technology
Xi'an Jiaotong University, Xi'an, China
‡Department of Computer Science, University of Massachusetts Boston, Boston, USA

***Abstract*—In this paper, a Boundary Assembling (BA) method is presented for Chinese entity mention recognition[1]. Given a sentence, instead of recognizing entity mentions in a unitary style, our BA method first detects boundaries of entity mentions, then assembles detected boundaries into entity mention candidates. Each candidate is further assessed by a classifier trained on non-local features. Our method can make better use of non-local features and effectively recognize nested entity mentions. Using the ACE 2005 Chinese corpus, our experimental results show an impressive improvement over the state-of-art techniques. It outperforms existing methods in F-score by 5% in entity mention detection and 4.23% in entity mention recognition.**

I. Introduction

In Chinese, named entities are an important part of Out-Of-Vocabulary (OOV) words, which is considered a major obstacle to segment Chinese sentences [1]. Many NLP tasks (e.g., POS tagging, parsing, etc.) rely on accurate outputs of Chinese word segmentation. Errors caused by Chinese Named Entity Recognition (CNER) can propagate and lead to a cascading failure. Therefore, CNER is a critical issue in Chinese NLP.

In Chinese the lack of capitalization information and delimitation word makes CNER difficult. Issues such as morphological complexity, segmentation ambiguity worsen the problem. Furthermore, in Chinese, every character can become part of an entity name. It is difficult to distinguish monosyllabic words from monosyllabic morphemes [2]. Currently CNER still remains a challenging task.

The CNER task can be modelled by a label tagging process. Each character (or word) is tagged by a label indicating whether or not it is part of a named entity. Traditionally, labels are encoded by "B-I-L-O" coding (*Beginning*, *Inside*, *Last* and *Outside* of entity name). According to the type of tagging units, CNER models can be divided into word-based models and character-based models.

Word-based models perform word segmentation first. The tagging units are words. They benefit from a large window of context, which leads to a flexible decision boundary. Moreover, after the segmentation has been done, other analytical processes (e.g. POS tagging, parsing) can be performed. One disadvantage of word-based models is that Chinese word segmentation is error-prone. The errors will propagate and worsen the performance.

In character-based models, a sentence is tokenized into a character sequence, then each character is labelled. Character-based models merge word segmentation and named entity recognition into a unified framework. It can reduce errors caused by segmentation. The main disadvantage of character-based models is that, without segmentation, some syntax or semantic features are not available.

Generally, given a sentence represented as a sequence tagging units, a sequence model (e.g., HMM, CRF) is used to give a maximized label sequence. Sequence models are effective in modelling sentence structure, and can make better use of dependencies between tagging units. But it suffers from the problem that all features are used in the same manner without distinguishing the local and non-local properties. With prior knowledge about entity boundaries, some non-local features are difficult to be used. Furthermore, giving a label sequence is not effective in recognizing nested entity mentions.

There are researchers (e.g., [3]) use parsing trees to recognize nested English named entity, where a node may be an entity mention containing another entity mention as child node. While parsing Chinese is difficult and error-prone, and heavily depends on Chinese segmentation. Falsely segmented entity fragments are often mixed with other characters. Furthermore, a parser itself needs to be trained on external resource with many entity names, which may affect the performance of a CNER method.

In this paper, a BA method is proposed for CNER. Section II discusses related work. The motivation and the BA method are discussed in Section III and Section IV. Experiments are conducted in Section V. Section VI gives our discussions and conclusions.

II. Related Work

As discussed above, CNER methods are divided into word-based and character-based models. Another criterion to classify them is whether the tagging units are dependent on each other or not.

If labels of tagging units are independent and identically distributed (I.I.D or iid), a likelihood function (a classifier) can be used to evaluate the distribution of tagging units independently. After tagging units were labelled, three inference

[1]A entity mention is a reference of a named entity. In this paper, we do not distinguish between *entity mention* and *named entity*, and use them interchangeably.

Digital Object Indentifier 10.1109/MIS.2015.71 0885-9000/$26.00 2015 IEEE

This article has been accepted for publication in IEEE Intelligent Systems but has not yet been fully edited.
Some content may change prior to final publication.

algorithms are employed to combine them: greedy matching, dynamic programming and beamsearch [4].

