New Word Detection Using BiLSTM+CRF Model with Features

Jianyong DUAN†,††(a), Member, Zheng TAN†,††(b), Mei ZHANG†(c), and Hao WANG†,††(d), Nonmembers

SUMMARY With the widespread popularity of a large number of social platforms, an increasing number of new words gradually appear. However, such new words have made some NLP tasks like word segmentation more challenging. Therefore, new word detection is always an important and tough task in NLP. This paper aims to extract new words using the BiLSTM+CRF model which added some features selected by us. These features include word length, part of speech (POS), contextual entropy and degree of word coagulation. Comparing to the traditional new word detection methods, our method can use both the features extracted by the model and the features we select to find new words. Experimental results demonstrate that our model can perform better compared to the benchmark models.

key words: new word detection, BiLSTM

1. Introduction

With the rapid development of the Internet, more and more social platforms begin to appear. When some news or new topic discussed by users, there will be a lot of new words about current events appear. According to statistics, more than 1000 new words are detected every year. These new words haven't appeared and also don't show up in the dictionary, but now they are frequently used.

However, such new words have made many natural language processing tasks more challenging, such as Chinese word segmentation, and named entity extraction [1]. There was a study [2] that has shown that more than 60% of word segmentation errors result from new words. Therefore, new word detection is an important task in NLP research.

Unknown words cause segmentation errors in that these out-of-vocabulary words in input text are often incorrectly segmented into single-character or other overly-short words [3]. Therefore, new words tend to be the wrong word segmentation and from multiple word fragments. For example, the new word "柠檬精(The person who always jealous of others)" is formed by: "柠檬(lemon)") and "精(demon)". Because these two words often come together, their combination has the meaning and a new word is formed in that way.

In that case, what kind of word fragments might form the new word becomes the crucial question. Relevant experts extracted some features of these word fragments such as part of speech, word frequency and so on. Then they discovered new words by summarizing the rules of these features, but it took a lot of time and energy to finish as well. Therefore, some methods which needn't handcraft features but use the model to extract features automatically were proposed.

However, there are limitations in both manual feature extraction and model feature extraction. If we only extract features by handcraft, much time and energy will be poured. If we only extract features by model, some crucial messages may be missed. So with that condition, we combine the model and handcraft to fulfill our task. Add some features selected by us first and then train the model to find new words. A better result will be come out in that way. The BiLSTM+CRF model which added some features extracted by us is chosen for our work.

We select words as the input unit to the model instead of characters, each word corresponds to a tag. As Table 1 shows, since new words will be segmented into single-character or other overly-short words, our task is not to label which one is a new word, but to label which word is the part of a new word. If a word and the following word are two parts of a new word, the combination of the two words will be called a new word, and in the opposite case, there is no new word come. We define two labels, one is 'Y' which means the word can combine with the word after it, the other is 'N' which means the word can't. In the following text, we will introduce our method in detail.

The rest of our paper is organized as follows: in Sect. 2, we will discuss the related work. in Sect. 3, The BiLSTM+CRF model will be introduced. Section 4 will show how the experience proceeds and our experimental results on Weibo data sets. Section 5 will draw the conclusion.

<table>
<thead>
<tr>
<th>Table 1 The tag of new word</th>
</tr>
</thead>
<tbody>
<tr>
<td>word</td>
</tr>
<tr>
<td>English</td>
</tr>
<tr>
<td>label</td>
</tr>
</tbody>
</table>

---

Manuscript received December 19, 2019.
Manuscript revised April 22, 2020.
†The authors are with College of Informatics, North China University of Technology, Beijing, China.
††The authors are with the CNONIX National Standard Application and Promotion Lab, Beijing, 100144 China.
a) E-mail: duan.jianyong@163.com
b) E-mail: 18601947784@163.com
c) E-mail: zm@ncut.edu.cn
d) E-mail: wanghaomails@gmail.com (Corresponding author)
DOI: 10.1587/transinf.2019EDP7330

Copyright © 2020 The Institute of Electronics, Information and Communication Engineers
2. Related Work

The new word detection as a traditional NLP task has been investigated for a long time. Among these researches, new word detection methods mainly contain four directions: The first one conducts the Chinese word segmentation and new word detection jointly. The second one uses linguistic rules and related knowledge summarized by specialists to match new words. The third one takes users behavior data into account to detect new words. The fourth one uses statistical features to find new words. We will introduce these four branches in detail on the following content of this part.

