Chinese Sentiment Analysis Exploiting Heterogeneous Segmentations

Da Pan, Meishan Zhang, Guohong Fu

School of Computer Science and Technology, Heilongjiang University
Harbin 150080, China
pandacs@live.cn, mason.zms@gmail.com, ghfu@hotmail.com

Abstract The Chinese language is a character-based language, with no explicit separators between words like English. Traditionally, word segmentation is conducted to convert Chinese sentences into word sequences, thus the same framework of English sentiment analysis can be exploited for Chinese. These work uses a specified word segmentor as a prerequisite step, yet ignores the fact that different segmentation styles exist in Chinese word segmentation, such as CTB, PKU, MSR and etc. In this paper, we study the influences of these heterogeneous segmentations for Chinese sentiment analysis, and then integrate these segmentations, based on both discrete and neural models. Experimental results show that different segmentations do affect the final performances, and the integrated models can achieve better performances.

Keywords: Sentiment Analysis; Heterogeneous Segmentations; Neural Network

1 Introduction

Sentiment analysis has receiving extensive attentions in recent years [1, 2, 3, 4]. A number of work has been proposed. Earlier work extracts discrete features such as word unigrams, bigrams and trigrams [5, 6], and then feed them into a classifier such as max-entropy (ME), support vector machine (SVM) to achieve the final goal [7, 8, 9, 10]. Recently, neural models dominate the research on sentiment classification [11], which exploit word embeddings as basic inputs, and then construct a deep neural network such as a convolution neural network to extract features automatically.

All these work assumes that input sentences are word sequences. The assumption is generally accepted in most cases such as English, which have explicit separators between words. However, it is an exception for the Chinese language.

王小丽心中有点儿不是滋味。(Xiaoli Wang was feeling slight upsetting.)	
王小丽| 心中| 有| 点儿| 不| 是| 滋味| 。	#CTB segmentation style
王| 小丽| 心中| 有| 点儿| 不| 是| 滋味| 。	#PKU segmentation style
王小丽| 心中| 有点儿| 不是| 滋味| 。	#MSR segmentation style

Figure 1. The CTB, PKU and MSR segmentation style with a microblog

In order to align with these work, we usually have a prerequisitive step, word segmentation, to convert original character-based Chinese sentences into word-based. The segmentation is conducted by a well-established segmentor, such as Stanford NLP Tools [12], LTP [13] and THULAC [14].

One problem arises when the Chinese word sequences are produced by a certain segmentor, since it complies with a fixed segmentation style of Chinese, and actually there is no uniform standards for Chinese word segmentation. For example, in the well-known open competition of SIGHAN 2005 [15], there are four different segmentation standards for Chinese word segmentation. There comes a question that which standard is more suitable for Chinese sentiment analysis? However little work has focused on this question.

In this paper, we present the first study of heterogeneous word segmentations for Chinese sentiment analysis to our knowledge. We choose three popular segmentation standards, including CTB, PKU and MSR. Figure 1 shows an example sentence with the three heterogeneous segmentations. As shown in the figure, we find that CTB differs with PKU in the person name, and MSR tends to get longer words. Actually, there are many other differences between the three segmentations, which can be referred to their guidelines for details[1].

We compare three different word segmentations under two settings, one exploiting discrete models and the other exploiting neural models. Further, we integrate three heterogeneous word segmentations, to observe whether better performances can be achieved. We conduct experiments on a manually-collected corpus. Experimental results demonstrate that three different standards can lead to different performances. In addition, we find the integrated models can bring better performances for Chinese sentiment analysis under both discrete and neural settings. We discuss these results to show possible reasons of our observation.

2 Related Work

There has been a lot of work for sentiment analysis [16, 6, 17, 2, 18, 19]. These work largely can be categorized into two types, namely discrete models and neural models. Discrete models explore manually-crafted features, which are designed by composing atomic features from internal information such as words in sentences and from external information such as sentiment lexicons manually. Feature engineering is crucial for these models including feature selection.

