Segmenting Chinese Microtext: Joint Informal-Word Detection and Segmentation with Neural Networks

Meishan Zhang, Guohong Fu*, Nan Yu
School of Computer Science and Technology, Heilongjiang University, China
mason.zms@gmail.com, ghfu@hotmail.com

Abstract

State-of-the-art Chinese word segmentation systems typically exploit supervised models trained on a standard manually-annotated corpus, achieving performances over 95% on a similar standard testing corpus. However, the performances may drop significantly when the same models are applied onto Chinese microtext. One major challenge is the issue of informal words in the microtext. Previous studies show that informal word detection can be helpful for microtext processing. In this work, we investigate it under the neural setting, by proposing a joint segmentation model that integrates the detection of informal words simultaneously. In addition, we generate training corpus for the joint model by using existing corpus automatically. Experimental results show that the proposed model is highly effective for segmentation of Chinese microtext.

1 Introduction

Word segmentation has been a fundamental task for Chinese language processing, as the Chinese language does not have explicit boundaries between words in a sentence. Mainstream methods for the task can be divided into two categories: the character-based models [Xue, 2003; Tseng *et al.*, 2005], which cast the task as a sequence labeling problem by assigning each sentential character with word boundary labels such as {**B**eginning, **M**iddle, **E**nding, **S**ingle-character word}, and the word-based models [Andrew, 2006; Zhang and Clark, 2007], which directly produce the resulting words incrementally in a left-to-right manner.

Early work exploits statistical models for both categories by using discrete features, such as character ngrams and words [Tseng *et al.*, 2005; Zhao *et al.*, 2006; Sun and Xu, 2011]. Recently, neural models have gained increasing interests, because they have achieved state-of-the-art performances in a number of natural language processing tasks [Collobert *et al.*, 2011]. Representative work includes character-based neural models of [Zheng *et al.*, 2013] and [Chen *et al.*, 2015], as well as word-based neural models for example [Cai and Zhao, 2016], [Liu *et al.*, 2016] and [Zhang *et al.*, 2016]. Overall, word-based neural models archive the top performing results, because they are capable of utilizing both character-level and word-level features.

The state-of-the-art segmentation models can report an F-score over 95% on a standard Chinese Treebank (CTB) test corpus, with supervised learning on a standard CTB training corpus as well [Zhang *et al.*, 2016]. However, these models can be ineffective for processing of Chinese microtext, such as SMS, weibo and weixin chat corpus, which has drawn much attention in the NLP community. The same model can have sharp decreases over 10% on a weibo dataset, resulting in an F-score only close to 83%. One important reason accounting for it is the issue of informal words. For example, "**麻麻**(mother) 吃饭(eat) 去(go to) 了(ready)", where "**麻麻**" should be "**妈妈**" by the standard form.

There have been several work on addressing the problem of informal words [Li and Yarowsky, 2008; Wang *et al.*, 2013]. In particular, [Wang and Kan, 2013] proposes a supervised learning method to jointly performing word segmentation and informal word recognition, achieving improved performances for word segmentation. They assume that informal words can be learned through well-annotated training corpus, which can be reasonable for the informal words limited to a certain style but may be problematic when the domain or the standard changes. For example, we can hardly image that "**手机**(phone)" could be a standard form twenty years ago, while it has been widely accepted as a formal word in recent years.

To solve the problem, [Qian *et al.*, 2015] offers an alternative method to build a training corpus with an external dictionary of formal/informal word pairs, which can be mined from the web and requires only minimal supervision. They propose a joint model for word segmentation, normalization and POS tagging, where informal words are detected and normalized concurrently. Their results rely heavily on the quality of the constructed dictionary. When an informal word falls out of the dictionary, it cannot be detected by using their model.

We combine the advantages of both work, aiming to enhance the segmentation of Chinese microtext. On the one hand, we construct training examples automatically with the help of an external dictionary. On the other hand, we leave the normalization as one future step, making our model being able to detect informal words even with a low-quality dictionary. In particular, our major goal is to improve the segmentation performance. We concern little on the difference of stan-

*Corresponding author.

dards between formal and informal words, especially when an irregular word does not influence the final segmentation results. For example, the word "代驾(drive replacement)" in the sentence "你(you) 需要(need) **代驾** 吗(do) ？ ", it can be treated by either formal or informal.