2.1 The Word Segmentation and New Word Detection Jointly

New word detection has been usually interweaved with word segmentation. Word segmentation can provide resources for new word detection. A typical method [4] used the CRF model to do word segmentation task, and words with high confidence but not considered as a word were added into the dictionary as new words. These new words can further improve the accuracy of word segmentation. There is also a joint model for Chinese word segmentation and new word detection [5]. They added high dimensional new features to the model and used online gradient descent for the improvement of speed.

2.2 Use Linguistic Rules And Related Knowledge

The formation of new words is rule-based, and experts can match the new words in the text by summarizing the template of some new words. The typical method [6] proposed a chunking model applied to detect unknown words by chunking one or more word atoms together according to the word-formation patterns of the word atoms. In addition, there is a model [1] that summarizes the lexical templates of these seed words by putting some seed words into the dictionary, and then iterates the process by matching the text to extract the new words into the dictionary. However, these methods need a lot of time and energy poured, which may bring difficulties to the task.

2.3 Take Users Behavior Data into Account

The third direction mainly takes users behavior data into account to detect new words. For example, The method [7] tries to discover the potential experts in domain-specific fields who use the terminologies frequently, and then further extract domain new words from the terminology of these experts.

2.4 Use Statistical Features

The fourth direction uses statistical features to find words that often come together and extract. There was a study [8] presented an unsupervised method to identify internet new words using improved Point-wise Mutual Information (PMI) combined with some basic rules. A recent study used enhanced mutual information (EMI) algorithm [9] to filter candidate new words in tourism field. Compared with the PMI algorithm, which can only consider the limitations of 2-gram, EMI can consider the n-gram word relationship so as to extract more accurate new words. Beyond that, there was also a method [8] using symmetrical conditional probability (SCP) and branching entropy (BE) as the statistical magnitude to extract the new words. One method of domain-specific new words extraction was also proposed [10], it used the Domain Word Dictionary Model (D-WDM) instead of the traditional WDM model, and it used a domain score function to distinguish whether a word from common word dictionary or from the domain-specific dictionary.

3. Method

In this section, we introduce our model generally in Sect. 3.1, then show the features of the word in Sect. 3.2, present the input to the model in Sect. 3.3, finally explain the construction of the model in Sect. 3.4.

3.1 The Model

We choose the state-of-the-art BiLSTM+CRF framework [11] as our model to do our task without using the CNN layer. As shown in Fig. 1, First, the words’ information combines with the features of them, and then input to the BiLSTM network together. Finally, the information in the network passes the CRF layer and gets the label.

Since the advantage of the LSTM structure, the model can extract features through the surrounding words of each word. Plus the features added to it, the model will get better performance. Comparing with the traditional methods which only use the features extracted by human labor to find new words, our method can utilize the features extracted by human labor and that extracted model at the same time, which can lead to a better result.

3.2 The Feature of Word

Each word contains a lot of information such as part of speech or word length, we call them features. These features are summarized by relevant experts and used to the new word detection. After added the features in our model, we can get better performance. In this subsection, we will introduce these features.

3.2.1 Word Length

Word length represents the length of every word segmented by the tool. When these words form a new word, the new word’s length will be the sum of these words’ length. When the sum of the length is too long, some phrases may be mistaken for new words, which we need to filter out. Therefore,
3.2.2 Part of Speech

The part of speech (POS) is also an important factor in a word. Most multiple character words in Chinese have word-internal syntactic structures, which is roughly the POS sequence of the component characters. Therefore, for every word constructing the new word, their POS’s sequence would have some patterns.

