Natural Language Engineering (2019), **25**, pp. 239–255
doi:10.1017/S1351324918000463

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Out-domain Chinese new word detection with statistics-based character embedding

Yuzhi Liang[1], Min Yang[2], Jia Zhu[3,*], and S. M. Yiu[4]

[1]Department of Information Engineering, Peking University Shenzhen Graduate School, Shenzhen, China,
[2]Frontier Science and Technology Research Centre, Shenzhen Institutes of Advanced Technology, Shenzhen, China,
[3]Department of Computer Science, South China Normal University, Guangzhou, China and [4]Department of Computer Science, The University of Hong Kong, Hong Kong, China
*Corresponding author. Email: jzhu@m.scnu.edu.cn

(Received 10 August 2017; revised 22 October 2018; accepted 22 October 2018)

Abstract

Unlike English and other Western languages, many Asian languages such as Chinese and Japanese do not delimit words by space. Word segmentation and new word detection are therefore key steps in processing these languages. Chinese word segmentation can be considered as a part-of-speech (POS)-tagging problem. We can segment corpus by assigning a label for each character which indicates the position of the character in a word (e.g., "B" for word beginning, and "E" for the end of the word, etc.). Chinese word segmentation seems to be well studied. Machine learning models such as conditional random field (CRF) and bi-directional long short-term memory (LSTM) have shown outstanding performances on this task. However, the segmentation accuracies drop significantly when applying the same approaches to out-domain cases, in which high-quality in-domain training data are not available. An example of out-domain applications is the new word detection in Chinese microblogs for which the availability of high-quality corpus is limited. In this paper, we focus on out-domain Chinese new word detection. We first design a new method *Edge Likelihood (EL)* for Chinese word boundary detection. Then we propose a domain-independent Chinese new word detector (DICND); each Chinese character is represented as a low-dimensional vector in the proposed framework, and segmentation-related features of the character are used as the values in the vector.

Keywords: Chinese character embedding; Chinese new word detection; Chinese word boundary detection

1. Introduction

Unlike English and other Western languages, many Asian languages such as Chinese and Japanese do not delimit words by space. Word segmentation and new word detection are therefore key steps for processing these languages. Chinese word segmentation can be considered as a part-of-speech (POS)-tagging problem. We can segment corpus by assigning a label for each character which indicates the position of the character in a word (e.g., "B" for word beginning, and "E" for the end of the word, etc.). Chinese word segmentation seems to be well studied. Machine learning models such as conditional random field (CRF) (Lafferty, McCallum, and Pereira 2001) and bi-directional long short-term memory (LSTM) have shown outstanding performances on this task. However, the segmentation accuracies drop significantly when applying the same approaches to out-domain cases, in which a high-quality in-domain training set is not available (Zhang et al. 2012a). An example of out-domain applications is the new word detection in Chinese microblogs for which the availability of high-quality corpus is limited. In this paper, we focus on out-domain

Chinese new word detection. We first design a new method *Edge Likelihood (EL)* for Chinese word boundary detection. Then we propose a domain-independent CRF-based Chinese word segmenter named DICND; each Chinese character is represented as a low-dimensional vector in the proposed framework, and segmentation-related features of the character are used as the values in the vector.

2. Related work

The existing Chinese new word detection approaches can be divided into two categories, namely supervised approach and unsupervised approach.

2.1 Supervised Chinese new word detection

The supervised approach considers Chinese new word detection as a sub-task of Chinese word segmentation and solving Chinese word segmentation problem by sequence labeling. In the past decade, CRF and neural networks are two of the most popular methods in supervised Chinese word segmentation. In CRF-based Chinese word segmenters (e.g., Peng, Feng, and McCallum (2004)), each character is represented as a one-hot vector. Then CRF takes the one-hot vector as the input and labels Chinese sentences according to the transition probabilities of the labels in the training set. However, in a one-hot character vector, the attributes of the vector are the set of vocabulary in the training set plus *n*-gram lexicon features (i.e., if 3-gram is used, any three-character sequence in the training corpus will be an attribute and take a dimension of the vector). A boolean value is assigned to each attribute to indicate whether the character is part of a word which relates to the attribute. Thus, this kind of *n*-gram representation is sparse, high dimensional, and biased to known words. As a consequence, the one-hot character vector constrains the capability of CRF in Chinese new word detection, especially when training data and test data are in different domains. Intuitively, the out-domain problem can be solved by using a domain-specific dictionary. Wang *et al.* (2012) applied CRF-based Chinese word segmenter to Chinese microblog data and handled the out-domain problem by leveraging an external word list of popular Internet slang. But this method does not work for Chinese new word detection. Another approach is utilizing domain adaption techniques or adding domain adaptive features in character representation. For instance, Liu *et al.* (2014) applied domain adaption techniques in Chinese word segmentation; Zhang *et al.* (2012b) designed a set of features to indicate the length of the domain-specific words which contain the current character. Xia *et al.* (2016) used features similar to that of Zhang *et al.* (2012b) but employed a large-scale external lexicon word in generating extra lexicon features. Leng *et al.* (2016) further designed more features for character representation. The features include reduplication feature which indicates whether the character is in a reduplication forms of word (e.g., "棒棒哒" (great) and "哈哈哈" (sound of laugh)), conditional entropy feature defined by the entropy of all the characters that follow or precede the current character in the given corpus (Gao and Vogel 2010), etc. Nevertheless, the feature augmented methods using the traditional one-hot vector as the base of the character representation; the huge dimension of the one-hot vector still dominates the segmentation results and makes the effect of domain-specific features minimal. On the other hand, the neural-network-based methods boost the accuracy of Chinese word segmentation by using a large number of parameters in neural networks to fit the training data. For example, Zheng, Chen, and Xu (2013) used multilayer perception as the labeling engine. Chen *et al.* (2015) improved the segmentation accuracy by leveraging an LSTM neural network to capture the historical information in the Chinese sentences. Zhang, Zhang, and Fu (2016) integrated recurrent neural networks with the transition model in Zhang and Clark (2007). Concretely, they used a neural network model to replace the discrete linear model in Zhang and Clark (2007) for scoring transition action sequences. Qian, Qiu, and Huang (2016) introduced a new evaluation metric for Chinese word segmentation, the weights of the words are different

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

in the evaluation metric. Cai and Zhao (2016) proposed a gated combination neural network (GCNN) which decides how to mix the character vectors by two gates, then GCNN works with LSTM to calculate a score for each sentence segmentation, and beam search scheme is used to search for the segmentation with the highest score. Cai *et al.* (2017) designed a greedy neural word segmenter (greedyCWS), which improves the GCNN model by keeping a short list of frequent words, and decided how to mix character vectors according to the frequent word list.