In contrast with discrete models, neural models do not rely on the extensive feature engineering work. These models require certain word embeddings to convert word sequences into vector sequences, and then build a neural network structure to abstract features from vector sequences. Typical work of word embeddings aimed for sentiment analysis includes Duyu Tang et al [11], Yafeng Ren et al [20], and typical neural network structures includes convolution neural works [21], recurrent neural networks [22] and recursive neural network [23].

Chinese sentiment analysis usually follows the same framework as the above related work. Most work focuses their interests on building better corpus or fea-

[1] http://www.sighan.org/bakeoff2005/

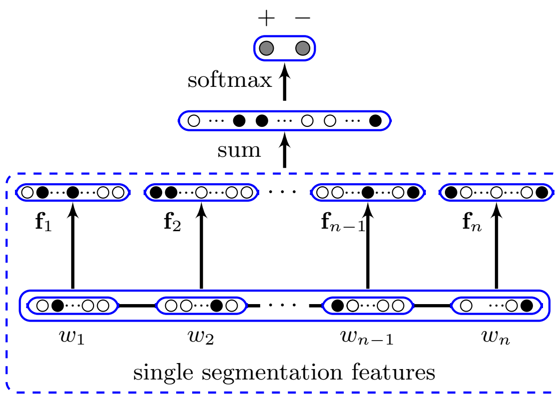


Figure 2. The baseline discrete model (• denotes 1, ◦ denotes 0).

tures. For example, Xiaojun Wan proposed a bilingual method [24]. These work generally ignore the influence of word segmentations by exploiting segmentors such as Stanford NLP tools [12], LTP [13] and THULAC [14]. Another line of previous studies that exploits lower-level information to improve the main classification performance [25, 26]. There exists work that directly using Chinese characters as basic units, which is out of our concern [27].

3 Discrete Models

In our task, sentiment analysis is regarded as a binary classification problem, where − denotes a negative microblogs, and + denotes a positive microblogs. In this section, we introduce two discrete models based on single segmentation and heterogeneous segmentations, respectively.

3.1 The Baseline Model

In this section, we introduce the baseline discrete model for Chinese sentiment analysis, with a single word sequence as input for a weibo sentence. The single word sequence can be CTB, PKU or MSR styles. We can use this baseline model to study the influences of the three different segmentation standards, and to see which one is more suitable for Chinese sentiment analysis.

Figure 2 shows the framework of the baseline discrete model. Given an input microblog, after word segmentation, we obtain an word sequence by $w_1 w_2 \cdots w_n$. We then extract discrete features at each position i ($1 \leq i \leq n$) according to specified feature templates. In this work, we exploit three kinds of features, including unigram, bigram and trigram. At position i, the unigram feature is w_i, the bigram feature is $w_{i-1} \circ w_i$, and the trigram feature is $w_{i-1} \circ w_i \circ w_{i+1}$, thus we get the feature vector $\mathbf{f}_i = \{w_i, w_{i-1} \circ w_i, w_{i-1} \circ w_i \circ w_{i+1}\}$.

After obtaining the feature sequence $\mathbf{f}_1 \mathbf{f}_2 \cdots \mathbf{f}_n$, we sum all the features so that the entire sentence is represented, which we call this process as (sum) pooling, in order to have a better understanding in contrast with neural models which

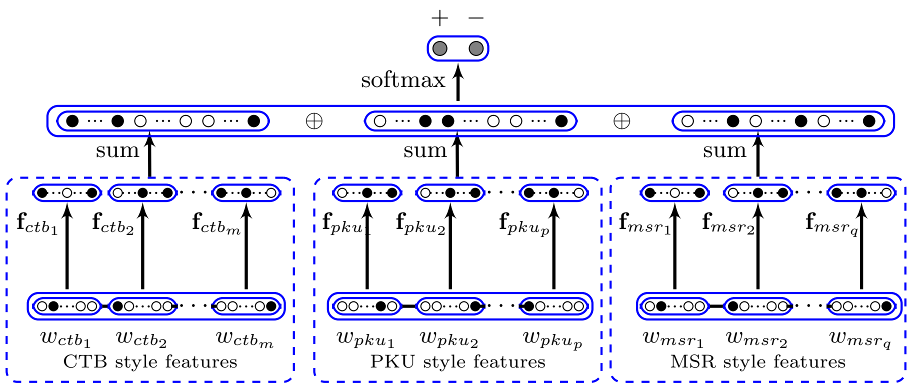


Figure 3. The proposed discrete model (• denotes 1, ∘ denotes 0).