Concretely, we follow the recent line of work by using neural features, adapting a word-based neural model for joint word segmentation and informal word detection. We take the transition-based neural model of [Zhang *et al.*, 2016] as the baseline, extending it being able of detecting informal words simultaneously. We design novel features from long short term memory (LSTM) networks for the joint model. Experimental results show that the joint model can bring significant improvements on a standard weibo test dataset, resulting in improvements by 3.6% for the segmentation F-score. Our code is publicly available under GPL at https://github.com/zhangmeishan/NNTransitionNormalization.

2 Baseline

We take word-based transition model of [Zhang *et al.*, 2016] as our baseline, which has reported state-of-the-art performances for Chinese word segmentation and can be easily adapted to the joint task.

2.1 Model

The baseline model treats Chinese word segmentation as an action-based state-transition problem, performing the task by a sequence of actions step-by-step incrementally. The key idea lies in the transition system, which consists of two components, termed by state and action, respectively. A state includes two parts, one being a stack that stores a sequence of partially-segmented words ($w_1 \cdots w_{m-1} w_m$) and the other being a queue that stores a sequence of unprocessed characters ($c_i c_{i+1} \cdots c_n$), as shown in Figure 1(a), which represents a partial result in spirit.

An action specifies how a state is transformed into another state, making a state advance by one step. There are two different actions in the baseline model:

- `Append` (APP), which shifts the first character c_i of the queue onto the stack, appending it to the front word w_m of the stack;
- `Separate` (SEP), which moves the first character c_i of the queue onto the stack as a new (sub) word.

Based on the definition, word segmentation is performed as follows. Initially, we have a start state with an empty stack and a queue of all sentential characters. At each step, we choose one action based on the resulting state and apply it, reaching a next-step state with one more character being processed. When all characters are processed, we reach an end state with an empty queue, and the word sequence in the stack is the final segmentation result. For example, the action sequence to reach "麻麻(mother) 吃饭(eat) 去(go to) 了(ready)" is "SEP APP SEP APP SEP SEP".

To facilitate word segmentation, we define a score function over states, and our goal is to find the highest-scored ending state during decoding. Assuming that one state s_i is reached

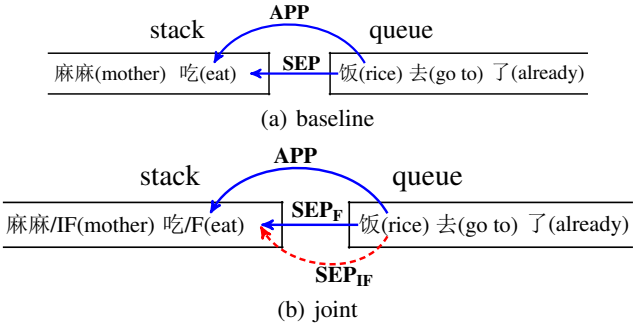

Figure 1: Examples to illustrate the transition systems of the baseline and joint models.

Algorithm 1 Beam-search for the baseline and joint models, where the in-box part is additional for the joint model only.

```
agenda ← { (empty state, score=0.0) }
for i in 1 ··· max-step
    next_states ← { }
    for state in agenda
        new ← APPLY(state, APP)
        ADDITEM(next_states, new)
        new ← APPLY(state, SEP/SEP_F)
        ADDITEM(next_states, new)
        new ← APPLY(state, SEP_IF)
        ADDITEM(next_states, new)
    agenda ← TOP-B(next_states, B)
best ← BESTITEM(agenda)
```

by an action sequence $a_1 a_2 \cdots a_i$, the score of s_i is defined by the following formula:

$$\text{score}(s_i) = \sum_{j=1}^{i} W_{a_j} \mathbf{h}_{s_{j-1}} = \text{score}(s_{i-1}) + W_{a_i} \mathbf{h}_{s_{i-1}}, \quad (1)$$

where W is a model parameter, s_0 is the start state whose score is zero, $s_1 s_2 \cdots s_{i-1}$ denotes the historical states, and $\mathbf{h}_{s_{i-1}}$ denotes the feature representation of s_{i-1}.

Given an input sentence, we search for the highest-scored ending state incrementally. At each step, we have two choices for each state, thus the number of states increases exponentially by the searching step, which makes exact decoding impractically. To solve the problem, we follow [Zhang *et al.*, 2016], exploiting standard beam-searching algorithm to find the optimum result, shown by Algorithm 1.

2.2 State Representation

We exploit different feature representations for different next-step actions, following [Zhang *et al.*, 2016]. The representations are derived from two sources of basic features, including the input character sequence $c_1 c_2 \cdots c_n$, and the partial word sequence $w_1 \cdots w_{m-1} w_m$ from a given state. Figure 2 shows the feature representation methods.