However, there are several issues that make the neural network methods cannot detect out-domain new words precisely. First of all, the performances of neural-network-based methods rely on the quality of the training set heavily. A high-quality domain-specific training set is not always available. There are more and more out-domain applications. For example, due to the increasing usage and timely information on Twitter, the problem of identifying new words from Chinese twitter is now a critical application in many organizations including companies and governments. High-quality labeled training sets for Chinese twitter do not exist and new words emerge every day. Moreover, the high lexicon variance in Chinese microblog makes it difficult to design a domain-specific training set which can cover most of the topics. Second, neural-network-based methods take continuous bag-of-words (CBOW) embedding as the input. The rationale behind CBOW character embedding is to learn a 30–50-dimensional vector for each character in a large corpus through a neural network. Theoretically, the vectors can capture the grammatical and semantic meaning of the characters. Nevertheless, the characters in some new words are not grammatical and semantic related with each other. For example, person name and organization name are usually not formed by their semantic meaning. In addition, the learned character embedding is identical for each Chinese character such that it is not context-aware. Moreover, the statistical information in the target data, which is important information in out-domain new word detection, is not utilized by CBOW character vector. Furthermore, the representation is not interpretable; it is hard to identify the problem when an error happens. The enhancement of the CBOW Chinese character embedding (e.g., using Chinese radical[a]) cannot solve these issues efficiently (Sun *et al.* 2014).

2.2 Unsupervised Chinese new word detection

The unsupervised approach is purely data-driven; tagged training sets are not required in the unsupervised method. In unsupervised Chinese new word detection, the probability of a character sequence being a valid word is evaluated by the frequency distributions relevant to the character sequence. There are several assumptions about unsupervised valid word detection; one of the them is that if the given character sequence is a valid word, it should appear in different contexts. Accessory variety (AV) (Feng *et al.* 2005) is one of the methods that define word boundary probability according to this assumption. Assume there is a character sequence "门把手" (doorknob), which is a valid word; we can find it in different contexts such as "门把手坏了" (the doorknob is broken), "要一个新的门把手" (need a new doorknob), "或者把这个门把手修好" (or repair this doorknob), and "这个门把手很漂亮" (this doorknob is pretty). In this case, there are three different preceding characters of "门把手" (i.e., sentence start, "的", and "个"), as well as four different succeeding characters (i.e., "坏", sentence end, "修", and "很"). AV(门把手) takes the minimum of these two values, that is, 3. An obvious drawback of AV is that AV is affected by the number of occurrences of the character sequence. Frequent character sequences often have high AV values since they have more chances to appear in different contexts. *Branching entropy* (*BE*) addressed the problem of AV by using conditional probability. Another assumption used in unsupervised Chinese word detection is that if the given character sequence is a valid word, the substrings of the character sequence will mainly co-occur with the character sequence. For example, assume the given character sequence is "氨基酸" (Amino acids), which is a valid word; its substrings (i.e., "氨基","酸","氨","基酸") should mainly co-occur with "氨基酸". Unsupervised

[a] https://en.wikipedia.org/wiki/Radical_(Chinese_characters).

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

Chinese word segmentation approaches developed based on this assumption include symmetric conditional probability (SCP) (Luo and Sun 2003) and mutual information (MI) (Xue 2003). Other unsupervised Chinese word segmentation methods contain description length gain (DLG); Kityz, Chunyu, and Yorick (1999) measure the word probability of a character sequence using techniques in information theory; Huang *et al.* (2014) tried to integrate different unsupervised approaches into a unified framework, so on.

However, the performance of unsupervised Chinese new word detection is limited by the following issues. Firstly, unsupervised methods often involve a large number of parameters to be set manually. Secondly, Chinese new word which occurs with a low frequency is difficult to be identified correctly by unsupervised methods. The unsupervised methods detect Chinese words mainly based on the statistical analysis of the words and their neighbors. However, the statistics of infrequent words are not reliable. For instance, the *BE* value of character sequence occurs once is always 0 since it appears in one environment only. In this case, valid words which appear only once in the corpus cannot be detected correctly by using *BE*.

3. Contribution

In this paper, we focus on utilizing CRF in out-domain new word detection. To tackle the issues we mentioned, first, we introduce a new method of Chinese word boundary detection named *EL*. Compared with *BE*, *EL* improves Chinese word boundary detection by taking not only context variance but also context cohesion into consideration. Second, we propose a domain-independent Chinese new word detector *DICND*. In *DICND*, each Chinese character is mapped into a low-dimensional discrete vector using a statistical representation layer. The idea is to identify the characters abstractly without considering the known words, therefore enhancing the flexibility of the algorithm in new word detection.

To the best of our knowledge, it is the first work merely using segmentation relevant features to represent a character. *DICND* has the following advantages: first, it is domain independent. All of the characters in the documents are represented by their segmentation-related features. Second, the statistics-based character embedding is context-aware. Third, unlike using neural network approaches, the elements in the proposed character embedding are interpretable such that it is easy to trace when an error happens.

The proposed methods are evaluated in the following two aspects. We first evaluate the proposed Chinese word boundary detection method *EL* on SIGHAN Bakeoff; the experiment result shows *EL* can identify word boundaries more accurately than the widely used measure *BE*. Then we compared the out-domain new word detection performance of *DICND* with that of CRF with the one-hot vector (Zhang, Yasuda, and Sumita (2008)), CRF with CBOW character embedding, LSTM neural networks with CBOW character embedding (Liu *et al.* 2014), unsupervised methods, GCNN (Cai and Zhao 2016), and GreedyCWS (Cai *et al.* 2017). We train the classifiers based on segmented Chinese news; then we apply the trained classifier to microblog data for Chinese new word detection. The training set used in our experiment is the PKU training set in SIGHAN Bakeoff, and the test set is a microblog data set provided by NLPCC (Qiu, Qian, and Shi (2016)). Although the size of the test set is small, which is not a perfect setting for statistics-based character embedding, *DICND* still achieves the highest *F* score in these methods. We also identified a few examples to illustrate why *DICND* performs better than existing tools which would provide more insights to researchers in this field.