we will introduce the next section. The process can be formulates as follows:

$$\mathbf{f} = \sum_{i=1}^{n} \mathbf{f}_i \tag{1}$$

Finally we apply a linear classifier on the feature vector $\mathbf{f}$ to computer the output scores:

$$\mathbf{o} = \mathsf{W}_o \mathbf{f} \tag{2}$$

where the matrix W_o is a parameter of the discrete model, and $\mathbf{o}$ is the output two-dimensional vector, with one denoting the positive score and the other denoting the negative score. If the positive score is larger, we label the input microblog as positive, otherwise negative.

3.2 The Proposed model based on multiple segmentations

In order to support Chinese sentiment analysis with multiple segmentations, we extend the baseline discrete model. The adaption is very easy by simply adding the feature vectors extracted from multiple segmentations. The framework of our proposed model based on multiple segmentations is shown in Figure 3.

Given an input microblog, we first segment the sentences into three word sequences based on the CTB, PKU and MSR segmentation styles: $w_1^{ctb} w_2^{ctb} w_m^{ctb}$, $w_1^{pku} w_2^{pku} w_p^{pku}$ and $w_1^{msr} w_2^{msr} w_q^{msr}$. Then we follow the baseline discrete model, obtaining the feature vectors ($\mathbf{f}^{ctb}$, $\mathbf{f}^{pku}$ and $\mathbf{f}^{msr}$) of the three word sequences, respectively. Last, we get the final feature vector $\mathbf{f}$ by:

$$\mathbf{f} = \mathbf{f}^{ctb} + \mathbf{f}^{pku} + \mathbf{f}^{msr} \tag{3}$$

And the $\mathbf{f}$ is fed into a linear classifier (the same as Formula 2) to compute the sentence's sentiment polarity scores.

4 Neural Models

In this section, we introduce our neural models for Chinese sentiment analysis. In contrast with discrete models, neural models exploit dense real-valued word embeddings as inputs, and abstract features from the input vectors by neural layers. Neural models have two main advantages: (1) First as low-dimensional real-valued features are used, these models avoid feature sparsity problem; (2) Second feature combinations are conducted by neural layers automatically, thus manually-crafted features are no longer required, saving the work of feature engineering. In the following, we first present the baseline neural model with only a single word sequence for each sentence. Then we extend this baseline to adapt multiple segmentation inputs.

Our baseline and proposed neural model based on the LSTM-RNN structure [28] which uses dense vectors to represent features. The dimension of dense vectors is much lower than one-hot vectors. Both of them use neural network to extract dense real-valued features $\mathbf{h}$ from microblogs.

4.1 The Baseline Neural Model

First, we introduce the baseline neural model of Chinese sentiment analysis. Figure 4 shows the framework. For a given sentence with the word sequence $w_1 w_2 \cdots w_n$, we first obtain word embeddings of w_i from a lookup matrix E, resulting in $\mathbf{e}_1\mathbf{e}_2 \cdots \mathbf{e}_n$. Then we consider a window size of three to extend the representation at each position i ($1 \le i \le n$), obtaining a sequence of $\mathbf{x}_1\mathbf{x}_2 \cdots \mathbf{x}_n$ which is in spirit similar to the trigrams in the baseline discrete model. Formally, $\mathbf{x}_i = \mathbf{e}_{i-1} \oplus \mathbf{e}_i \oplus \mathbf{e}_{i+1}$, which is a concatenation of embeddings of previous word, current word and next word.