3.2.3 Contextual Entropy

The contextual entropy stand for the freedom of words, which means a word usually whether it appears alone or with some other words. As shown in Eq. (1) and (2):

\[
LCE(w) = \frac{1}{n} \sum_{i=1}^{n} C(a_i, w) \ln \left( \frac{C(a_i, w)}{n} \right) \quad (1)
\]

\[
RCE(w) = \frac{1}{n} \sum_{i=1}^{n} C(w, b_i) \ln \left( \frac{C(w, b_i)}{n} \right) \quad (2)
\]

Where \(LCE(w)\) represents the left contextual entropy; \(RCE(w)\) represents the right contextual entropy; \(w\) stands for the current word; \(n\) represents the number of times of current word appearance; The \(a\) and \(b\) stand for the set of the left adjacent word of \(w\) and the set of right adjacent word of \(w\) separately. \(C(a_i, w)\) means the number of times word \(a_i\) and \(w\) appearance together; \(C(w, b_i)\) means the number of times word \(w\) and \(b_i\) appearance together.

So the smaller of the contextual entropy is, the more fixed the current word appears and it more likely forms a new word with the words in set \(a\) or \(b\).

3.2.4 Degree of Word Coagulation

The degree of word coagulation stands for whether two words usually appear combined together, the combined words maybe form a new word. As shown in Eq. (3):

\[
N(s_1, s_2) = \frac{C(s_1, s_2)}{C(s_1) + C(s_2)} * 2 \quad (3)
\]

Where \(s_1\) and \(s_2\) represent the two words separately; \(C(s_1)\) and \(C(s_2)\) stand for the number of times of the two words appear in the text alone separately; \(C(s_1, s_2)\) means the number of times of the two words appear and combined.

So the bigger of the \(N(s_1, s_2)\), the more likely the word \(s_1\) and \(s_2\) form a new word.

3.3 The Input to the Model

We give an input sentence \(s = w_1, w_2, \ldots, w_n\), in which \(w_i\) denotes the \(i\)th words and \(n\) denotes the position of words in the sentence. These words are cut by the word segmentation tool and each one of them will be assigned with a label \(l_i (l_i \in \{Y, N\})\) which represents if \(w_i\) can joint with the \(w_{i+1}\) and they can form a new word.

We combine the BiLSTM+CRF framework with the new word detection together, as shown in Fig. 1, which is the model structure and the input: “听完我也蓝瘦香菇(难受想哭)！(I also feel uncomfortable and want to cry after listening to that)”.

In the following contents, we will introduce each layer of the model separately.

3.4 The Construction of Model

3.4.1 Embedding Layer

As shown in Fig. 1, every word from the input sentence
matches a word vector. Through the embedding layer, every word will be converted to the corresponding word vector using the embedding lookup method as shown in Eq. (4):

\[ x_{b,e}^w = e^w (c_b, c_{b+1}, \ldots, c_e) \]  

\[ x_{b,e}^p = e^p (c_b, c_{b+1}, \ldots, c_e) \]  

\( x_{b,e}^w \) denotes the vector transformed by the word through the word embedding, \( e^w \) denotes the embedding lookup function. The embedding look up function means getting the corresponding vector of word through the index of it in the embedding table.

Besides the word itself, other features of the word such as POS are added in the model. So we add the feature embedding layer after the word embedding layer. For example, the POS embedding layer presented in Eq. (5).

\[ x_{b,e}^p = e^p (c_b, c_{b+1}, \ldots, c_e) \]  

Where \( e^p \) denotes the POS embedding lookup table.

After getting these vectors from different features of the word and the word itself, we connect them and form a longer vector to put it into the model.

3.4.2 LSTM

LSTM is a type of recurrent neural network (RNN). Compare with the traditional RNN, it solves the problem of the gradient vanishing/exploding by making the information pass between cells [13]. In the same way, the information of words should be kept all the time in the network to extract new words, so it’s necessary to use the LSTM structure in our research.

It uses three gates: the input gate determines which part of the information we need to keep, and the information only comes from the current cell. The forget gate determines which part of the information needs to keep as well, but the difference is that the information comes from all the nodes before the current one. The output gate determines which part of the information needs to transfer to the next node, and the information comes from the combination of the input gate’s output and the forget gate’ output.

3.4.3 BiLSTM

Although the construction of LSTM can relieve the defect of the traditional RNN model, it still has its inadequacy. Every LSTM node can only take the information from the previous node, but it can’t get any message from any latter node. To solve this problem, the bi-directional LSTM (BiLSTM) was represented. It added a backward process after the normally forward propagation of the LSTM nodes. In that case, every LSTM node will also get the message form the latter nodes after getting the message from the previous nodes. In that way, the node can get a more comprehensive message and make a better output. For the new word detection, the new word can be found more precisely.