4. Word boundary detection by *EL*

4.1 Overview of EL

BE assumes if a character sequence frequently appears in different contexts, the character sequence is a valid word. However, this assumption does not work for all the character sequences in

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

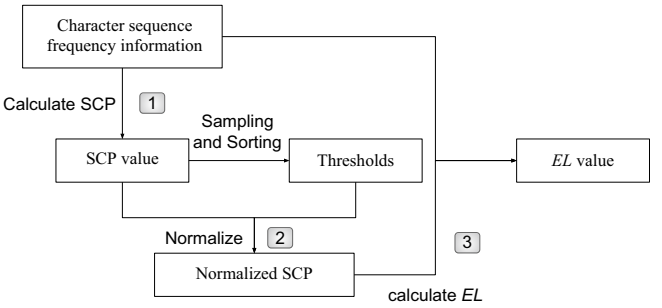

Figure 1. Process of calculating *EL* value.

Chinese word. For example, "国媒体" is a character sequence that has many different neighbors. Its preceding character can be "美(国)" (America), "中(国)" (China), "英(国)" (England), and its succeeding character can be "宣(称)" (claim), "报(道)" (report). The reason is that "国" is a component of many different words. *BE* only considers context variance without checking whether these contexts and the character sequence are tightly coupled. In this paper, we propose a new word boundary detection method named *EL* (*BE*). *BE* defines word boundaries of a character sequence not only based on the context variance but also based on the context cohesion. *SCP* is used to measure the cohesion of the character sequence and its neighbors. The process of generating *EL* is shown in Figure 1.

The process of *EL* calculation can be divided into three steps: calculating *SCP*, normalizing *SCP*, and calculating the final *EL* value.

4.2 Symmetric conditional probability

SCP measures the cohesiveness of a character sequence s according to the co-occurrence of the character sequences $c_1, \ldots, c_i$ and $c_{i+1}, \ldots, c_{|s|}$ $(1 < i < |s|)$. In general, *SCP* assumes if s is a valid word, the substrings of s will mainly appear along with s. For example, given sentence "氨基酸/是/构成/蛋白质/的/基本/单位" (Amino acids constitute the basic unit of protein), the character sequence "氨基酸"(Amino acids) is a valid word, and its substrings (i.e., "氨基";"酸";"氨";"基酸") should mainly co-occur with "氨基酸". The probability of the occurrence of s, denoted as $P(s)$, is the frequency of the character sequence in the given corpus in this case. The SCP value of s can be calculated by Equation (1):

$$SCP(s) = \frac{P(s)^2}{\frac{1}{|s|-1}\sum_{i=1}^{|s|-1} P(c_1, \ldots, c_i)P(c_{i+1}, \ldots, c_{|s|})} \tag{1}$$

$SCP(s)$ is high when all the binary segmentations of s mainly appear along with s, and the value of $SCP(s)$ is in $(-\infty, 1]$.

4.3 Postprocessing and normalization

The postprocessing of SCP value is mapping the raw *SCP* values to $\{0, \ldots, N\}$, N is a user-defined parameter. The postprocessing not only normalizes the values into $\{0, \ldots, N\}$ but also discretizes the numerical SCP values into $N + 1$ classes. Then the processed values can be input into CRF which can deal with discrete attributes.

The postprocessing of SCP values has three steps:

(1) Randomly select a set of items from the data set as the samples; sort the samples according to their descending SCP values.

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

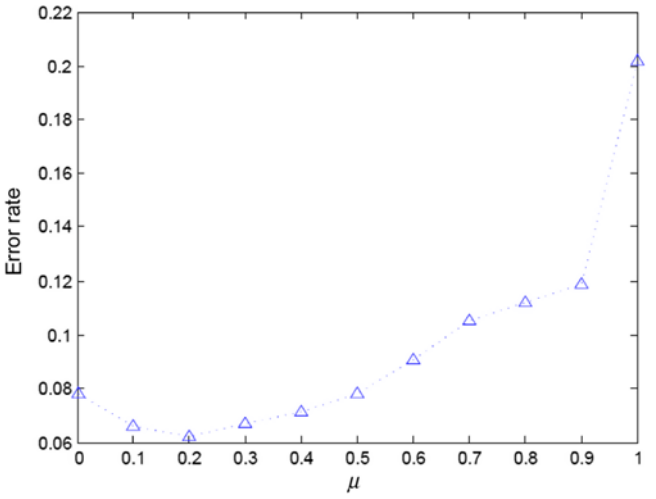

Figure 2. Effectiveness of μ in accuracy.

(2) Select cut points from the samples. The cut points of the bucket n ($n \in \{0, \ldots, N\}$) are the $\frac{n}{N}$th and the $\frac{n+1}{N}$th items in the sorted samples.
(3) Bin all the raw $SCP(s)$ in the $N+1$ buckets.

After that, high $SCP(s)$ values are mapped to large n, vice versa. It is worth noting that the statistical information of infrequent character sequences is not as reliable as that of frequent character sequences. In this case, we are more interested in the frequent character sequences. Specifically, infrequency character sequences (e.g., character sequences appear only once in the corpus) should be filtered in the samples.

4.4 Edge Likelihood

EL is composed of *Left EL* and *Right EL*. *Left EL* of s is calculated based on the frequency distribution of s and the cohesion of s with its preceding characters. Let p denote a preceding character of s; $p+s$ is a string concatenating p and s. The cohesion of $p+s$ is evaluated by $SCP'(p+s)$ (Equation (2)):

$$\varphi(p,s) = \frac{N - SCP'(p+s)}{N} \tag{2}$$

The value of $\varphi(p,s)$ is in $[0,1]$. High $\varphi(p,s)$ indicates the connection between p and s is tight such that p should contribute less to $EL_{left}(s)$. Let $\beta(p,s)$ denote a parameter that indicates the importance of p to $EL_{left}(s)$ as:

$$\beta(p,s) = \mu + (1-\mu)\varphi(p,s) \tag{3}$$

The parameter μ is a lower bound of $\beta(p,s)$ which is to ensure every p can have certain importance to $EL_{left}(s)$. The value of μ is in $[0,1]$, and $\mu = 0$ means we will ignore p in calculating $EL(s)$ if $SCP'(p+s) = N$ (the cohesion of p and s is very strong). $BE_{left}(s)$ can be treated as a special case that $\mu = 1$ in $EL_{left}(s)$ calculation, which means considering each p equally regardless of the $SCP'(p+s)$. The effectiveness of different μ is shown in Figure 2; the highest accuracy is achieved when $\mu = 0.2$.