If labels of tagging units are dependent, the task is modelled as a sequence analysis. Under this condition, each sentence is processed as a sequence data. A generative method or a discrimination method can be adopted to find a maximized label sequence. In addition to probabilistic methods, by using discriminant functions, other sequence methods are also developed (e.g. SVMs chunker [5]).

Hand-crafted rules were used for English entity recognition. Due to characteristics of Chinese, without additional information, the rule-based methods are incompetent for CNER. Most rule-based CNER models combine statistical information with rules (or patterns) [6] or directly use rules as features for classification [7].

According to the type of tagging units and dependence between them, we can roughly classify existing models into five categories: *character-iid*, *word-iid*, *character-sequence*, *word-sequence* and *rule-based*. Some also use mixed models. Moreover, there are three additional issues about CNER: employed features, external resource and recognition framework.

Various features have been adopted. N-Gram and lexical features are the most popular ones. Syntactic features are also widely used, e.g., POS tags, phrase chunker, dependency trees, etc. In most cases, syntactic features are obtained by using toolkits trained on external corpora.

Utilization of external resources is important for named entity recognition. Various resources have been explored, e.g., domain knowledge, lexicon, gazetteer, WordNet, thesaurus, heuristic information (e.g., family or transliterated names), bilingual information [8] and title of named entity [6]. Even the outputs generated by other CNER models are explored [9].

Many CNER models include an integrated framework, recognizing all kinds of entities in a unified manner. However, if different named entity types are under consideration, word-formations may be different. For example, organization names and location names are more likely to be nested. Characters for transliterated person names usually come from a fixed set. There are researchers who categorize Chinese words into different types, then different approaches are adopted to recognize them [2], [10].

In the next section we will discuss motivations, which lay the foundation for BA method.

III. Motivation

Let L_i, C_i be random variables over label set and tagging units. L_i has value in {B, I, O}. $\mathbf{L} = L_{1,\cdots,n}$ and $\mathbf{C} = C_{1,\cdots,n}$ are variable sequences. Given a sentence $(C_{1,\cdots,n})$, the task of named entity recognition is to find a maximized label sequence(s) $L_{1,\cdots,n}$. It satisfies $\arg\max P(L_{1,\cdots,n}|C_{1,\cdots,n})$.

If tagging units are dependent, a sequence method can be used to model this process. A generative method $P(\mathbf{L}, \mathbf{C}|\theta)$ or a discrimination method $P(\mathbf{L}|\mathbf{C}, \theta)$ can be used (θ are model parameters). Both have forms as:

$$P = \frac{1}{Z}\prod_{i=1}^{n} P(L_i|L_{(i-1,\cdots,i-k)}, \mathbf{f}(C_i)) \quad (1)$$

Where Z is the normalization factor. k is the order of Markov chain. $\mathbf{f}(C_i)$ denotes visible features about C_i, and $L_{(i-1,\cdots,i-k)}$ are labels of $C_{(i-1,\cdots,i-k)}$. If tagging units are I.I.D, then the distribution can be rewritten as:

$$P = \frac{1}{Z}\prod_{i=1}^{n} P(L_i|\mathbf{f}(C_i)) \quad (2)$$

In sequence method, predicting the current label L_i depends on its predecessors $L_{(i-1,\cdots,i-k)}$. It is considered that the sequence method has the ability to model a sentence structure. In the I.I.D method, predicted labels should be combined.

To recognize an entity mention from a sequence of characters, human readers will identify the boundary first. If features about a character indicate that a boundary is detected, we scan further for the other matching boundary. More than one boundary can be matched in this step. Then they are combined into entity candidates for evaluation. Because the combined entity candidates are considered, more features can be used for better recognition.

A cascading framework can be used to model this process. It depends on two facts. First, the boundaries should be detected accurately, which enables suitable entity candidates are collected. Secondly, how to appropriately extract and use features is the central issue for evaluating the collected entity candidates.

In this section, we examine the sequence and I.I.D methods and their results on the ACE 2005 Chinese corpus[2], which motivates our BA method. The result is shown in Table I,

TABLE I: Performance on Boundary

No.		Model	P	R	F
(1)	Sequence	Character	84.78	77.67	81.07
(2)	Sequence	*Beginning*	90.20	79.94	84.76
(3)	Sequence	*Last*	91.63	84.63	87.99
(4)	I.I.D	Character	81.07	73.11	76.88
(5)	I.I.D	*Beginning*	90.51	78.79	84.25
(6)	I.I.D	*Last*	91.76	83.85	87.62

(1) and (4) are *character-sequence* and *character-iid* models for entity mention detection, which use the *Basic Feature* (See Table II of Section II) and a CRF toolkit (the I.I.D method does not set the 1-order Markov dependence). In (2), (3), (5) and (6), *Beginning* and *Last* are the models detecting only the *Beginning* or *Last* boundaries.