3.4.4 CRF Layer

For sequence labeling tasks, it’s useful to consider the relationship of adjacent words because in that way we can select the best chain of labels out. For example, the ‘Y’ label usually won’t appear continuously more than 5 times in case the phrase is treated as new words.

We use \( x = \{x_1, \ldots, x_n\} \) here to represent the input sequence of words and let \( y = \{y_1, \ldots, y_n\} \) to represent the labels of corresponding words like ‘Y’ or ‘N’. The \( Y(x) \) denotes all possible label sequences of \( x \) set and \( p(y|x; W, b) \) represents the probability of every label sequence given \( x \) as shown in Eq. (6):

\[ p(y|x; W, b) = \frac{\prod_{i=1}^{n} \psi_i (y_{i-1}, y_i, x)}{\sum_{y' \in Y(X)} \prod_{i=1}^{n} \psi_i (y'_{i-1}, y'_i, x)} \]  

In the form, the \( W \) and \( b \) stand for the weight transfer matrix and bias transfer matrix separately. Each parameter of the two matrixes means the transfer score of two status like label ‘Y’ to label ‘N’ or label ‘Y’. Besides that, there is a potential function \( \psi_i \) of \( y_1, y_2, x \) = \exp \( (W_{y_1,y_2}^T x_i + b_{y_1,y_2}) \). The \( W_{y_1,y_2}^T \) and \( b_{y_1,y_2} \) are come from the \( W \) and \( b \) respectively.

With the conditional probability \( p(y|x; W, b) \), it’s easy to get our loss function \( L(W, b) \) for training the CRF model as shown in Eq. (7):

\[ L(W, b) = -\sum_{i} \log p(y|x; W, b) \]  

The logarithmic maximum conditional likelihood estimation was used here. What we need to do is find out a proper conditional probability and make the loss function reach the maximum. The conditional probability we found is the indispensable premise for the final result. With the conditional probability, the final labeling sequence will be determined and the new words will be extracted.

In general, we use the Viterbi algorithm [14] to train the CRF model and make the prediction. Therefore, we also handle that in the same way.

The setting of our model’s Hyper-parameters are shown in Table 2.

4. Experience

In this section, the following experiments will be conducted: first, we will compare our model with some benchmark

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Hyper-parameters for the model</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>state size of every LSTM layer</td>
<td>100</td>
</tr>
<tr>
<td>LSTM</td>
<td>number of LSTM layers</td>
<td>4</td>
</tr>
<tr>
<td>Dropout</td>
<td>dropout rate</td>
<td>0.5</td>
</tr>
<tr>
<td>Dropout</td>
<td>batch size</td>
<td>10</td>
</tr>
<tr>
<td>Dropout</td>
<td>initial learning rate</td>
<td>0.01</td>
</tr>
<tr>
<td>Dropout</td>
<td>decay rate</td>
<td>0.05</td>
</tr>
<tr>
<td>Dropout</td>
<td>gradient clipping</td>
<td>5.0</td>
</tr>
</tbody>
</table>
models. Second, some features added our model will improve the performance. Third, the different length of the feature’s vector will be presented. Finally, we will compare the model having a different Number of Sentences Input one time (NSI).

4.1 Data Preparation

The domain of new word detection has no public open data sets. So the researchers often create data sets by crawling some websites or using some universal data source themselves. We use SogouT, an open data source of the internet from the Sogou lab as our data sets. We extract train set (about 101,136 lines), validation set (about 9,985 lines), and test set (about 9,357 lines) from it.

We make word segments of these data with ICTCLAS, a word segmentation tool created by the Institute of Computing Technology, Chinese Academy of Sciences. For the golden result of new words, we’re going in two steps. First, the tool to find the candidates for new words was used. Second, we filter out some candidates that are not new words.

In the first step, we use the tool employed by [10] to find the candidates of new words, that tool is base on the frequency of characters. It produce a total of 3,838 new word candidates.

However, some of these candidates usually are not new words. It’s just the word with a high frequency to appear (like “足球” which means ”football”, or “喜欢” which means ”like”), or even some are not a word at all (like “粉丝最” which means ”fans the most” or “我们摔坏” which means ”we break”), they are just the incomplete segments and can’t convey a complete information.