$EL_{right}(s)$ is the probability that the left boundary of s is a valid word boundary. The value of $EL_{left}(s)$ is defined by the number of preceding characters of s as well as the cohesion of s and its

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

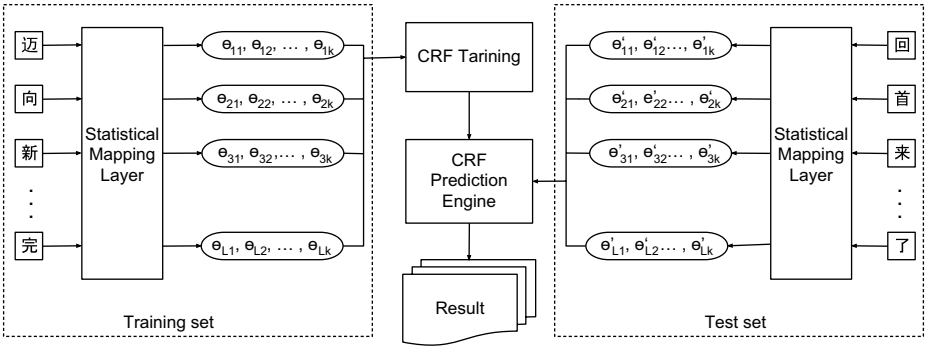

Figure 3. DICND overview.

preceding character (Equation (4)):

$$EL_{left}(s) = -\sum_{p \in \mathcal{P}_s} \beta(p, s) P(p|s) \log P(p|s) \tag{4}$$

where $\mathcal{P}_s$ denotes the set of preceding characters of s.

The $EL_{right}(s)$ can be calculated similarly. The value of $EL(s)$ is defined by Equation (5):

$$EL(s) = \min \{EL_{left}(s), EL_{right}(s)\}. \tag{5}$$

Given a character sequence s, $EL(s)$ evaluates the probabilities of the boundaries of s are valid word boundaries not only based on whether s occurs in different contexts but also based on how s connects to its neighbors. Take the wrong segmentation "国媒体" (country media) mentioned above as an example; this character sequence does appear in different contexts, but it often tightly connects to its preceding characters, for example, "美国" (America) and "中国" (China). In other words, the β values of the preceding characters of "国媒体" are low such that the overall EL(国媒体) should not be too high.

5. Domain-independent Chinese new word detector

5.1 Overview of domain-independent Chinese new word detector

In this section, we introduce a domain-independent Chinese new word detector, named *DICND*, which tries to address the issues of the current CRF-based new word detection by using a statistics mapping layer. After an unsupervised pre-trained process, each character in the documents is represented as a low-dimension discrete vector which reflects the segmentation-related information of the character. Figure 3 shows an overview of *DICND*.

5.2 Statistics-based character embedding

The statistics mapping layer is used to embed the Chinese characters into low-dimensional vectors. The segmentation-related statistical features, as well as the POS attributes of the characters and their neighbors, are leveraged as the features of the characters. We follow Peng *et al.* (2004) and categorize the features into closed feature and open feature.

The closed features are obtained from the training data alone. From our study, we notice *EL* defines word boundary by context variance while *SCP* using inner cohesiveness of the character sequence. They can work together to represent the characteristics of a character sequence from two

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

Table 1. Closed features of c_0

Measure	Character sequence			
	c_0	c_0c_1	$c_0c_1c_2$	$c_0c_1c_2c_3$
$Freq'$	√	√	√	√
SCP'		√	√	√
EL'_{left}	√	√	√	
EL'_{right}	√	√	√	
MI'_{left1}	√	√	√	
MI'_{left2}	√	√	√	
MI'_{right1}	√	√	√	
MI'_{right2}	√	√	√	

Table 2. Open features of c_0

Measure	Character sequence			Measure	Character sequence		
	c_0	c_0c_1	$c_0c_1c_2$		c_0	c_0c_1	$c_0c_1c_2$
isDictWord	√	√	√	isStopWord	√	√	√
isVt	√	√	√	isParticle	√	√	
isPreposition	√	√		isPosition	√	√	
isAdj	√	√		isAdv	√	√	
isQuantifier	√	√		isPronoun	√	√	
isConj	√	√		isNumber	√	√	
isPrefix	√			isInterj	√		

different aspects. Other than *EL* and *SCP*, we further use *MI* to evaluate the association degree of the target character sequence and its surroundings. *MI* is defined as in Equation (6):

$$MI(s_{surd}, s) = \log_2 \frac{P(s_{surd}|s)}{P(s_{surd})P(s)} \tag{6}$$

where $p(s_{surd}|s)$ is the probability of co-occurrence of s_{surd} and s. In the proposed character embedding, the s_{surd} of character sequence $s = c_0, \ldots, c_{|s|}$ is $c_{-2}c_{-1}$, c_{-1}, $c_{|s|+1}$, and $c_{|s|+1}c_{|s|+2}$.

It is worth mentioning *SCP* is utilized to calculate the inner cohesiveness of the s in the closed features, while in the calculation of *EL*, the *SCP* is used to evaluate the connection of s and its neighbors (outer cohesiveness).

The closed features of a character c_0 are listed in Table 1.[b] The "′" symbol means the values are normalized (discretized).

On the other hand, the open features (Table 2) are generated according to knowledge base other than the training set, namely a Chinese word list, a stop word list, and several POS character lexicons from various sources. The details are in Table 5. The POS or lexical attributes of a character sequence are obtained by checking the existence of the character sequence in the specific list (e.g., the boolean value is *Preposition*(s) is obtained by checking if s exists in the list of the preposition).

[b]For a character sequence $c_0, \ldots, c_l$, $MI_{left1} = MI(c_{-1} : c_0, \ldots, c_l)$, $MI_{left2} = MI(c_{-2}c_{-1} : c_0, \ldots, c_l)$, $MI_{right1} = MI(c_0, \ldots, c_l : c_{l+1})$, $MI_{right2} = MI(c_0, \ldots, c_l : c_{l+1}c_{l+2})$.

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

Table 3. Character tags

Tag	Description
P	Punctuation (i.e., “ ” ”) and special characters
S	Single character as a word
B	Word beginning
M	Middle of word
E	Word end

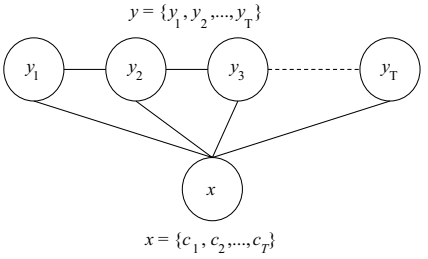

Figure 4. Linear chain CRF.