Upon the sequence of $\mathbf{x}_1\mathbf{x}_2 \cdots \mathbf{x}_n$, we build a bi-directional LSTM-RNN [29] to model a microblog, which is able to capture semantic and syntactic information of the input sentence automatically, and meanwhile avoids the gradient vanishing and exploding problem during training. The bi-directional LSTM-RNN means two LSTM-RNNs, one being computed from left-to-right ($\mathbf{h}_1^l\mathbf{h}_2^l \cdots \mathbf{h}_n^l$) and the other being computed from right-to-left ($\mathbf{h}_1^r\mathbf{h}_2^r \cdots \mathbf{h}_n^r$). Bi-directional LSTM-RNN has been widely used in a number of NLP tasks to achieve similar goals[30, 31, 32]. Taking the left-to-right LSTM-RNN as an example, the hidden sequence ($\mathbf{h}_1^l\mathbf{h}_2^l \cdots \mathbf{h}_n^l$) is computed as following:

$$
\begin{aligned}
\mathbf{ig}_i^l &= \sigma(\mathsf{W}_{ig}^l\mathbf{x}_i + \mathsf{U}_{ig}^l\mathbf{h}_{i-1}^l + \mathsf{V}_{ig}^l\mathbf{ce}_{i-1}^l + \mathbf{b}_{ig}^l) \\
\mathbf{fg}_i^l &= \sigma(\mathsf{W}_{fg}^l\mathbf{x}_i + \mathsf{U}_{fg}^l\mathbf{h}_{i-1}^l + \mathsf{V}_{fg}^l\mathbf{ce}_{i-1}^l + \mathbf{b}_{fg}^l) \\
\mathbf{ce}_i^l &= \mathbf{fg}_i^l \odot \mathbf{ce}_{i-1}^l + \mathbf{ig}_i^l \odot \tanh(\mathsf{W}_{ce}^l\mathbf{x}_i + \mathsf{U}_{ce}^l\mathbf{h}_{i-1}^l + \mathbf{b}_{ce}^l) \\
\mathbf{og}_i^l &= \sigma(\mathsf{W}_{og}^l\mathbf{x}_i + \mathsf{U}_{og}^l\mathbf{h}_{i-1}^l + \mathsf{V}_{og}^l\mathbf{ce}_i^l + \mathbf{b}_{og}^l) \\
\mathbf{h}_i^l &= \mathbf{og}_i^l \odot \tanh(\mathbf{ce}_i^l),
\end{aligned}
\tag{4}
$$

where $\mathsf{W}^l, \mathsf{U}^l, \mathsf{V}^l, \mathbf{b}^l$ are all the model parameters, and $\odot$ denotes the element-wise Hadamard product. In the equations, $\mathbf{h}_i^l$ is computed by the previous hidden

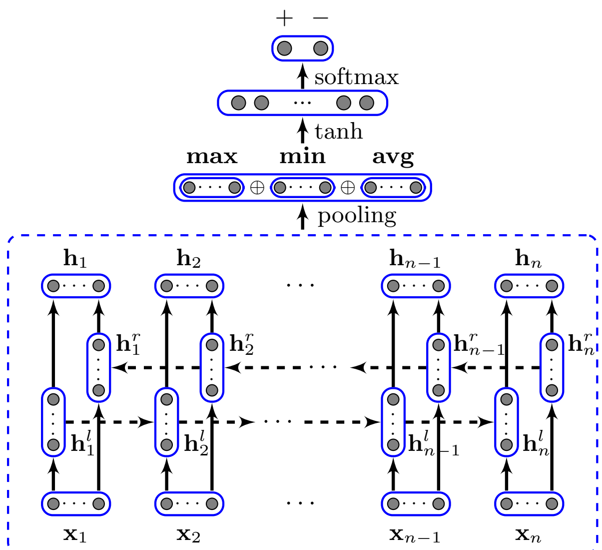


Figure 4. The baseline neural model (● denotes a real-value feature).

vector $\mathbf{h}^l_{i-1}$, cell vector $\mathbf{ce}^l_{i-1}$, and the current input vector $\mathbf{x}_i$. The cell structure $\mathbf{ce}$ is used to deliver the long-term information, which is controlled by three gates, namely input gate $\mathbf{ig}$, output gate $\mathbf{og}$ and forget gate $\mathbf{fg}$, respectively.