First, we make use of sentential characters by the character unigrams, bigrams as well as their types.[1] At each position i, we make embeddings for c_i, $c_{i\pm1} c_i$ and $type(c_i)$ by

[1]We define four different character types, namely Chinese char-

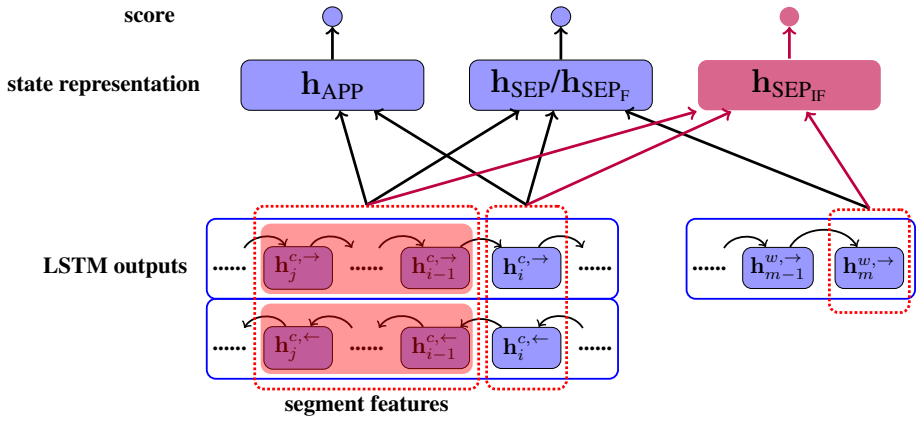

Figure 2: Neural feature representations of the baseline and joint models, where the solid lines denote the neural networks of the baseline model, and the dashed lines denote the additional parts for the joint model.

using three look-up matrixes E^c, E^{bc} and E^{ct}, respectively, resulting in $\mathbf{x}_i^c = \mathbf{e}_i^c \oplus \mathbf{e}_i^{bc} \oplus \mathbf{e}_i^{ct}$.[2] For each word w_i, we obtain its embedding $\mathbf{e}_i^w$ by looking up from E^w similar to the character-level embeddings.

Then we build three LSTM neural networks [Graves and Schmidhuber, 2005] over the vector representations of characters and words, namely the left-to-right character LSTM, the right-to-left character LSTM and the left-to-right word LSTM, receiving three output vector sequences, $\mathbf{h}_1^{c,\rightarrow}\mathbf{h}_2^{c,\rightarrow}\cdots\mathbf{h}_n^{c,\rightarrow}$, $\mathbf{h}_1^{c,\leftarrow}\mathbf{h}_2^{c,\leftarrow}\cdots\mathbf{h}_n^{c,\leftarrow}$ and $\mathbf{h}_1^{w,\rightarrow}\mathbf{h}_2^{w,\rightarrow}\cdots\mathbf{h}_m^{w,\rightarrow}$, respectively.

Further, we use the three output hidden vectors from LSTMs to extract features for state representation. We extract different features according to the next action. Given a state s_i, assuming the next incoming character is c_i and the last word in the stack is w_m, we use four different features for the action APP, which are all obtained from the two character LSTMs: (1) $\mathbf{h}_i^{c,\rightarrow}$, (2) $\mathbf{h}_i^{c,\leftarrow}$, (3) $\mathbf{h}_{i-1}^{c,\rightarrow} - \mathbf{h}_{j-1}^{c,\rightarrow}$ and (4) $\mathbf{h}_j^{c,\leftarrow} - \mathbf{h}_i^{c,\leftarrow}$, where j is the start position of w_m. The latter two features denote the segment features of the character sequence in w_m ($c_j \cdots c_{i-1}$), which are suggested by [Wang and Chang, 2016]. For the action SEP, we exploit the above four features as well. Besides, we use $\mathbf{h}_m^{w,\rightarrow}$ from the word LSTM to incorporate word-level features.

Finally, we concatenate all extracted features, and feed the result vector into a non-linear neural layer, resulting in the feature representation of a state. The resulting feature vector is totally different for different actions of the next step. Thus Equation 1 should be more accurately rewritten as:

$$\text{score}(s_i) = \text{score}(s_{i-1}) + W_{a_i}\mathbf{h}_{s_{i-1}}^{a_i}, \qquad (2)$$

where $\mathbf{h}_{s_{i-1}}^{a_i}$ is the representation of s_{i-1} by the action a_i.