Most of the valid Chinese words in the knowledge base consist of one or two characters. Thus, in most of the features, we only consider up to two characters.

5.3 Character tagging

The statistics-based character embedding will be input into the CRF classifier, and five tags are used as the segmentation labels (Table 3).

Linear chain CRF is used as the tagging algorithm (Figure 4).

Given a sentence x with T characters, $x = c_1, c_2, \ldots, c_T$, each character in the sentence is represented by its statistical embedding vector. The character vector of the tth character serves as the observed variable at the current time step, t. The tag of the tth character, denotes as y_t, is related to the observed variable of t and the tag of its preceding character, that is, y_{t-1}. Let y be the label of the sequence, CRF defines y by Equation (7):

$$p_\Lambda(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t) \right\} \tag{7}$$

The model parameters are a set of real weights $\Lambda = \{\lambda_k\}$, one weight for each feature, and $\{f_k(y, y', t)\}_{k=1}^{K}$ is a set of binary-valued indicator function reflecting the transitions between y_{t-1} and y_t, K is the total number of feature functions. The feature functions can measure a state transition $y_{t-1} \to y_t$ and the entire observation sequence, x, centered at t. For example, one possible feature function could measure how much we suspect that the current word should be labeled as “B” given that the previous character is an adjective. The value of x_t is the statistical embedding vector of c_t. Large positive values for λ_k indicate a preference for such an event; large negative values make the event unlikely.

$Z(x)$ is a normalization factor over all state sequences for the sequence x:

$$Z(x) = \sum_{y} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t) \right\} \tag{8}$$

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

Table 4. SIGHAN Bakeoff

	Dataset	Size (sentences)
SIGHAN Bakeoff	MSRA	80,000
	CityU	53,000
	CTB	24,000
	PKU	19,000

The most probable labeling sequence for an input x is

$$\hat{y} = \arg\max p_{\Lambda}(y|x) \tag{9}$$

6. Experiment

In this section, we evaluate the performance of *EL* and *DICND*. We first compare the effectiveness of *EL* and *BE* using the data sets in SIGHAN Bakeoff. Second, we train *DICND* based on PKU data set or CTB data set which contains mainly news. Then the classifier is applied to a microblog data set to get the out-domain new word detection result. We compared the new word detection result with that of CRF with one-hot vector, CRF with CBOW character embedding, LSTM with CBOW character embedding, GCNN with CBOW character embedding, GreedyCWS, and an unsupervised method.

6.1 Data set

There are several data sets and lexicon data used in our experiment:

- **SIGHAN Bakeoff** (Sproat and Emerson 2003) is the most widely used data set in Chinese word segmentation. There are four sub data sets in SIGHAN Bakeoff, namely PKU, CityU, MSR, and CTB6. The data are collected from newspapers (e.g., China Daily, South China Morning Post) such that the sentences in the data sets are formal. Some of the statistics of SIGHAN Bakeoff are shown in Table 4.
- **NLPCC microblog data set** (2016) contains 6000 segmented Chinese tweets. The topics of the tweets include politics, weather, music, sports, etc.
- **Sogou Web corpus**[c] contains 6,000,000 unlabeled Chinese sentences from web data.
- **Dictionary** is the dictionary used in Stanford Word Segmenter[d] which contains 423,000 Chinese words.
- **Word Lists** are obtained from `http://xh.5156edu.com`; the detailed information of the word lists is listed in Table 5.

6.2 EL experiments

Theoretically, the $EL(s)$ is linear with the probability of the boundaries of s, which are valid word boundaries. In other words, the proportion of valid character sequence boundaries in $\{s|EL'(s) = n\}$ should be $I(n) = \frac{n}{N}$ in the ideal case. For instance, assume $n = 2$ and $N = 10$, $2/N = 20\%$ of s in $\{s|EL'(s) = 2\}$ should be character sequence with valid word boundary. In this experiment,

[c] `http://www.sogou.com/labs/dl/c.html`.
[d] `https://nlp.stanford.edu/software/segmenter.shtml`.

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

Table 5. Dictionary

Dictionary	Size (words)	Dictionary	Size (words)
Stop word list	1300	Chinese numbers	43
Verb list	1578	Particle list	172
Preposition list	182	Position list	45
Adjective list	105	Adverb list	1337
Quantifier list	402	Pronoun list	178
Prefix list	7	Interjection list	151
Conjunction list	401		

Table 6. Comparison of EL and BE on PKU data set

Rank	Valid ratio			Error rate	
	BE	*EL*	Ideal	*BE*	*EL*
0	0.1204	0.1204	0	**0.1204**	**0.1204**
1	0.2892	0.0657	0.125	0.1642	**0.0593**
2	0.3602	0.1335	0.25	**0.1102**	0.1165
3	0.5962	0.3313	0.375	0.2212	**0.0437**
4	0.8105	0.5037	0.5	0.3105	**0.0037**
5	0.3929	0.619	0.625	0.2321	**0.006**
6	0.4652	0.7229	0.75	0.2848	**0.0271**
7	0.6793	0.8082	0.875	0.1957	**0.0668**
8	0.7668	0.8826	1	0.2332	**0.1174**
Average	/	/	/	0.2340	**0.0623**

we define the error rate as the deviation between the real valid boundary ratio and the ideal valid boundary ratio. Denote the set of s with valid word boundary as S_{valid}, the valid ratio of s in $\{s|EL'(s) = n\}$ as $R(n)$, $R(n) = \frac{|s|EL'(s)=n \& s \in S_{valid}|}{|s|EL'(s)=n|}$. The error rate is the absolute value of $I(n) - R(n)$.

We evaluate the *EL* and *BE* value of the character sequences which contain 2–5 tokens in the PKU data set. The result is shown in Table 6. The *EL* and *BE* values are normalized into $\{0, \ldots, 8\}$ (i.e., N is set as 8). Note that for character sequence "上海队 教练" (coach of Shanghai team), it contains valid boundary for not only "上海队" (Shanghai team) and "教练" (coach) but also "上海队教练" (coach of Shanghai team). The results which have lower error rate are bold.