In the second step, we invite two annotators taking two days to filter the data. If a disagreement exists, the two annotators should decide after discussion. After filtering, the 3,838 candidate words reduced by nearly 50 percent, the 1,547 remained candidates formed the final answers of new words. The final golden result of new words having 1,083 words in the train set, 309 words in the validation set and 155 words in the test set.

The new words is very rare in a text. As show in Fig. 2, in our test text, the number of the new words is just take 0.44 percent while the old words takes 99.56 percent. The great contrast is because the words we usually used are common words while the new words is used very rare, It will directly lead the phenomenon that the precision of the model is very low.

4.2 Evaluation Metrics

The precision, recall and F1 score are used, the standard approach to evaluate our model’s performance, as shown in Eq. (8):

\[
\begin{align*}
    p &= \frac{TP}{TP + FP} \\
    r &= \frac{TP}{TP + FN} \\
    F1 &= \frac{2 \times p \times r}{p + r}
\end{align*}
\]

Where TP is the number of the new words predicted correctly; FP is the number of words incorrectly predicted as new words; FN stands for the number of the new words remained undetected.

4.3 Comparing with the Benchmark Models

The benchmark models include the Conditional Random Field model (CRF model) and the tool of new word detection created by the Chinese Academy of Sciences (CAS model). As shown in Table 3, it can be seen that the model without features added has a close performance compared to the benchmark models. While the model with the features added can get a great improvement.

The state of the art model of new word detection: the Domain Word Dictionary Model (D-WDM) added the dictionary of specific domain. The model can predict the new words in the specific domain like MOOC or blog and it has the greatest performance. Every new word which the model predict followed a weight, the larger of the weight is, the new word predicted is more reliable. When we select the top 100 new words predicted by the model and sorted by weights, its MAP score can reach to 79.5 percent.

However, this model has its limitation. It can only predict the new words of specific domain and it need a large dictionary of the domain which need pour the energy and time. On the other hand, our method don’t need any dictionary and the features can be automatically calculate by our algorithm. What is more, our method can predict the new words of all field in the internet because the rules we summarized suit common to all new words.

4.4 The Performance of the Model Having Features

There are four features added to our model including the

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS</td>
<td>.1504</td>
<td>.4669</td>
<td>.2274</td>
</tr>
<tr>
<td>CRF</td>
<td>.1506</td>
<td>.3408</td>
<td>.2087</td>
</tr>
<tr>
<td>BiLSTM+CRF</td>
<td>.1859</td>
<td>.3098</td>
<td>.2324</td>
</tr>
<tr>
<td>BiLSTM+CRF+Features</td>
<td>.3519</td>
<td>.5236</td>
<td>.4209</td>
</tr>
</tbody>
</table>
LE  RE

I am very sad and sorrow

Table 4  Compare of model having different features

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base-Model</td>
<td>.1859</td>
<td>.3098</td>
<td>.2324</td>
</tr>
<tr>
<td>POS</td>
<td>.2552</td>
<td>.1959</td>
<td>.2217</td>
</tr>
<tr>
<td>LEN</td>
<td>.2443</td>
<td>.33</td>
<td>.2808</td>
</tr>
<tr>
<td>RE</td>
<td>.3336</td>
<td>.2965</td>
<td>.314</td>
</tr>
<tr>
<td>DWC</td>
<td>.4487</td>
<td>.2152</td>
<td>.3109</td>
</tr>
</tbody>
</table>

LENth of the word (LEN), the Part Of Speech (POS), the Right contextual Entropy (RE) and the Degree of Word Coagulation (DWC). The Right contextual entropy is only chosen but without the left contextual entropy.

As shown in Fig. 3, the word "blue" has two adjacent words: "very" and "thin". The current word and the left word can be used to calculate the Left contextual Entropy (LE), and the Right contextual Entropy (RE) can be calculated by the current word and the right word. Because we only concern which word can combine the next word and just take the behind words into consideration, so the LE is abandoned, the RE is remained.