Similarly, we obtain the right-to-left hidden vector sequence $(\mathbf{h}^r_1\mathbf{h}^r_2\cdots\mathbf{h}^r_n)$ by the reverse order, and the corresponding model parameters are $\mathsf{W}^r, \mathsf{U}^r, \mathsf{V}^r, \mathbf{b}^r$. After hidden sequences of both directions are computed, we concatenate them at each position one by one, receiving the final hidden vectors $\mathbf{h}_1\mathbf{h}_2\cdots\mathbf{h}_n$ ($\mathbf{h}_i = \mathbf{h}^l_i \oplus \mathbf{h}^r_i$).

Furthermore, we use three different pooling functions to obtain the sentence features, which project the variable length hidden sequence into a fixed-dimensional vector $\mathbf{h}$. The three pooling functions are namely max, min, avg (average). Concretely, given a hidden sequence $(\mathbf{h}_1\mathbf{h}_2\cdots\mathbf{h}_n)$, pooling is executed by $\mathbf{h} = \sum_{i=1}^{n} \alpha_i \odot \mathbf{h}_i$. For max pooling, $a^{max}_{i,j}$ equals 1 only when $i = \arg\max_s(\mathbf{h}_{s,j})$, and 0 otherwise; for min pooling, $\alpha^{min}_{i,j}$ equals 1 only when $i = \arg\min_s(\mathbf{h}_{s,j})$, and 0 otherwise, where $s \in [1, n]$; for avg pooling, $a^{avg}_i = \frac{1}{n}$. Correspondingly, we can obtain three sentence vectors $\mathbf{h}^{max}, \mathbf{h}^{min}, \mathbf{h}^{avg}$ by the three pooling methods, respectively. Finally, we use a non-linear feed-forward layer to get the final, the output can be computed by:

$$\begin{aligned} \mathbf{s} &= tanh\big(\mathsf{W}_s(\mathbf{h}^{max} \oplus \mathbf{h}^{min} \oplus \mathbf{h}^{avg}) + \mathbf{b}_s\big), \\ \mathbf{o} &= \mathsf{W}_o\mathbf{s}, \end{aligned} \tag{5}$$

where the $\mathsf{W}_s, \mathsf{W}_o$ and $\mathbf{b}_s$ are the model parameters, and the $\oplus$ denotes the concatenation.

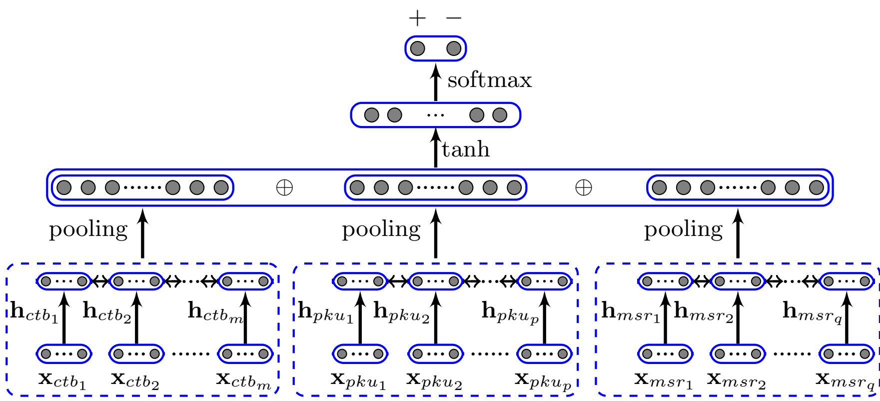


Figure 5. The proposed neural model (• denotes a real-value feature).