Table I shows two results. **First**, compared to entity mention detections, boundary detections have higher performance. **Secondly**, comparing Row 2 with Row 5 and Row 3 with Row 6, both sequence and I.I.D methods have similar performance in boundary detections.

The first result is reasonable, because the detection of entity mention depends on the detection of both *Beginning* and *Last* boundaries. For the second result, entity mentions tend to be short chunks separated by irrelevant tokens [4]. Compared to a

[2]Details of this corpus are included in Section V-A, entity head mentions are used here.

Digital Object Indentifier 10.1109/MIS.2015.71 0885-9000/$26.00 2015 IEEE

I.I.D method, a sequence method does not show any advantage for detecting entity boundaries.

This conclusion gives us some important insight. We can use traditional methods to detect entity mention boundaries, then assemble these boundaries into entity candidates. Finally, a classifier, trained by non-local features, can be employed to evaluate such entity candidates.

IV. Boundary Assembling Method

Our BA method is presented in Figure 1. Each C_i represents a tagging unit. It is assumed to be a Chinese character. But it can also be a word or other language tagging unit.

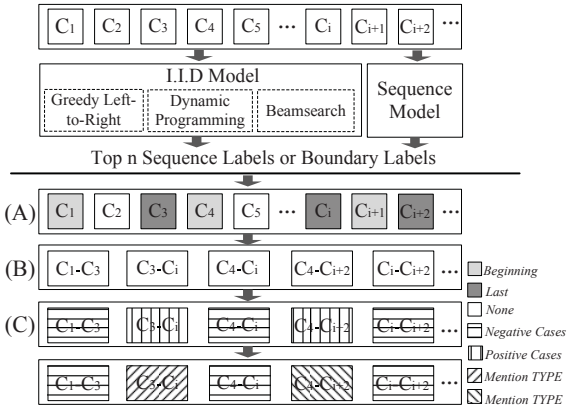

Fig. 1: Boundary Assembling Method

As Figure 1 shows, three steps are presented in the BA method. They are discussed as following.

(A) Boundary Step: This step focuses on entity mention boundaries (boundaries for short) detection. Boundaries are detected by traditional I.I.D or sequence methods. Outputs of this step can be boundaries, entity mentions or both.

(B) Assembling Step: This step focuses on generating entity mention candidates (candidates for short). Using traditional CNER models, if outputs in boundary step are candidates directly, then the assembling step can be skipped. If outputs of boundary step are boundaries, they should be assembled into candidates by using assembling strategies. Various assembling strategies can be developed as Figure 2 showing:

(a) and (b) are the greedy matching approaches. In (c), we present a *paring method*. For a *Last* label, we try to find another left *Last* label that is across a *Beginning* label. Then, among the paired *Last* labels, top *N Beginning* labels with higher probability are assembled with the referring label. For example, in (c), setting C_5 as the referring label, the matching character is C_2. From C_2 to C_5, supposing that C_3 has a higher *Beginning* probability, then $C_3C_4C_5$ and C_4C_5 are assembled (*N*=2). This method uses the information that, in the ACE corpus, recognizing *Last* labels has higher performance. A *Beginning* label between two *Last* labels is expected. Paring method (d) uses the *Beginning* as referring labels.

(C) Discrimination Step: This step focuses on deleting improper candidates. A classifier is trained to classify positive and negative candidates. Because the candidates were detected, more features can be extracted.