According to Table 6, the error rate of *EL* is significantly lower than that of *BE*, which means the *EL* value is closer to the ideal case compared with the *BE* value. We can see if we set the value of the threshold to 8, 88.26% of s in $\{s|EL(s) = 8\}$ are character sequences with valid word boundaries while that of *BE* is 76.88% (the value should be 100% in the ideal case).

Similarly, we evaluated the error rates of *EL* and *BE* on data set MSRA, CityU, and CTB (Table 7). The results which have lower error rate are bold. From the table, we can see the error rate of *EL* value is about 10% less than that of *BE* value. In other words, *EL* is closer to the ideal case and can detect word boundaries more accurate than *BE*.

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

Table 7. Comparison of EL and BE on SIGHAN Bakeoff

Data set	*BE* error rate	*EL* error rate
PKU	0.2340	**0.0623**
MSRA	0.1808	**0.1153**
CityU	0.1607	**0.0805**
CTB	0.1254	**0.1175**

6.3 DICND experiments

In this section, we compared the performance of Chinese out-domain new word detection of *DICND* with several baselines. The baselines include one-hot character vector + CRF (Zhang *et al.* 2008),[e] CBOW character vector + CRF, CBOW character vector + LSTM (Liu *et al.* 2014),[f] CBOW character vector + GCNN (Cai and Zhao 2016),[g] greedyCWS (Cai *et al.* 2017),[h] and an unsupervised model *MI* + *BE*.[i]

Concretely, we train the classifiers on PKU or CTB data set which contains mainly news. Then the classifier is applied to microblog data to get the out-domain new word detection result. The definition of new words in this experiment is the words in the microblog data set but not in PKU data set and our knowledge base. Infrequency new words (words appear once only) are excluded since their statistical information are unreliable. Non-Chinese characters and character sequences containing stop words are also excluded since they are often not the interest of Chinese new word detection. There are 847 valid new words in the microblog data set according to our definition of the new words.

6.3.1 Performance comparison

The new word detection result of *DICND* and the baselines is shown in Table 9. The experiment setting is as follows. The unsupervised model is a combination of term frequency, *MI* and *BE*.[j] For any s, $l = |s|$, if $Freq(s) > 1$, $\sum_{i=0}^{n-1} MI(c_0, \ldots, c_i : c_{i+1}, \ldots, c_l) > 100$ and $BE(s) > 0.5$, s will be considered as a valid word. The thresholds are optimized by trying different combination of the values. The CBOW character embedding is pre-trained with Sogou web data; the number of dimension of CBOW character embedding is 40, while the number of dimension of the proposed statistic character embedding is 41 (25 closed features in Table 1, and grouped features in Table 2 into 16 attributes). We use the code at `https://github.com/FudanNLP/CWS_LSTM` as the implementation of CBOW character embedding with LSTM. Note that the formula of F score is

$$F = \frac{2PR}{P+R}, \tag{10}$$

where P and R are the precision and recall, respectively.

For the baseline "CRF + sparse vector" we use the pre-trained model "pku.gz" in the Stanford Word Segmenter.[k] The training time of other supervised methods is in Table 8.[l]

[e] `https://nlp.stanford.edu/software/segmenter.shtml`.
[f] `https://github.com/FudanNLP/CWS_LSTM`.
[g] `https://github.com/jcyk/CWS`.
[h] `https://github.com/jcyk/greedyCWS`.
[i] `https://github.com/qiaofei32/new-word-recognition`.
[j] We used the code at `https://github.com/qiaofei32/new-word-recognition`.
[k] `https://nlp.stanford.edu/software/segmenter.shtml`.
[l] All the codes in the experiments run on a machine with a 3.40 GHz i5-3570 CPU, 16 GB main memory

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

Table 8. Training time comparison

Method	Training time
CRF + CBOW character embedding	31 min
LSTM + CBOW character embedding	9 h 44 min
DICND	41 min

Table 9. *F* score comparison

Method	*P* (%)	*R* (%)	*F*
Unsupervised model	36.39	18.29	0.2435
CRF + one-hot vector	**71.03**	24.32	0.3642
CRF + CBOW character embedding	53.54	39.32	0.4534
LSTM + CBOW character embedding	40.40	**73.08**	0.5204
GNCC + CBOW character embedding	46.65	63.28	0.5371
GreedyCWS	53.08	53.84	0.5346
DICND	49.70	68.71	**0.5768**

The neural-network-based methods involve a large number of parameters such that require much more training time than that of CRF based methods.

Table 9 shows the *F* score of new word detection with *DICND* and the scores of baseline methods. The best results are bold. From the table, we can see *DICND* achieves the highest *F* score among the methods. CRF with sparse representation has the highest accuracy because of the high-dimensional representation of each character, but the recall rate is low due to the representation also restrains the flexibility of the algorithm.

In general, long new words are more difficult to be detected correctly since they require more labels to be categorized. Thus, new word detection on short words often has a higher recall and precision than that of long new words. In this part, we compare the out-domain new word detection results of different word lengths. Table 10 shows the *F* score of *DICND* and the baseline methods with different length of character sequences. In Table 10, "Gold" is the gold standard provided by the data set, "Detect" is the number of new words detected by the method, and "Valid" is the number of valid new words detected by the method. The best results in their categories are bold.

In addition, we analyzed the out-domain Chinese new word detection results of different training set. Specifically, other than PKU data set, we also train the proposed classifier on CTB data set. The result is in Table 11. The best results in their categories are bold. The overall *F* score of classifier trained with PKU data set and that of CTB data set are similar, but the classifier trained with CTB has better performance on recall but lower precision compared with that of PKU.

6.3.2 Case studies

In this section, we identified a few cases in our experiments to verify our observation on the tools.

The CRF with sparse representation does not achieve good performance in compound words detection. For instance, the word "贪吃蛇" (A game called "Greedy snake") in the microblog data set is a combination of two known words "贪吃" (greedy) and "蛇" (snake). With the sparse representation, the algorithm segments "贪吃蛇" (Greedy snake) into two words rather than considering it as one word.