4.2 The Proposed Neural Model with multiple segmentation

We follow the method of extending the baseline discrete model into the proposed discrete model with multiple segmentations to adapt the baseline neural model being able to receive multiple segmentations. Figure 5 shows the overall framework of our proposed neural model.

In contrast to the baseline neural model, it has three components, which correspond to the three segmentation styles, respectively. Each component uses a bi-directional LSTM-RNN to model the input sentences. Similarly, three components are used to extract dense real-valued features from word sequence of different segmentation styles, respectively. Then we project the hidden sequences into a fix-dimensional vectors with the pooling functions to obtain the representation vectors $h_{ctb}, h_{pku}, h_{msr}$, respectively. Finally, we exploit a non-linear feed-forward layer to combine multiple segmentation sentence vectors for further classification, which is in spirit similar to the baseline model, and the corresponding model parameters are $\mathsf{W}_s, \mathsf{W}_o, \mathbf{b}_s$. The output nodes are computed by:

$$
\begin{aligned}
\mathbf{h}_{ctb} &= \mathbf{h}_{ctb}^{max} \oplus \mathbf{h}_{ctb}^{min} \oplus \mathbf{h}_{ctb}^{avg} \\
\mathbf{h}_{pku} &= \mathbf{h}_{pku}^{max} \oplus \mathbf{h}_{pku}^{min} \oplus \mathbf{h}_{pku}^{avg} \\
\mathbf{h}_{msr} &= \mathbf{h}_{msr}^{max} \oplus \mathbf{h}_{msr}^{min} \oplus \mathbf{h}_{msr}^{avg} \\
\mathbf{s} &= tanh\big(\mathsf{W}_s(\mathbf{h}_{ctb} \oplus \mathbf{h}_{pku} \oplus \mathbf{h}_{msr}) + \mathbf{b}_s\big) \\
\mathbf{o} &= \mathsf{W}_o \mathbf{s}
\end{aligned}
\tag{6}
$$

5 Training

The cross-entropy loss is exploited as our training objective function for both discrete and neural models under supervised learning. Our goal is to minimize the objective loss with a set of training examples $(\mathbf{x}_i, \mathbf{y}_i)|_{i=1}^{N}$, and plus with a

Table 1. Experimental corpus statistics.

	train	development	test
positive	5719	633	2117
negative	6126	701	1168

Table 2. Hyper-parameter values in our models.

Type	hyper-parameters
neural	$\mathbf{h}_{lstm-rnn} = 100$, $\mathbf{e}_{word} = 50$, $p_{drop} = 0.25$
discrete and neural	$\lambda = 10^{-8}$, $\alpha = 0.01$

l_2-regularization term,

$$L(\theta) = -\sum_{i=1}^{N} \log p_{y_i} + \frac{\lambda}{2} \parallel \theta \parallel^2, \tag{7}$$

where θ is the set of model parameters, and the probability of the oracle output $\mathbf{y}_i$ is denoted by p_{y_i}, which is computed using softmax over the output vector $\mathbf{o}$ for both discrete and neural models. Online AdaGrad [33] is used to minimize the objective function for models.

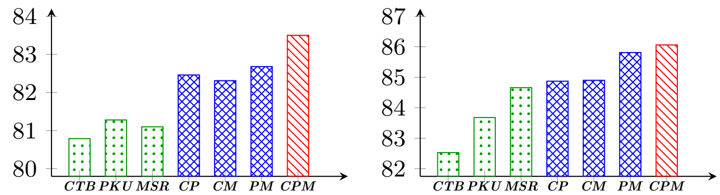
All the matrix and vector parameters are randomly initialized by uniform sampling in (-0.01, 0.01), the look-up table E for word embedding are also a model parameter. The values of E are assigned by pre-training on a large-scale segmented corpus. In this work, we use *word2vec*[2] to pre-train word embeddings on a collected weibo corpus, with the word sequences obtained by our segmentors automatically.