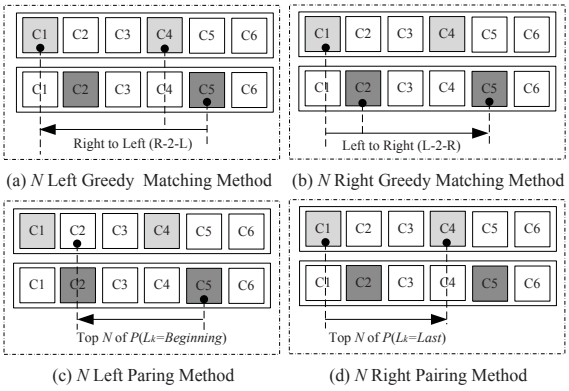

Fig. 2: Assembling Strategies

The idea of discrimination step also discussed by [11], which uses a machine learning method to filter entity candidates. The difference is that, instead of implementing boundaries detection and assembling, their entity candidates come from a dictionary.

Evaluation Process: For a given corpus, the process to evaluate the BA method is shown in Figure 3.

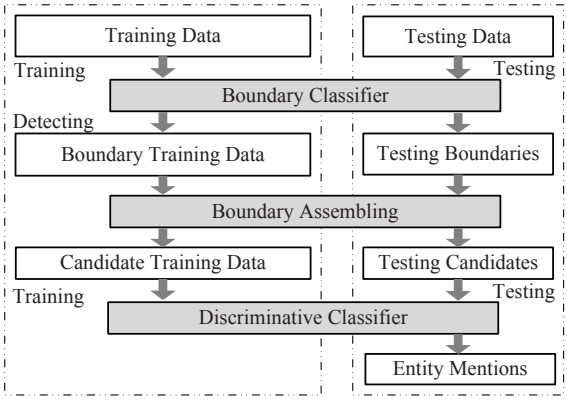

Fig. 3: Evaluation Method

Two classifiers are employed: *boundary classifier* and *discrimination classifier*. In the boundary step, local features are used to train the classifier. In the discrimination step, both local and non-local features can be employed. In order to train a discrimination classifier, the *candidate training data* is generated by running the boundary classifier on training data again. Boundaries (or candidates) in the training data are assembled (or collected) to train the discrimination classifier.

V. Experiments

In this section, after discussing settings about experiments, we implement the-state-of-the-art method for comparison. Several experiments are conducted to show the performance of BA method.

Digital Object Indentifier 10.1109/MIS.2015.71 0885-9000/$26.00 2015 IEEE

TABLE II: Feature Sets

ID	Feature Set	Description	Examples	
(1)	Character-based	Character Unigram and Bigram.	C_3, C_4, C_5, C_6, C_7, C_4C_5, C_5C_6	Basic Feature
(2)	Gazetteer-based	Boolean value: whether it is a surname, a pronoun or located in a location name.	$IsSur(C_5)$, $InLoc(C_4C_5)$, $InLoc(C_5C_6)$, $IsPro(C_5)$, $IsPro(C_4C_5)$, $IsPro(C_5C_6)$	
(3)	Segmenting-based	Boolean value: Is a Segmented Left or Right Boundary.	$IsLeftBefore(C_5)$, $IsRightAfter(C_5)$	
(4)	Boundary Characters	Characters around a predicting boundary.	$C_2C_3C_4$, C_3C_4, C_4, C_5, C_5C_6, $C_5C_6C_7$	Extended Feature
(5)	Inner n-Gram	n-Gram within the entity mention.	C_5, C_6, C_5C_6	
(6)	Combined n-Gram	Combined features in two sides of an entity mention.	$C_4\_C_7$, $C_3C_4\_C_7C_8$	
(7)	Internal Words	The bag-of-words features in an entity mention candidate.	Every possible word in C_5C_6	
(8)	External Words	Word feature before (-) or after (+) an entity mention candidate.	$Word\_{-i}$, $Word\_i$ $(i \in \{1,2,3\})$	

A. Data

The ACE 2005 Chinese corpus is employed in our experiments, which contains 633 documents. There are 15,264 entities and 33,932 entity mentions. An *entity mention* is a reference to an entity. 7 entity types (e.g. *person*, *organization*, etc.) are defined. An entity mention is annotated with its full *extent* and its *head*. An *extent mention* includes both the *head* (*head mention*) and its modifiers.

In the ACE corpus, many *extent mentions* are nested. In this paper, we focus on these nested entity mentions. It is a good way to exhibit the capabilities of the BA method and show how to use non-local features. Furthermore, nested entity mentions play linguistic or syntactic functions in a sentence, recognizing them is helpful for tasks, e.g., chunking, segmenting or parsing.