The CBOW character embedding using an identical vector represents a specific character without considering its surrounding context. For instance, no matter the character "卡" appears in the

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

Table 10. *F* score comparison of different new word length

Method		Character sequence length			
		2	3	4	5
	Gold	519	299	23	6
	Detect	267	81	6	2
Unsupervised model	Valid	102	38	1	0
	P (%)	38.20	46.91	16.67	0
	R (%)	19.65	12.71	4.35	0%
	F	0.2595	0.2	0.069	0
	Gold	519	299	23	6
	Detect	234	52	3	1
CRF + one-hot vector	Valid	158	45	2	1
	P (%)	**67.52**	**86.54**	**66.67**	**100**
	R (%)	30.56	15.10	8.70	16.67
	F	0.4187	0.2571	0.1539	0.2858
	Gold	519	299	23	6
	Detect	516	99	4	3
CRF + CBOW character embedding	Valid	277	55	1	0
	P (%)	53.68	55.56	25	0
	R (%)	53.58	18.46	4.35	0
	F	0.5363	0.2722	0.0371	0
	Gold	519	299	23	6
	Detect	892	339	64	9
LSTM + CBOW character embedding	Valid	**398**	163	12	3
	P (%)	44.62	48.08	18.75	33.33
	R (%)	**76.69**	54.51	52.17	50.00
	F	0.5641	0.5110	0.2759	**0.4000**
	Gold	519	299	23	6
	Detect	879	238	32	/
GNCC + CBOW character embedding	Valid	378	148	10	/
	P (%)	43.00	62.18	31.25	/
	R (%)	72.83	49.50	43.48	/
	F	0.5408	0.5512	**0.3636**	/
	Gold	519	299	23	6
	Detect	547	249	63	/
GreedyCWS	Valid	298	146	12	/
	P (%)	54.48	58.63	19.05	/
	R (%)	57.42	48.83	52.17	/
	F	0.5591	0.5328	0.2791	/

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

Table 10. Continued

Method		Character sequence length			
		2	3	4	5
	Gold	519	299	23	6
	Detect	736	290	118	27
DICND	Valid	387	**174**	**17**	**4**
	P (%)	52.58	60.0	14.41	14.81
	R (%)	74.57	**58.19**	**73.91**	**66.67**
	F	**0.6167**	**0.5908**	0.2411	0.2424

Table 11. *F* score comparison of different training set

Data set	*P* (%)	*R* (%)	*F*
PKU	**49.70**	68.71	**0.5768**
CTB	45.34	**76.39**	0.5690

word “礼品卡” (gift card) or the word “卡夫卡” (Kafka, a writer) the character vector of “卡” has no difference. In our experiment, only the new word “礼品卡” (gift card) is detected by CRF with CBOW character embedding while *DICND* can identify both of the “礼品卡” (gift card) and “卡夫卡” (Kafka). This drawback of CBOW character embedding can be partially solved by using a neural network, such as LSTM, as a label classifier. Since the neural network will modify the feature representation automatically. However, the neural network defines the label of a character only based on the character and its several neighbor characters. In other words, the neural-network-based approach lacks overview of the whole sentence which might make it less competitive with CRF which predicts sequences of labels for sequences of input samples. Compared with *DICND*, the GNCC model (Cai and Zhao 2016) has a stronger performance on formal word detection, for example, “资金链” (capital chain), “艺考” (art exam). *DICND*, meanwhile, can detect more person name or internet slang words, for example, “广场舞” (open-air fitness dancing), “卖萌” (acting cute). This indicates *DICND* is more adaptive to the target domain (i.e., tweets corpus in this experiment). The reason is that the statistics-based embedding of a character in the test set is calculated according to the frequency distribution of the character sequence in test data, thus the proposed embedding can utilize the statistics in the target domain. On the other hand, the CBOW character vector is pre-trained such that the statistics of the target data cannot be used to improve the word segmentation result. The domain adaption capability of GNCC is improved in greedyCWS. However, DICND still performs better in detecting names of people or organizations in the target data.

One of the reasons that the unsupervised method does not have a good performance is that the test data used in our experiment contains just 6000 sentences, and the sentences are on different topics. Actually, 95% of the 2–5-gram character sequences appear only once or twice in the test set. The statistical information of infrequent character sequences is unreliable. For instance, “亚奥” (short for Asian Olympic) is a new word failed to be detected by the unsupervised method. The character sequence appears twice in the document; its succeeding character sequence is “理事会” (Council) for both of the occurrences which make the *BE* value equals to 0 since it always appears in the same environment in test data. On the other hand, with supervised machine learning mechanism, *DICND* can detect “亚奥” successfully. But the insufficiency of test data also hinders the performance of *DICND*, especially for long word detection.

https://doi.org/10.1017/S1351324918000463 Published online by Cambridge University Press

7. Conclusion

In this paper, we first proposed *EL*, a novel method for Chinese word boundary detection. *EL* defines the probability of the boundaries of a character sequence that are valid word boundaries based on both of context variance and of context cohesion of the character sequence. Our experiment shows *EL* can detect Chinese character sequence boundaries better than the widely used *BE*. Second, we designed *DICND*, which is a domain-independent CRF-based Chinese new word detector. Each Chinese character is represented as a low-dimensional vector in *DICND*; the values in the character vector are defined by segmentation-related features of the corresponding character. The experiment on out-domain new word detection shows *DICND* can significantly outperform existing methods. However, the performance of *DICND* is affected by the size of test data. This is because the proposed character embedding is generated based on the distributions of the character sequences in the given corpus, but the statistics of infrequent character sequences are unreliable.

Although *DICND* shows improvement over existing methods, the performance of out-domain new word detection still has a large room for improvement. We hope our work can provide insights into the problem.

References

Cai, D. and Zhao, H. (2016). Neural word segmentation learning for Chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin: Association for Computational Linguistics (ACL), pp. 409–420.

Cai, D., Zhao, H., Zhang, Z., Xin, Y., Wu, Y. and Huang, F. (2017). Fast and accurate neural word segmentation for Chinese. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vancouver: Association for Computational Linguistics (ACL), pp. 608–615.

Chang, P.C., Galley, M. and Manning, C.D. (2008). Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, Madison: Omnipress Inc., pp. 224–232.

Chen, X., Qiu, X., Zhu, C., Liu, P. and Huang, X. (2015). Long short-term memory neural networks for Chinese word segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Lisbon: Association for Computational Linguistics (ACL), pp. 1197–1206.

Eddy, S.R. (1996). Hidden Markov models. *Current Opinion in Structural Biology*, **6**, 361–365.

Feng, H., Chen, K., Kit, C. and Deng, X. (2004). Unsupervised segmentation of Chinese corpus using accessor variety. In *International Conference on Natural Language Processing*, India: NLP Association of India, pp. 694–703.

Gao, Q. and Vogel, S. (2010). A multi-layer Chinese word segmentation system optimized for out-of-domain tasks. In *Proceedings of CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP2010)*, Beijing, Chinese Information Processing Society of China, pp. 210–215.