3 Joint

In this section, we introduce the joint model of word segmentation and informal word recognition. First, we describe the background of the informal word detection, illustrating representative sources of informal words by several examples. Then we present the proposed joint model, which is extended from the baseline model by making adaptations in the transition system and feature representation, respectively.

3.1 Informal Words

The microtext such as weibo texts, weixin chats as well as SMS texts contains a number of informal words, which have been used seldom in the texts of formal written language. According to [Wang and Kan, 2013], there are three representative sources of informal words:

- The first source of informal words are generated by phonetic substitutions, for example, "**木(mu)**有" corresponding to "**没(mei)**有" (do not have), and "妹**纸(zhi)**" corresponding to "妹**子(zi)**" (sister).
- The second kind of informal words are abbreviation words, which are used for convenience without loss in understandability, for example, "高铁(high-speed railways)" corresponding to "**高**速(high-speed) **铁**路(railways)".
- The third kind of words are Chinese neologisms, which have been never defined in any formal language before, for example, "达人(important person)", "五毛(commentator at web)" and "粉丝(fans)".

The use of informal words can have many reasons, which can be caused by accidental errors or by well-thought ideas in order to attack the interests of readers. We do not concern the standard definition of the informal words since our goal is to study their influence for word segmentation.

[Wang and Kan, 2013] demonstrates that recognition of informal words can be helpful for word segmentation on the microtext. We follow their work to enhance the baseline model by jointly detecting informal words.

3.2 The Transition System

Shown by Figure 1(b), we extend the baseline transition system by attaching additional word category information, making it be able of handling informal word detection as well.

acter, letter, digit and other, respectively.

[2] We take $c_{i-1}c_i$ as inputs for left-to-right LSTM and take $c_{i+1}c_i$ as inputs for right-to-left LSTM.

The queue of a state being $c_i c_{i+1} \cdots c_n$ remains unchanged between the baseline and joint models, while the stack for sole word segmentation being $w_1 \cdots w_{m-1} w_m$ is changed into $w_1/t_1 \cdots w_{m-1}/t_{m-1} w_m/t_m$, where $t_i \in \{\text{F}, \text{IF}\}$, F denotes a formal word and IF denotes an informal word.

In addition, we make changes in transition actions as well, using three actions:

- APP, which is the same as the baseline model;
- SEP_F, which produces a new formal (sub) word by moving the first character c_i of queue onto the stack;
- SEP_IF, which produces a new informal (sub) word by moving the first character c_i of queue onto the stack.

If we assume the newly-generated word being a formal word by default, SEP_F is in spirit the same as the baseline SEP action. Thus the only action difference between the baseline and joint models is the third SEP_IF action.

Using the above transition system, "麻麻/IF(mother) 吃饭/F(eat) 去/F(go to) 了/F(ready)" can be produced by the action sequence "SEP_IF APP SEP_F APP SEP_F SEP_F". The searching algorithm during decoding remains the same as the baseline model, except that there is one additional action for each state, as shown in Algorithm 1 by the in-box part.

3.3 State Representation

We exploit the same three kinds of LSTMs as the baseline model to extract features for further state representation, and follow the baseline model using separate state representations for different actions. Figure 2 shows the state representation method of the joint model.

There are two changes in comparison with the baseline models. First, we add an additional channel for state representation of the next-step SEP_IF action, where the extracted features are the same as the SEP_F/SEP action, but use different neural layers for feature composition. These neural layers are formally the same as the SEP_F/SEP action but using a different set of model parameters.

Second, we make special normalization for informal words, by using a unique symbol to represent them. This change only affects the input vector of the word-level LSTM. When an informal word is detected, we obtain its embedding from the special symbol, rather than its form. For example, the input word-level embeddings for "w_1/IF w_2/F w_3/F w_4/F" are changed from $\mathbf{e}^w_1\mathbf{e}^w_2\mathbf{e}^w_3\mathbf{e}^w_4$ into $\mathbf{e}^{\text{sym}}\mathbf{e}^w_2\mathbf{e}^w_3\mathbf{e}^w_4$, where $\mathbf{e}^w_i$ denotes the embedding of w_i, and $\mathbf{e}^{\text{sym}}$ denotes the embedding of the special symbol, which is a model parameter that can be learned during training.