B. Feature Sets

All the employed features are listed in Table II. A character sequence $C_1C_2C_3C_4C_5C_6C_7C_8$ is given as an example for extracting corresponding features, where C_5 is used as a boundary to extract the *Basic Feature*, and C_5C_6 is an entity mention candidate for the *Extended Feature*.

To extract feature sets (3), (7) and (8), Maximum Matching (MM) is used to segment each sentence, where the *Lexicon of Common Words in Contemporary Chinese* is adopted for word matching. The employed gazetteers contain 539 surnames, 31,726 location names and 125 pronouns. Feature sets (1), (2) and (3) are adopted by [12]. They are referred to as *Basic Features*. Feature sets from (4) to (8) are referred to as *Extended Features*. Some of them are considered as non-local features.

To determine whether a feature is local or non-local is not always straightforward. In general *Character-based* and *Boundary Characters* features are local features. While *Inner n-Gram*, *Combined n-Gram*, *Internal Words* and *External Words* features are considered as non-local features, because to extract these features, the candidates must be found first. For *Gazetteer-based* and *Segmenting-based* features, it is difficult to specify whether they are local or non-local.

C. Experimental Settings

In the ACE evaluation, reference mentions and recognized mentions are allowed to have a mutual overlap of at least *min_overlap*[3]. Because we focus on the detection of entity mention boundary. We assume that a correct detection of entity mention is achieved when two boundaries are found correctly. In all experiments, *entity mention detection* refers to determining whether a phrase is an entity mention or not. In addition to detect entity mentions, *entity mention recognition* should further recognize the type of each entity mention.

To implement sequence methods, the CRF toolkit[4] is used. I.I.D methods use the Maximum Entropy (ME) toolkit[5]. To evaluate the BA method, the corpus is randomly partitioned into five even parts at the document level. Then the 5-cross valuation is carried ou. We use the P/R/F (Precision/Recall/F-score) measurement. F-score is computed by $(2\times P\times R)/(P+R)$. In all tables, the performance about "total" is counted by merging the recognized mentions on all entity types. To recognize entity mentions, according to the employed method, multiple labels may be used (e.g., B_PER, I_PER, O, etc.). "Det" is the performance of entity mention detection that all entity mentions are encoded with the same type (the "B-I-O" coding).

D. Comparisons with Existing Methods

When entity mentions are nested, if a method always labels the outermost mentions, it is referred to as an *outermost* model. Otherwise, it is referred to as an *innermost* model (labelling the innermost). In the corpus, there are 33,932 *extent mentions* in total. There are 24,731 outermost mentions and 25,766 innermost mentions. Both the *outermost* and the *innermost* are used as baseline performance for named entity recognition. Because the nestification problem, directly training a sequence model on the outermost mentions or the innermost mentions will lead to worse performance.

[3]min_overlap has value 0.30 in the ACE 2007 evaluation.
[4]http://crfpp.googlecode.com/svn/trunk/doc/index.html
[5]http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

Digital Object Indentifier 10.1109/MIS.2015.71 0885-9000/$26.00 2015 IEEE

TABLE III: Comparing Performance with Other State-of-art Methods

		Cascading-out			Cascading-in			Inside-out			Inside-in		
TYPE	**Count**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
FAC	1,688	61.98	22.21	32.70	63.20	21.26	31.82	53.26	30.50	38.79	56.90	28.08	37.60
GPE	8,627	80.10	60.17	68.72	84.03	63.40	72.27	76.68	63.31	69.36	76.40	66.85	71.31
LOC	1,546	64.15	32.53	43.17	65.47	32.01	43.00	59.23	33.63	43.90	61.02	34.73	44.27
ORG	6,636	72.53	49.14	58.58	72.77	48.22	58.00	68.93	49.30	57.49	68.11	48.71	56.80
PER	14,393	75.19	55.22	63.67	75.91	55.04	63.81	72.45	63.55	67.71	72.84	61.44	66.66
VEH	665	57.06	31.57	40.65	56.50	30.67	39.76	59.48	38.19	46.52	63.69	35.33	45.45
WEA	377	59.03	25.99	36.09	59.14	25.72	35.85	60.00	27.85	38.04	63.64	28.11	39.00
total	33,932	74.93	51.82	61.27	76.52	51.80	61.80	71.54	56.80	63.32	71.93	56.57	63.33
Det	33,932	74.18	48.55	58.69	74.59	49.39	59.43	74.85	61.57	67.56	74.23	61.53	67.28

Features proposed by [12] are referred to as *Basic Feature* (See Table II). They reported the best performance on the ACE 2005 Chinese corpus. However, in their paper, the nestification problem is not considered. To model and recognize nested named entity, [13] introduced and compared three techniques: *cascading*, *layering* and *joined tagging*. In Table III, we compare our method with other state-of-art methods.The *cascading* (*cascading-out*, *cascading-in*) and *layering* (*inside-out*, *inside-in*) are conducted for comparison.