6 Experiments

6.1 Experimental Settings

We use the data of NLP&CC2014 Share Task 1, which is collected for emotion analysis in weibo texts. The original task aims to determine the emotion of a weibo text and to classify the weibo texts by its emotion category, including anger, disgust, rear, happiness, like, sadness and surprise. In our work, we exploit the data for Chinese sentiment analysis by labeling the weibo texts with emotions of anger, disgust sadness or fear being negative and being positive with emotions of happiness or like. We have a pre-process step to normalize nickname, URL, emoji and hashtag in weibo texts, and further, we use ZPar [34] to segment weibo texts into word sequences with different segmentation styles. The models of ZPar are trained using CTB, PKU and MSR corpora by ourselves. Table 1 shows the corpus statistics.

[2] http://word2vec.googlecode.com/



(a) discrete models (b) neural models

Figure 6. Experiment results on test dataset, where the boxes with green dots denotes the performances of baseline model, the boxes with blue crosshatch lines denotes models based on two segmentation styles and the boxes with red north west lines denotes the model based on three segmentation styles.

Table 3. Final results on both discrete and neural models. CPM denotes the integration of three heterogeneous segmentations CTB, PKU and MSR

Model	baseline(CTB)	baseline(PKU)	baseline(MSR)	proposed(CPM)
Discrete	80.79	*81.28*	81.10	**83.50**
Neural	82.53	83.68	*84.66*	**86.06**

We adopt accuracy to evaluate our model. There are several hyper-parameters in our discrete and neural models, which are tuned according to the developmental performances. The values of parameters are shown in Table 2, where both of discrete and neural models use the same regularization parameter λ and initial updating value α. The other parameters in neural models, including the dimension size $\mathbf{h}_{lstm-rnn}$ of LSTM-RNN layers, dimension size of word embeddings $\mathbf{e}_{word}$ and the probability of dropout p_{drop} are shown is Table 2 as well.

6.2 Results

We study the influences of heterogeneous segmentations on sentiment analysis, in details, by comparing all possible combinations of these segmentations. The results are shown in Figure 6. We can see that different segmentation styles lead to different accuracies in baseline models. Integrating any two styles of segmentation can obtain higher results than the baseline models, and further our final models which integrate three heterogeneous segmentations achieve the best performance.

Table 3 lists the results of our baseline and final models, including performances with both discrete and neural settings, respectively. For the discrete setting, the best model with single-segmentation input can achieve an accuracy of 81.28%, while it is 84.66% under the neural setting. After three heterogeneous segmentations are combined as input, the discrete model obtains an accuracy of 83.50%, which is 2.22% higher than the single-segmentation input. The neural model with heterogeneous segmentations can also outperform the single segmentation input baseline, achieving an accuracy of 86.06%, gaining an improvement of 1.4%. Overall, the results show that neural models can achieve better performances than the discrete models with similar inputs.

6.3 Model Analysis

In this section, we show several intrisic analysis to interpret the benefits by exploiting multi-segmentations. As for the discrete setting, the number of high-frequency features is greatly boosted by multi-segmentations as input. As known, high-frequency features are highly useful for discrete models, since low-frequency features suffer the sparsity problem. Thus it is reasonable that our final discrete model with the heterogeneous segmentations achieves the best performance.

Similarly, under the neural setting, word embeddings are useful external resources, which are learnt from unlabeled texts. For the models using single-segmentation as input, only one kinds of embeddings can be used. While for the multiple segmentation input model, we can utilize all three different embeddings, thus our results are consist with the common intuition.

Moreover, since pratical Chinese segmentors are trained by fixed annotated corpora, the output word sequences inevitably have incorrect segmentations. When these results are fed into the final-classifiers, it can lead to error propagation for final sentiment analysis. While for our proposed models with multiple segmentations as inputs, the problem can be alleviated as segmentations of different styles have different error distributions.

7 Conclusion

We investigated the influences of heterogeneous word segmentations for Chinese sentiment analysis, which has been generally ignored in previous work. First, we showed that the segmentation style does affect the sentiment classification performances. A different segmentation style can lead to a gap of 2% of the final performances. Second, we combined the multiple segmentations of different styles under both discrete and neural setting, finding better accuracies can be achieved by such an integration.