4 Training

4.1 Training Corpus Generation

To avoid our joint model limited to a certain style, we remove the dependency of the requirement of manually-annotated corpus for training. Instead, we construct an automatic training corpus based on an existing corpus, which has been annotated with word segmentations. We make use of a dictionary with formal/informal word ones, making random substitutions to change formal words into informal words, and obtaining the final training corpus for the joint model.

First, we follow [Qian *et al.*, 2015] to construct the dictionary of informal/formal word pairs, by querying Baidu search engine with manually defined queries. The example queries include "informal也是formal 的 意思"(informal is also means formal), "informal也称formal"(informal can be also said as formal), "informal(formal)" and etc. For more details, one can refer to their paper. In this work, we simply use a subset of the dictionary generated by their work, obtaining a set of 3,000 formal/informal word pairs.[3]

Then we use one already annotated corpus, for example, the training corpus of CTB6.0 in this work, to generate the training corpus for our joint model automatically. We assume that all words of the existing corpus are formal words. Given one sentence $w_1 w_2 \cdots w_n$, we traverse all sentential words by sequence, substituting a word w_i randomly by its informal pair with a certain probability ρ. For example, we can obtain "**麻麻**/IF(mother) 吃饭/F(eat) 去/F(go to) 了/F(ready)" by "**妈妈**(mother) 吃饭(eat) 去(go to) 了(ready)" if we have the pair "**妈妈**(mother)/**麻麻**" in our dictionary.

By the above substitution, we can produce a number of training examples for our joint model. Although the automatic training corpus is pseudo, we can use it to train a model to segment real-world microtexts. We will demonstrate its effectiveness in the experimental part.

4.2 Model Update

We exploit online learning to update model parameters sequentially by each training example, and use a similar global learning strategy to guide the update following [Zhang *et al.*, 2016]. We are different in the objective function, where they exploit a max-margin objective to maximize the score of a gold-standard state, and we exploit a max-entropy objective function instead [Andor *et al.*, 2016], because we find that their learning method is highly sensitive to the initialization of model parameters according to our preliminary experiments. Formally, at step i, assuming the gold-standard state is s^g_i, our goal is to minimize the following objective function:

$$\text{loss}(s^g_i, \Theta) = -\log p_{s^g_i} = -\log \frac{\text{score}(s^g_i)}{\sum_{s'_i} \text{score}(s'_i)}, \quad (3)$$

where $p_{s^g_i}$ is the probability of the gold-standard state by the model, Θ denotes the model parameters, s'_i denotes all the valid states that can be produced at step i.

The number of s'_i increases exponentially by the step number i, since we can expand one state into two (the baseline model) or three (the joint model) states at each step. Thus exact calculation of $p_{s^g_i}$ is impossible because of the denominator $\sum_{s'_i} \text{score}(s'_i)$ in Equation 3. We exploit an approximation to compute the probability instead, following [Andor *et al.*, 2016]. Different from their work that uses only the beam states to evaluate the denominator, we use all the expanded states by the beam states, which can approximate more accurately, since we have twice or thrice more states. As shown in Algorithm 1, we use the set of states in *next_states* for computation, rather than the states in *agenda* that they use.

[3] Their dictionary consists of 32,787 informal/formal word pairs, which is available at https://github.com/qtxcm/JointModelNSP.

We exploit the same early-update strategy as [Zhang *et al.*, 2016] and [Andor *et al.*, 2016], to learn better model parameters, making the training and decoding be consistent thus reducing the search errors.

5 Experiments

5.1 Data and Evaluation

We use CTB6.0 and the released Weibo corpus of [Qian *et al.*, 2015] to evaluate our models. The CTB6.0 is one standard benchmark for Chinese word segmentation, of which the sentences are mainly chosen from well-formatted news corpus. We regard all the words in CTB6.0 being formal words. We follow [Zhang *et al.*, 2016] to split the corpus into training, development and test corpus. The Weibo corpus contains two thousand sentences, which have been annotated with word segmentation together with informal words already.[4] We randomly split the corpus into development and test corpus by a proportion of 4:6.

We evaluate our models mainly by the word segmentation performances, using three metrics: recall (R), precision (P) and their F-measure score (F). To evaluate the performance of informal word recognition, we use the recall as the major metric, which is the ratio of correctly recognized informal words by the gold-standard informal words.