In the *cascading* models, each entity type is trained and tested separately by a classifier. If nested mentions within the same type are encountered, the *cascading-out* chooses the outermost and the *cascading-in* chooses the innermost. In layering, each level of nesting is modelled by a sequence model. The *inside-out* implements the outermost model and the innermost model respectively, and the nested mentions of the innermost model are merged into the outermost model. Each type is encoded independently (e.g., B_PER, I_PER, O, etc.). In the *inside-in* model, each entity mention in the innermost is collected. If it is contained by an entity mention in the outermost, then the latter also added. The data in each type of entity mention is unbalanced[6]. Therefore, some entity types achieve lower performance. The *layering* has higher performance than the *cascading* because in the corpus, some nested mentions have the same type, which cannot be divided in the *cascading* models.

E. Impacts of Assembling Strategies

As discussed in Section IV, Figure 2 presents four assembling strategies for BA method. Different assembling strategies may influence the collected candidates. The number of assembled candidates is an important issue. A large number can increase recall rate, meanwhile, decrease precision. This experiment shows the impact of assembling strategies on the final performance. For each assemblage strategy, *Basic Feature* is used for boundary detection. After the candidates are collected, the *Basic Feature* and *Extended Feature* are used to divide them. The result is shown in Table IV.

The boundary classifier will be implemented on the training data for generating the *candidate training data* (See Figure 3). If N=1, it is possible that the assembled candidates are all true candidates. To train a discriminative classifier, we need to ensure that both positive and negative candidates are generated. If all candidates are positive (or negative), a re-ranking technique will be more appropriate. In Table IV, setting N>1 can ensure that both positive and negative candidates are produced. Strategy (3) with N=3 is set as our default setting.

[6]For example, WEA has 377 cases, while PER has 14,393 cases.

TABLE IV: Impacts of Assembling Strategies

	N=2			N=3		
	P	**R**	**F**	**P**	**R**	**F**
(a)	81.10	58.10	67.70	80.59	58.23	67.61
(b)	81.85	55.13	65.89	81.61	56.41	66.71
(c)	75.35	61.13	67.50	74.47	63.75	68.70
(d)	73.42	52.82	61.44	77.53	59.65	67.43

F. Performance of Discrimination

This experiment shows the impact of the discrimination step. The result is shown in Table V. Where (1) and (2) are the baseline performance of a sequence model recognizing nested entities. (3) and (4) are greedy matching models with N=1. (5) is also a greedy matching model with N=2. (6) is the same as (5) in the boundary step and the assembling step. They all use I.I.D classifiers and *Basic Feature* to detect boundaries. In (6), the discrimination step is implemented by using *Basic Feature* and *Extended Feature*.

TABLE V: Performance of Discrimination

No.	**Model**	**P**	**R**	**F**
(1)	Outermost	74.18	48.55	58.69
(2)	Innermost	74.59	49.39	59.43
(3)	Greedy L-2-R	72.30	47.34	57.22
(4)	Greedy R-2-L	62.52	62.24	62.38
(5)	Greedy R-2-L†	49.66	64.13	55.97
(6)	Greedy BA	81.10	58.10	67.70

In greedy matching models, nested entity mentions are allowed, which increases the recall rate. Therefore, (4) shows better performance than sequence methods. Because detecting *Last* boundaries has higher performance, (4) outperforms (3). In (5), matching each boundary to two left boundaries worsens the performance. Compared (5) with (6), after performing the discrimination step, the performance is improved considerably, increasing F-score by 11.73%. It outperforms the baseline performance about 8.27% in F-score.

Digital Object Indentifier 10.1109/MIS.2015.71 0885-9000/$26.00 2015 IEEE

G. Ability to Use Feature

In the BA method, each feature can be used in the boundary step or in the discrimination step. This experiment is set up to show the impact of features when used in different step.