4.4.1 The Analysis of the Model with Single Feature

Table 5  Compare of model having different features

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base-Model</td>
<td>.1859</td>
<td>.3098</td>
<td>.2324</td>
</tr>
<tr>
<td>POS+LEN</td>
<td>.2647</td>
<td>.327</td>
<td>.2926</td>
</tr>
<tr>
<td>POS+RE</td>
<td>.3471</td>
<td>.3359</td>
<td>.3414</td>
</tr>
<tr>
<td>POS+DWC</td>
<td>.3603</td>
<td>.36</td>
<td>.302</td>
</tr>
<tr>
<td>LEN+RE</td>
<td>.2716</td>
<td>.33</td>
<td>.298</td>
</tr>
<tr>
<td>LEN+DWC</td>
<td>.2685</td>
<td>.3375</td>
<td>.2991</td>
</tr>
<tr>
<td>RE+DWC</td>
<td>.3046</td>
<td>.5126</td>
<td>.3821</td>
</tr>
</tbody>
</table>

predicted, these words which constructing the wrongly predicted new word don’t often appear together, so the precision can get improve. However, for some real new words, their containing words may have a high freedom, these new words shouldn’t be filterd but the model with RE could filter them, that’s why the recall rate decreased.

When the ‘DWC’ added to the model, the precision get a great improve but the recall rate is also decreased. Like the RE, the DWC can help the model filter some new words wrongly predicted, the only difference is the DWC can find out which words often fixed together while the RE is in the opposite way, find out which words often not fixed together. Comparing with the RE, the DWC take a more directly way to assist the model, so the improve it bring is greater. For the recall rate, it also filter some correct new words, that’s why the recall rate decreased.

4.4.2 The Analysis of the Model with Two Features

Every feature can assist the model get the different degrees of improvement, and research their combination’s influence to the model is also meaningful. As shown in Table 5, there are 6 type of combination from these features.

When the ‘POS’ and ‘DWC’ added to the model, the model’s precision can get the greatest improve compare with other feature’s combination. However, the precision is still lower than that in the model with just DWC. That’s because different features maybe conflict each other. To the precision of model, the POS bring a bad effect to the DWC. However, different features can also help each other in some ways. To the recall rate of model, it’s greater than that in the model with just DWC and the model with just POS.

When the ‘DWC’ and ‘RE’ added to the model, the model’s recall rate can get the greatest improve compare with other feature’s combination. That’s because the DWC and RE are the same type feature, they can assist the model predict more correct new words. However, To the precision of model, it’s lower than that in the model with just DWC and the model with just RE. That’s prove that the RE and DWC may conflict each other in the effect of precision.

Different features’ combination may help each other in some ways but also could conflict each other in some ways. So it’s necessary to find out a combination which can help the model get the greatest improve.
Table 6  Compare of model having different features

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base-Model</td>
<td>.1859</td>
<td>.3098</td>
<td>.2324</td>
</tr>
<tr>
<td>POS+LEN+RE</td>
<td>.2426</td>
<td>.3388</td>
<td>.2825</td>
</tr>
<tr>
<td>POS+LEN+DWC</td>
<td>.2775</td>
<td>.3347</td>
<td>.3034</td>
</tr>
<tr>
<td>POS+RE+DWC</td>
<td>.3192</td>
<td>.5194</td>
<td>.3954</td>
</tr>
<tr>
<td>LEN+RE+DWC</td>
<td>.2763</td>
<td>.3317</td>
<td>.3015</td>
</tr>
<tr>
<td>POS+LEN+RE+DWC</td>
<td>.2756</td>
<td>.3347</td>
<td>.3023</td>
</tr>
</tbody>
</table>

4.4.3 The Analysis of the Model with More than Two Features

We make more combination of features added the model and observe the effect. As shown in Table 6, there are five types of feature combination added the model, every model has three or more than three features added.

The combination of DWC, POS, and RE can help the model get the best result in which the F1 score reaches 39.54 percent. What’s more, it also help the model reach the highest recall rate among all the model we have set, it reaches 51.94 percent. That’s because these three features are merged well and can help each other. Among them, DWC offers a greatest contribution.

There is a rule in the experiment. As shown in Fig. 4: the combination of RE and DWC can improve the precision of the model and the combination of DWC and POS can improve the recall of the model. Moreover, the combination of these three features can improve the precision, recall and F1 score of the model at the same time.

When all the four features added the model, the model’s scores is also lower than the three features above. So it doesn’t mean that the more features we add to the model, the better performance our model can achieve. This phenomenon may be caused by the conflict of these features, or some useful information is faded when passing in the network because be disturbed by the useless information from some features.