5.2 Hyper-Parameters

We tune all model hyper-parameters according to the development results on the CTB6.0 dataset. Most of the hyper-parameter values are the same as [Zhang *et al.*, 2016]. We make all embedding sizes equal to 50, including the embeddings of words, characters, character bigrams and character types. The dimensions of hidden outputs of all LSTM structures are 100, including the bi-directional character LSTMs and the left-to-right word LSTM. The sizes of vector representations of states for APP, SEP/SEP$_{\text{F}}$ and SEP$_{\text{IF}}$ are 80, 100, and 100, respectively.

5.3 Training Details

We use Adam [Kingma and Ba, 2014] with an initial learning rate 10^{-3} to update model parameters by using back-propagation. In addition, we use gradient clipping by a max norm 16 and l_2-regularization by a parameter 10^{-6}, which have been widely adopted to tune model parameters of neural networks [Orr and Müller, 2003]. We use pre-trained word embeddings to initialize the look-up tables of word embeddings E^w, character embeddings E^c and character bigram embeddings E^{bc}, respectively.[5] We keep E^w fixed without fine-tuning during training following [Zhang *et al.*, 2016].

5.4 Baseline Performances

Our baseline model achieves better results than [Zhang *et al.*, 2016]. Based on the same dataset of CTB6.0, we achieve an F-score of 95.4% on the test corpus with a beam size of 4, slightly higher than their reported number of 95.0% with beam size 16. The reason can be various since the feature representation and training approach are both different. Overall, our baseline achieves strong performances among the neural models of word segmentation.

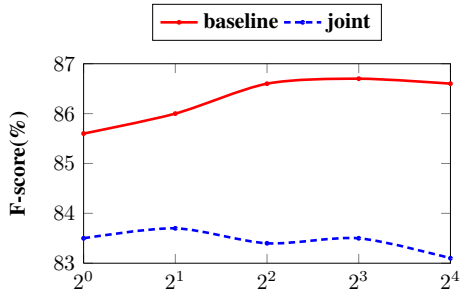

Figure 3: F-scores with respect to beam sizes for the baseline and joint models.

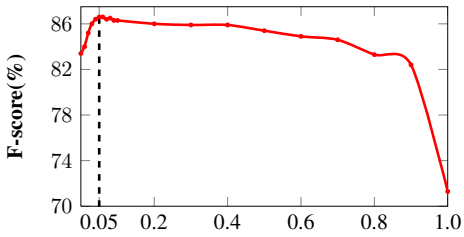

Figure 4: F-scores with respect to substitution probabilities for the joint models.

5.5 Development Results

To better understand our models, we conduct development experiments on the Weibo dataset.

The Influence of Beam Size

We investigate the influence of beam size in our models, to study the performances of the baseline and joint models both. Figure 3 shows the comparison results, where beam sizes are 1, 2, 4, 8 and 16, respectively. As shown, the F-score of the baseline segmentation increases little with larger beam sizes. While the F-score of the joint model increases apparently when the beam size increases from 1 to 4, and a larger beam size over 4 does not bring significant improvements. Thus we use beam size 4 for the later experiments in this work. In addition, the performances of the joint models are consistently better than the baseline models, and the gap between their best-reported performances can reach larger than 3%.

The Influence of Word Substitution Probability

We produce the training corpus of the joint model by randomly substituting formal words in CTB60 training corpus into their informal ones according to a certain dictionary. We use a hyper-parameter ρ to control the probability of the random substitution. Here we study the influence of the probability ρ. Shown in Figure 4, the performance increases as ρ increases from 0 to 0.05, and then it remains stable until it reaches 0.1, after that the performance drops gradually. According to the observation, we choose ρ by 0.05 to build the training corpus for the joint model.

[4] Available at https://github.com/qtxcm/JointModelNSP.

[5] We use the same embeddings shared by the authors of [Zhang *et al.*, 2016].

Models	P	R	F	R_{IF}
baseline	83.6	82.8	83.2	—
joint (this work)	**86.4**	**87.3**	**86.8**	**56.4**
joint (FCRF)	85.5	86.6	86.0	52.5

Table 1: Main results on the Weibo test corpus.

Models	formal	Informal
baseline	83.7	54.1
joint	**88.1**	**66.4**

Table 2: Recalls of the baseline and joint models with respect to formal and informal words.

5.6 Final Results

Table 1 shows the final results of our joint model and the baseline model on the Weibo test corpus. The joint model achieves significantly better results compared with the baseline segmentation model (the p-value is below 10^{-5} by using the pairwise t-test), bringing increases of 3.6% in the F-score values, which demonstrates that the joint informal-word detection and word segmentation model is highly effective for segmentation of Chinese microtext. We report the recall of informal-word recognition as well, which can be potentially useful for further analysis.