Two models are conducted. BA_α uses *Basic Feature* in the boundary step, while BA_β uses *Character-based* features and *Boundary Characters* features in this step. In the discrimination step, both use *Basic Feature* and *Extended Feature*. All use the I.I.D classifiers and the left paring approach with *N*=3. The result is shown in Table VI.

TABLE VI: Ability to Use Feature

	BA_α			BA_β		
TYPE	**P**	**R**	**F**	**P**	**R**	**F**
FAC	53.81	30.15	38.65	57.82	36.14	44.48
GPE	76.42	70.45	73.32	80.03	74.71	77.28
LOC	58.27	39.20	46.87	58.28	45.86	51.71
ORG	67.00	54.58	60.16	71.25	56.72	63.16
PER	72.83	60.44	66.06	74.68	63.86	68.85
VEH	58.08	34.59	43.36	60.15	36.54	45.46
WEA	56.46	31.30	40.27	61.79	34.75	44.48
total	71.20	58.53	64.25	73.98	62.16	67.56
Det	74.47	63.75	68.70	76.88	68.70	72.56

The result shows that it is better to use *Character-based* features and *Boundary Characters* features in the boundary step, while error-prone features (e.g. *Gazetteer-based* features and *Segmenting n-Gram* features) are more suitable in the discrimination step. Comparing the performance of BA_β with Table III, our method outperforms the *inside-in* methods in F-score by 5% in entity mention detection and 4.23% in entity mention recognition.

H. Outputs Modification

This experiment shows the capability of our BA method to improve performance of other classifiers. Sequence methods are used to generate n-Best outputs (label sequences). In this experiment, instead of re-ranking the outputted label sequences, we collect the detected entity mentions in the top n ($n \in \{1, 2, 3\}$) label sequences. Then, a discrimination classifier is adopted to filter these entity candidates.

As Table VII shows, *Outermost* and *Innermost* are conducted. *Post-Boundary* is obtained by directly merging outputs of *Outermost* and *Innermost*. In *Post-Discrimination*, the merged outputs are further classified by a discrimination classifier.

TABLE VII: Outputs Modification

	P	**R**	**F**	**P**	**R**	**F**
n-Best	**Outermost**			**Innermost**		
n=1	74.18	48.55	58.69	74.59	49.39	59.43
n=2	48.92	56.56	52.46	52.13	57.25	54.57
n=3	36.40	60.30	45.39	39.12	61.14	47.71
	Post-Boundary			**Post-Discrimination**		
n=1	71.06	62.94	66.75	76.63	59.75	67.15
n=2	46.54	70.08	55.93	53.30	65.28	58.68
n=3	34.17	72.78	46.51	41.28	66.91	51.06

In *Post-Discrimination*, two CRF classifiers are implemented to detect *Beginning* and *Last* boundaries respectively. Marginal probabilities of labels are collected. The paring method is used to generate the candidates for training a discrimination classifier, where the *Baseline* and *Extended Features* are used.

Results show that both boundary step (in *Post-Boundary*) and discrimination step (in *Post-Discrimination*) improve the performance. Discrimination step improves the precision, but hurts the recall. Even so, the final performance is increased. In the top two and top three outputs, discrimination step shows clear improvement.

VI. Discussion and Conclusion

In the traditional methods, a maximized label sequence is returned by a sequence model. In [14], n-Best label sequences are output first, then non-local features are used to re-rank the outputs for an optimized label sequence. The main problem for finding a label sequence is that it hard to identify nested mentions. For example, given the label sequence "O-B-I-B-I-O", whether it consists of two independent or two nested mentions is hard to determine.

In order to tackle the nestification problem, each entity type can be recognized separately in a *cascading* model. However, processing each entity type separately still cannot recognize nested entities with the same type. Also it suffers from the weakness that classifiers cannot make full use of annotated information. The general understanding is that, by returning a labelled sequence, sequence methods are not suitable for recognizing nested entity mentions due to the following reasons:

Firstly, in traditional methods, all features are used in the same manner and decisions are made mainly based on a monolithic feature set $\mathbf{f}(C_i)$ (See Equation 1 and Equation 2). However, in practice such features often behave differently. Features may be local or non-local, segmentation error-prone (or independent). It is better to use them differently. Without boundary information about entity mentions, it is difficult to segment a sentence properly. Therefore, faulty segmentation may actually hurt entity mention recognition.