Huang, M., Ye, B., Wang, Y., Chen, H., Cheng, J. and Zhu, X. (2014). New word detection for sentiment analysis. In *ACL (1)*, Baltimore: Association for Computational Linguistics (ACL), pp. 531–541.

Jin, Z. and Tanaka-Ishii, K. (2006). Unsupervised segmentation of Chinese text by use of branching entropy. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, Sydney: Association for Computational Linguistics (ACL), pp. 428–435.

Kityz, C. and Wilks, Y. (1999). Unsupervised learning of word boundary with description length gain. In *Proceedings of the CoNLL99 ACL Workshop*, Bergen: Association for Computational Linguistics (ACL), pp. 1–6.

Lafferty, J., McCallum, A. and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, San Francisco, Morgan Kaufmann Publishers Inc., pp. 282–289.

Leng, Y., Liu, W., Wang, S. and Wang, X. (2016). A feature-rich CRF segmenter for Chinese micro-blog. In *International Conference on Computer Processing of Oriental Languages*, Kunming, Springer LNAI, pp. 854–861.

Li, Y., Li, W., Sun, F. and Li, S. (2015). Component-enhanced Chinese character embeddings. arXiv preprint arXiv:1508.06669.

Liu, Y., Zhang, Y., Che, W., Liu, T. and Wu, F. (2014). Domain adaptation for CRF-based Chinese word segmentation using free annotations. In *EMNLP*, Doha: Association for Computational Linguistics (ACL), pp. 864–874.

Luo, S. and Sun, M. (2003). Two-character Chinese word extraction based on a hybrid of internal and contextual measures. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, Sapporo: Association for Computational Linguistics, vol. 17, 24–30.

McCallum, A., Freitag, D. and Pereira, F.C. (2000). Maximum entropy Markov models for information extraction and segmentation. In *ICML*, California, Morgan Kaufmann Inc., vol. 17, pp. 591–598.

Miao, C.-J. and Chen, X.-M. (2011) *The Interpretation of Modern Chinese Verbs*. Beijing Normal University Press, pp.3–22.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pp. 3111–3119.

Pei, W., Ge, T. and Chang, B. (2014). Max-margin tensor neural network for Chinese word segmentation. In *ACL (1)*, Baltimore: Association for Computational Linguistics (ACL), pp. 293–303.

Peng, F., Feng, F. and McCallum, A. (2004). Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th international conference on Computational Linguistics*, Barcelona: Association for Computational Linguistics (ACL), p. 562.

Qian, P., Qiu, X. and Huang, X. (2016). A new psychometric-inspired evaluation metric for Chinese word segmentation. In *Proceedings of the 54th international conference on Computational Linguistics*, Berlin: Association for Computational Linguistics (ACL), vol. 1, pp. 2185–2194.

Qiu, X., Qian, P. and Shi, Z. (2016). Overview of the NLPCC-ICCPOL 2016 shared task: Chinese word segmentation for micro-blog texts. In *International Conference on Computer Processing of Oriental Languages*, Kunming: Springer LNAI, pp. 901–906.

Sproat, R. and Emerson, T. (2003). The second international Chinese word segmentation bakeoff. In *Proceeding of the Sighan Workshop on Chinese Language*, Sapporo: Association for Computational Linguistics, pp.133–143.

Sun, Y., Lin, L., Yang, N., Ji, Z. and Wang, X. (2014). Radical-enhanced Chinese character embedding. In *International Conference on Neural Information Processing*, Montreal: Neural Information Processing Systems Foundation, Inc., pp. 279–286.

Wang, L.Y., Wong, F., Chao, S. and Xing, J.W. (2012). CRFs-based Chinese word segmentation for micro-blog with small-scale data. In *Association for Computational Linguistics*, Tianjin: Association for Computational Linguistics, pp. 51–57.

Wang, Y., Jun'ichi Kazama, Y.T., Tsuruoka, Y., Chen, W., Zhang, Y. and Torisawa, K. (2011). Improving Chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In *IJCNLP*, Chiang Mai: Asian Federation of Natural Language Processing, pp. 309–317.

Xia, Q., Li, Z., Chao, J. and Zhang, M. (2016). Word segmentation on micro-blog texts with external lexicon and heterogeneous data. In *International Conference on Computer Processing of Oriental Languages*, Kunming: Springer LNAI, pp. 711–721.

Xue, N. (2003). Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing* **8**(1), 29–48.

Yao, Y. and Huang, Z. (2016). Bi-directional LSTM recurrent neural network for Chinese word segmentation. In *International Conference on Neural Information Processing*, Barcelona: Neural Information Processing Systems Foundation, Inc., pp. 345–353.

Zhang, H.P., Yu, H.K., Xiong, D.Y. and Liu, Q. (2003). HHMM-based Chinese lexical analyzer ICTCLAS. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, Sapporo: Association for Computational Linguistics, vol. 17, pp. 184–187.

Zhang, K., Sun, M. and Zhou, C. (2012a). Word segmentation on Chinese microblog data with a linear-time incremental model. In *Second CIPS-SIGHAN Joint Conference on Chinese Language Processing*, Tianjin: Association for Computational Linguistics, pp. 41–46.

Zhang, M., Deng, Z., Che, W. and Liu, T. (2012b). Combining statistical model and dictionary for domain adaption of Chinese word segmentation. *Journal of Chinese Information Processing* **26**(2), 8–12.

Zhang, M., Zhang, Y. and Fu, G. (2016). Transition-based neural word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin: Association for Computational Linguistics (ACL), pp. 421–431.

Zhang, R., Yasuda, K. and Sumita, E. (2008). Chinese word segmentation and statistical machine translation. *ACM Transactions on Speech and Language Processing (TSLP)* **5**(2), 4.

Zhang, Y. and Clark, S. (2007). Transition-based parsing of the Chinese Treebank using a global discriminative model. In *IWPT '09 Proceedings of the 11th International Conference on Parsing Technologies*, Paris: Association for Computational Linguistics (ACL), pp. 162–171.

Zheng, X., Chen, H. and Xu, T. (2013). Deep learning for Chinese word segmentation and POS tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Seattle: Association for Computational Linguistics (ACL), pp. 647–657.

Cite this article: Liang Y, Yang M, Zhu J, and Yiu S. M (2019). Out-domain Chinese new word detection with statistics-based character embedding. *Natural Language Engineering* **25**, 239–255. https://doi.org/10.1017/S1351324918000463