At last, the three features are selected and used for further experiment.

4.5 The Effect of Vector Length

After find out the best combination of features: POS, RE and DWC, the vector length of these features is also a factor effecting the result from our model. As shown in Fig. 5: the vector length of POS is equals to 50, while that of RE is equals to 100. So when they combine together and form a new vector, the longer the vector length of feature is, the larger weight it will take in the new vector. Each vector of feature is correspond to a length, our aim is to find out the best vector length of every feature which make our model perform better.

We chose five different vector lengths for every feature: 10, 30, 50, 70, 100.

As shown in Fig. 6, we fix the vector length of POS as 100, and then observe the change about F1 score of our model by changing the vector length of DWC and RE. The horizontal axis represents the vector length of RE, different color of column represents different vector length of DWC, the vertical axis represents the F1 score.

It can be seen that, when the vector length of DWC and RE is short, the F1 score of the model is relatively high, which the length is 10 or 30, the F1 score can above 0.4. We will conduct further experiments based on this result.

We try four different combinations of DWC and RE in vector length of 10 or 30. As shown in Fig. 7, the horizontal axis represents the vector length of POS, the vertical axis represents the F1 score of the model, and different broken lines represent different combinations of vector lengths in the features.

It can be found that when the number of vector length of DWC and RE is 10, and the vector length of POS is 70, the model has the best effect, the precision reach to 35.19 percent, the recall reach to 52.36 percent and the F1 score reach to 42.09 percent.

The RE and DWC are calculated by the current word and its neighbor, so they are confused features and need to
take less weight in the whole vector. While the POS is the feature from the current word itself and doesn’t be disturbed by other words, so it takes a larger weight. All of the vector lengths of these features can’t longer than the vector lengths of the word itself because they're just auxiliary in the predicting process.

The final vector length of these features and of the word itself is shown in Table 7.

### 4.6 The Effect of NSI

The Number of Sentences Input to model one time (NSI) for the prediction is also the factor influence the performance of the model. If the NSI is too small, the model can’t extract enough information from words to make a prediction, while if the NSI is too large, the speed of the model’s prediction will be too slow, and the computer may out of memory. There is a threshold of NSI exists, and it makes the model predict fast and accurately.

We set 13 NSI which is 10 to 130, the gap is 10.

As shown in Fig. 8, the horizontal axis represents the NSI, and the vertical axis represents the scores of the model, and there are three scores in the graph. The red one represents the precision, the blue one represents the F1 score and the black one represents the recall. We can find that with the increase of the NSI, our model’s recall get continuous improvement. But when the NSI is more than 40, the speed of growth is beginning to slow, and when the NSI reaches 100, the score is basically steady. On the other hand, the precision has little change. As to the F1 score, we can see the slow increase of it and get steady after the NSI reaches 100. So it can be seen that the recall score has a more contribution to the increase of F1 score. When NSI is 100, the model has the best effect, the precision reach to 33.82 percent, the recall reach to 73.1 percent and the F1 score reach to 46.24 percent.

### 5. Conclusion

This paper uses the BiLSTM+CRF model to solve the old task: new word detection. Compare to the traditional method, our model have a better performance. What’s more, we added many different features into the model and tried the best combination and the best dimension of them. In that way, we can choose the best model out. We also tried once to input a different number of sentences into our model to find the best of it.

### Acknowledgements

This work was supported by the National Natural Science Foundation of China (61972003,61672040). We would also like to thank the anonymous reviewers for their helpful comments. We would like to thank the referees for their comments, which helped improve this paper considerably.

### References


Jianyong Duan is a professor, born in 1978.10. He graduated from Department of computer science, Shanghai Jiao Tong University by 2007.12. His major research field including natural language processing and information retrieval.

Zheng Tan is a master in College of Informatics, North China University of Technology. His major research field is New word detection.

Mei Zhang is an associate professor, born in 1980.11. Her major research field focus on knowledge map.

Hao Wang received the Ph.D. degree in Computer Application Technology from Tsinghua University in 2013. He is now an assistant professor in College of Informatics, North China University of Technology. He has published more than 20 papers in international conferences and journals. His research interests include machine learning and data analysis.