Secondly, utilization of external resources (e.g. gazetteer or thesaurus) in sequence methods may also become problematic. In Chinese, every character can act as part of an entity name. Therefore, distributions of such features among entity mentions or sentences may be different. For example, given an entity mention candidate, if a surname appears in the first position, very likely this is a person name. However, (e.g., in a sequence method) if given a sentence instead, because Chinese surnames are also widely used as monosyllabic morphemes to form arbitrary words, the probability that this is a person name becomes much lower. Such features are more suitable to be used in the discrimination step.

Thirdly, to label nested entity mentions in a sequence method, if the outermost mention is chosen, boundaries of inner entity mentions will be labelled as "I". If the innermost mention is chosen, boundaries of the outermost mentions are labelled as "O". This approach used in sequence methods can hurt the performance since falsely labelled boundaries used in training can distort feature weights for later classification.

Digital Object Indentifier 10.1109/MIS.2015.71 0885-9000/$26.00 2015 IEEE

In this paper a BA method is presented, which divides entity mention recognition into three steps. It is effective for recognizing nested entity mentions and make better use of non-local features. In future, it can be developed to support other languages. And more local, non-local features can be explored for better performance.

VII. Acknowledges

We thank our students: Yao Zhao, Yunfei Hong and Cong Li. The research was supported in part by the National Science Foundation of China under Grant Nos. 91118005, 91218301, 91418205; National High Technology Research and Development Program 863 of China under Grant No. 2012AA011003.

References

[1] R. Sproat and T. Emerson, "The first international chinese word segmentation bakeoff," in *Proceedings of the SIGHAN '03*, pp. 133–143, ACL, 2003.

[2] K.-J. Chen and M.-H. Bai, "Unknown word detection for chinese by a corpus-based learning method," *IJCLCLP*, vol. 3, no. 1, pp. 27–44, 1998.

[3] J. R. Finkel and C. D. Manning, "Nested named entity recognition," in *Proceedings of the EMNLP '09*, pp. 141–150, ACL, 2009.

[4] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the CoNLL '09*, pp. 147–155, ACL, 2009.

[5] G. C. Ling, M. Asahara, and Y. Matsumoto, "Chinese unknown word identification using character-based tagging and chunking," in *Proceedings of the ACL '03*, pp. 197–200, ACL, 2003.

[6] L.-J. Wang, W.-C. Li, and C.-H. Chang, "Recognizing unregistered names for mandarin word identification," in *Proceedings of the COLING '92*, pp. 1239–1243, ACL, 1992.

[7] T.-H. Tsai, S.-H. Wu, C.-W. Lee, C.-W. Shih, and W.-L. Hsu, "Mencius: A chinese named entity recognizer using the maximum entropy-based hybrid model," *IJCLCLP*, vol. 9, no. 1, 2004.

[8] M. Wang, W. Che, and C. D. Manning, "Joint word alignment and bilingual named entity recognition using dual decomposition," in *Proceedings of the ACL '13*, pp. 1073–1082, 2013.

[9] R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, S. Roukos, and T. Zhang, "A statistical model for multilingual entity detection and tracking," in *Proceedings of the HLT-NAACL '04*, 2004.

[10] J. Sun, J. Gao, L. Zhang, M. Zhou, and C. Huang, "Chinese named entity identification using class-based language model," in *Proceedings of the COLING '02*, pp. 1–7, ACL, 2002.

[11] Y. Tsuruoka and J. Tsujii, "Boosting precision and recall of dictionary-based protein name recognition," in *Proceedings of the ACL '03*, pp. 41–48, ACL, 2003.

[12] W. Li, D. Qian, Q. Lu, and C. Yuan, "Detecting, categorizing and clustering entity mentions in chinese text," in *Proceedings of the SIGIR '07*, pp. 647–654.

[13] B. Alex, B. Haddow, and C. Grover, "Recognising nested named entities in biomedical text," in *Proceedings of the BioNLP '07*, pp. 65–72, ACL, 2007.

[14] M. Collins, "Ranking algorithms for named-entity extraction: Boosting and the voted perceptron," in *Proceedings of the ACL '02*, pp. 489–496, ACL, 2002.

Digital Object Indentifier 10.1109/MIS.2015.71 0885-9000/$26.00 2015 IEEE