We also report the performances by using a factorial CRF (FCRF) model, which is proposed by [Wang and Kan, 2013]. In particular, we use the similar neural features proposed in this work instead for fairer comparisons, rather than the discrete features exploited in their paper, because neural features can give slightly better segmentation performances according to our preliminary experiments. As shown in Table 1, our transition-based joint model achieves slightly better performances than the model of factorial CRF.

5.7 Analysis

We conduct analysis on the test corpus of the Weibo dataset, to study the baseline and joint model in detail. Our main motivation lies in that the overall segmentation performance can be potentially boosted by identifying the informal words correctly. Thus we study the segmentation performances of the formal and informal words separately in the two models, by investigating their recall values. Table 2 shows the comparison results. We find that the recalls of informal words are much worse than that of formal words as a whole. By using the joint model to recognize informal words at the same time, the recall of informal words is greatly increased, and meanwhile brings a positive effect to the formal words.

6 Related Work

Word segmentation is generally accepted as the first step for Chinese language processing, and has been studied intensively in the NLP community [Xue, 2003]. Supervised learning based on a manually-annotated training corpus has dominated the research work of word segmentation [Tseng *et al.*, 2005; Andrew, 2006]. Early work exploits discrete features using statistical models, which are extracted by sophistical human supervision [Sun *et al.*, 2012; Zhao, 2009; Zhang and Clark, 2007]. Recently, neural network models have received great interests. With pre-trained character/word embeddings and highly effective neural structures such as LSTM, these models report promising results [Pei *et al.*, 2014; Chen *et al.*, 2015; Liu *et al.*, 2016; Cai and Zhao, 2016]. We follow the line of work using neural networks to segment Chinese microtext.

The processing of Chinese informal texts including microtext has gained increasing attentions [Wang *et al.*, 2013]. [Li and Yarowsky, 2008] proposes a ranking model among informal/formal word pairs, which are collected from Baidu search engine. The dictionary of informal/formal word pairs is valuable, which is further demonstrated by [Qian *et al.*, 2015]. They propose a joint model for word segmentation, normalization and POS tagging. The above work all involves word normalization, which relies on a high-quality dictionary of informal/formal word pairs. It is quite impractical, and on the other hand, they are unable of recognizing OOV informal words. Thus our work resorts to another line of work by only performing informal word recognition and word segmentation. [Wang and Kan, 2013] presents a joint model for the two tasks based on an annotated training corpus, showing that informal word recognition is helpful for word segmentation, which motivates our work. Our model does not rely on an annotated training corpus. We merely require a dictionary with less than 3000 pairs of informal/formal words, and generate training corpus automatically by the dictionary.

The transition-based framework has been widely adopted for structural NLP tasks [Zhang and Clark, 2011], including syntactic parsing [Zhu *et al.*, 2013], information extraction [Li and Ji, 2014] and the work of joint models [Zhang *et al.*, 2013; Wang and Xue, 2014]. Recently, a number of work has devoted their efforts to study the framework under the neural settings [Zhou *et al.*, 2015; Andor *et al.*, 2016]. In this work, we apply the framework to the joint informal word detection and word segmentation.

7 Conclusion

We proposed a joint model to enhance the segmentation of Chinese microtext, by performing word segmentation and informal word detection simultaneously. We extended a state-of-the-art transition-based neural model, and made slight changes to adapt for the joint task. In addition, we constructed a training corpus for the joint model by using a dictionary of informal/formal word pairs automatically. Experiments show that the proposed joint model can significantly improve the segmentation performance on a weibo dataset, resulting in increases over 3%. The results demonstrate that our joint model is highly effective for Chinese microtext.

Acknowledgments

We thank the anonymous reviewers to improve the final paper. This work is supported by National Natural Science Foundation of China (NSFC) grants 61602160 and 61672211, Natural Science Foundation of Heilongjiang Province (China) grant F2016036.

References

[Andor *et al.*, 2016] Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. Globally normalized transition-based neural networks. In *ACL*, pages 2442–2452, August 2016.

[Andrew, 2006] Galen Andrew. A hybrid markov/semi-markov conditional random field for sequence segmentation. In *EMNLP*, pages 465–472, July 2006.

[Cai and Zhao, 2016] Deng Cai and Hai Zhao. Neural word segmentation learning for chinese. In *ACL*, pages 409–420, 2016.