8 Acknowledgments

We thank the anonymous reviewers for their constructive comments, which helped to improve the paper. This study was supported by Natural Science Foundation of Heilongjiang Province under Grant No.F2016036, National Natural Science Foundation of China under Grant No.61170148, and the Returned Scholar Foundation of Heilongjiang Province, respectively.

References

[1] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th ACL*, pages 151–160, 2011.

[2] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[3] Guohong Fu, Yu He, Jiaying Song, and Chaoyue Wang. Improving chinese sentence polarity classification via opinion paraphrasing. *CLP 2014*, page 35, 2014.

[4] Duy-Tin Vo and Yue Zhang. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the 29th IJCAI*, pages 1347–1353, 2015.

[5] Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th ACL: Short Papers-Volume 2*, pages 90–94, 2012.

[6] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[7] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the EMNLP*, pages 79–86, July 2002.

[8] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th ACL: HLT-Volume 1*, pages 142–150, 2011.

[9] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[10] Ronen Feldman. Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89, 2013.

[11] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL*, pages 1555–1565, 2014.

[12] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[13] Wanxiang Che, Zhenghua Li, and Ting Liu. Ltp: A chinese language technology platform. In *Proceedings of the 23rd COLING: Demonstrations*, pages 13–16, 2010.

[14] Zhongguo Li and Maosong Sun. Punctuation as implicit annotations for chinese word segmentation. *Computational Linguistics*, 35(4):505–512, 2009.

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

[16] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th ACL*, pages 417–424, 2002.

[17] Guohong Fu and Xin Wang. Chinese sentence-level sentiment classification based on fuzzy sets. In *Proceedings of the 23rd COLING: Posters*, pages 312–319, 2010.

[18] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Unsupervised sentiment analysis with emotional signals. In *Proceedings of the 22nd WWW*, pages 607–618, 2013.

[19] Bishan Yang and Claire Cardie. Context-aware learning for sentence-level sentiment analysis with posterior regularization. In *ACL (1)*, pages 325–335, 2014.

[20] Yafeng Ren, Yue Zhang, Meishan Zhang, and Donghong Ji. Context-sensitive twitter sentiment classification using neural network. AAAI, 2016.

[21] Cícero Nogueira dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78, 2014.

[22] Xin Wang, Yuanchao Liu, Chengjie Sun, Baoxun Wang, and Xiaolong Wang. Predicting polarities of tweets by composing word embeddings with long short-term memory. In *Proceedings of the ACL and the IJCNLP*, volume 1, pages 1343–1353, 2015.

[23] Mohit Iyyer, Peter Enns, Jordan L Boyd-Graber, and Philip Resnik. Political ideology detection using recursive neural networks. In *ACL (1)*, pages 1113–1122, 2014.

[24] Xiaojun Wan. Co-training for cross-lingual sentiment classification. In *Proceedings of ACL and IJCNLP: Volume 1-Volume 1*, pages 235–243, 2009.

[25] Ainur Yessenalina, Yisong Yue, and Claire Cardie. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1046–1056, Cambridge, MA, October 2010. Association for Computational Linguistics.

[26] Duyu Tang, Furu Wei, Bing Qin, Li Dong, Ting Liu, and Ming Zhou. A joint segmentation and classification framework for sentiment analysis. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 477–487, Doha, Qatar, October 2014. Association for Computational Linguistics.

[27] Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

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

[30] Jie Zhou and Wei Xu. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd ACL and the 7th IJCNLP (Volume 1: Long Papers)*, pages 1127–1137, Beijing, China, July 2015.

[31] Di Wang and Eric Nyberg. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd ACL and the 7th IJCNLP (Volume 2: Short Papers)*, pages 707–712, Beijing, China, July 2015.

[32] Pengfei Liu, Shafiq Joty, and Helen Meng. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the EMNLP*, pages 1433–1443, Lisbon, Portugal, September 2015.

[33] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159, 2011.

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