[Chen *et al.*, 2015] Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. Long short-term memory neural networks for chinese word segmentation. In *EMNLP*, pages 1197–1206, 2015.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537, 2011.

[Graves and Schmidhuber, 2005] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

[Kingma and Ba, 2014] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Li and Ji, 2014] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *ACL*, pages 402–412, 2014.

[Li and Yarowsky, 2008] Zhifei Li and David Yarowsky. Mining and modeling relations between formal and informal Chinese phrases from web corpora. In *Proceedings of EMNLP*, pages 1031–1040, October 2008.

[Liu *et al.*, 2016] Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, and Ting Liu. Exploring segment representations for neural segmentation models. In *Proceedings of IJCAI 2016*, 2016.

[Orr and Müller, 2003] Genevieve B Orr and Klaus-Robert Müller. *Neural networks: tricks of the trade*. Springer, 2003.

[Pei *et al.*, 2014] Wenzhe Pei, Tao Ge, and Baobao Chang. Max-margin tensor neural network for chinese word segmentation. In *ACL*, pages 293–303, 2014.

[Qian *et al.*, 2015] Tao Qian, Yue Zhang, Meishan Zhang, Yafeng Ren, and Donghong Ji. A transition-based model for joint segmentation, pos-tagging and normalization. In *EMNLP*, pages 1837–1846, 2015.

[Sun and Xu, 2011] Weiwei Sun and Jia Xu. Enhancing chinese word segmentation using unlabeled data. In *EMNLP*, pages 970–979, July 2011.

[Sun *et al.*, 2012] Xu Sun, Houfeng Wang, and Wenjie Li. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of the 50th ACL*, pages 253–262, July 2012.

[Tseng *et al.*, 2005] Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. A conditional random field word segmenter for sighan bakeoff 2005. In *SIGHAN*, pages 168–171, 2005.

[Wang and Chang, 2016] Wenhui Wang and Baobao Chang. Graph-based dependency parsing with bidirectional lstm. In *ACL*, pages 2306–2315, August 2016.

[Wang and Kan, 2013] Aobo Wang and Min-Yen Kan. Mining informal language from chinese microtext: Joint word recognition and segmentation. In *Proceedings of the 51st ACL*, pages 731–741, Sofia, Bulgaria, 2013.

[Wang and Xue, 2014] Zhiguo Wang and Nianwen Xue. Joint pos tagging and transition-based constituent parsing in chinese with non-local features. In *Proceedings of the 52nd ACL*, pages 733–742, 2014.

[Wang *et al.*, 2013] Aobo Wang, Min-Yen Kan, Daniel Andrade, Takashi Onishi, and Kai Ishikawa. Chinese informal word normalization: an experimental study. In *IJCNLP*, pages 127–135, Nagoya, Japan, 2013.

[Xue, 2003] Nianwen Xue. Chinese word segmentation as character tagging. *International Journal of Computational Linguistics and Chinese Language Processing*, 8(1), 2003.

[Zhang and Clark, 2007] Yue Zhang and Stephen Clark. Chinese segmentation with a word-based perceptron algorithm. In *ACL*, pages 840–847, 2007.

[Zhang and Clark, 2011] Yue Zhang and Stephen Clark. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, 37(1):105–151, 2011.

[Zhang *et al.*, 2013] Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. Chinese parsing exploiting characters. In *Proceedings of the 51st ACL*, pages 125–134, 2013.

[Zhang *et al.*, 2016] Meishan Zhang, Yue Zhang, and Guohong Fu. Transition-based neural word segmentation. In *ACL*, pages 421–431, Berlin, Germany, August 2016.

[Zhao *et al.*, 2006] Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. Effective tag set selection in chinese word segmentation via conditional random field modeling. In *Proceedings of PACLIC*, pages 87–94, 2006.

[Zhao, 2009] Hai Zhao. Character-level dependencies in chinese: Usefulness and learning. In *Proceedings of the EACL*, pages 879–887, 2009.

[Zheng *et al.*, 2013] Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. Deep learning for Chinese word segmentation and POS tagging. In *EMNLP*, pages 647–657, 2013.

[Zhou *et al.*, 2015] Hao Zhou, Yue Zhang, Shujian Huang, and Jiajun Chen. A neural probabilistic structured-prediction model for transition-based dependency parsing. In *Proceedings of the ACL*, 2015.

[Zhu *et al.*, 2013] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. Fast and accurate shift-reduce constituent parsing. In *ACL*, pages 434–443, 2013.