Ensemble of Neural Networks with Sentiment Words Translation for Code-Switching Emotion Detection

Tianchi Yue, Chen Chen, Shaowu Zhang(✉), Hongfei Lin, and Liang Yang

Dalian University of Technology, Dalian 116023, Liaoning, China
zhangsw@dlut.edu.cn

Abstract. Emotion detection in code-switching texts aims to identify the emotion labels of text which contains more than one language. The difficulties of this task include problems in bridging the gap between languages and capturing crucial semantic information for classification. To address these issues, we propose an ensemble model with sentiment words translation to build a powerful system. Our system first constructs an English-Chinese sentiment dictionary to make a connection between two languages. Afterwards, we separately train several models include CNN, RCNN and Attention based LSTM model. Then combine their classification results to improve the performance. The experiment result shows that our method has a good effect and achieves the second place among nineteen systems.

Keywords: Emotion detection · Code-switching · Neural networks
Sentiment words translation

1 Introduction

With the rapid development of the Internet, more and more people tend to express their emotions through text in the online community. Emotion detection has become a hot research topic. Previous work on emotion detection mostly focused on analyzing emotions from monolingual text [1]. However, many users often post code-switching texts in social media, and some researchers start to study how emotions are expressed with code-switching texts.

Code-switched emotion detection is a challenging task in emotion analysis which aims to assign emotion labels to code-switching texts. Code-switching texts contain more than one language [2]. For example, the code-switching instances below show that happiness emotion in [a] is expressed by monolingual form and sadness emotion in [b] is expressed by bilingual form.

[a] 美好假期已经开始, have a nice time。 (happiness)
(The happy vocation has begun, have a nice time.)
[b] 上了一天的课，嗓子hold不住了啊。 (sadness)
(I have been teaching the whole day, my throat can't take it any more)

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11109, pp. 411–419, 2018.
https://doi.org/10.1007/978-3-319-99501-4_37

Most existing methods respectively focus on emotion detection [3–6] and code-switching text analysis [7, 8]. Relatively few researches [9, 10] consider detecting emotion in code-switching text. The problems of this task are how to bridge the gap to between two languages and how to model the code-switching text to detect emotion.

Motivated by the significant improvements of deep neural networks in many NLP tasks [11], we propose an effective ensemble model with sentiment words translation to tackle these problems. Our system mainly consists of three parts. Firstly, we utilize both Chinese and English sentiment lexicons and an English-Chinese dictionary to translate the English sentiment words in a sentence to Chinese. Afterwards, we separately train several models include CNN, RCNN and Attention based LSTM model. Finally, we ensemble their classification results to improve the performance.

The main contributions of our work can be summarized as follows:

- We construct an English-Chinese sentiment words dictionary to build a connection between bilingual forms.
- We propose an ensemble of neural networks to detect emotion which can extract local features, focus on silent parts and capture contextual information of a whole sentence.
- Experimental result indicates that our approach outperforms several base methods and has a good effect.

2 Related Works

Most existing emotion detection methods focus on the monolingual text. Yang et al. propose an emotion-aware topic model to build a fine-grained domain-specific emotion lexicon [4]. Li et al. build a factor graph to incorporate both the label and context dependency for emotion classification [5]. Neural networks are also introduced because of the good performance in many NLP tasks. Abdul-Mageed et al. use gated recurrent neural networks (GRNNs) for fine-grained emotion detection [6]. Emotion detection can be formalized as the task of classifying whether the post belongs to the given emotion or not. There are some neural networks proposed for text classification, including convolutional neural network (CNN) [12], recurrent convolutional neural network (RCNN) [13], and hierarchical attention network (HAN) [14]. Inspired by these neural networks, we propose an ensemble model to combine the virtues of them.

Recently with the trends of code-switching text on social media, several methods have been proposed to detect emotions of bilingual texts. Lee et al. construct a Chinese-English code-switching corpus for emotion detection and combine maximum entropy based Chinese text classifier and English text classifier [1]. Wang et al. use both the machine translation based bilingual information and sentimental information to build a relation between two languages [9]. Wang et al. apply a bilingual attention model to focus on important words on both monolingual and bilingual contexts and combine the attention vectors to predict the emotion [10].

3 System Preprocessing

3.1 Text Preprocessing

Many texts contain hyperlinks to other web pages which are senseless for emotion detection. We use regular expression to replace all the links with the token URL.

We convert all the English characters into lower case and remove some special tokens such as '\xa0' and '\u3000'.

We further segment the tokens which contain bilingual words. For example, split 'high起' into 'high' and '起'.

3.2 Word Embeddings

Word embeddings are continuous low-dimensional vector space representations of words which can better capture semantic and syntactic information. Word embeddings play an important role in sentiment analysis with neural networks [11]. The pre-trained word embeddings from a large task-related unlabelled data can enhance the performance of classifiers. In our experiments, all word embeddings are initialized by word2vec [12], the word vectors are pre-trained on a large unlabelled corpora which is collected from Sina Weibo. We measure the performance of some bilingual pairs from our pre-trained word embeddings, we find some English words and their Chinese translation have been mapped to close vector space. Thus, in this paper, we only translate the English sentiment words.

3.3 Sentiment Words Translation

Sentiment words are vital for emotion detection. If the English sentiment words in text have been mapped to a wrong space, the neural model can't predict the emotion correctly. To bridge the gap between two languages, we build an English-Chinese sentiment words dictionary. Firstly we use iciba dictionary[1] to translate all of English sentiment words [16] to Chinese. Then choose the word from candidate translation which is also included in a Chinese sentiment lexicon[2] as the only translation. Finally we get an English-Chinese sentiment dictionary with the length of 4040 and we utilize it to map the English sentiment words in text to the Chinese sentiment words.

4 Ensemble Model

In this paper, we respectively implement three models include CNN, RCNN and Attention based LSTM. For each model we first embed every word in the sentence into a word vector space so that a sentence can be represented as a matrix $X = [x_1, x_2, \ldots, x_t]$

[1] http://dict-co.iciba.com/search.php?word=good.

[2] DUTIR Chinese Sentiment Lexicon: http://ir.dlut.edu.cn/EmotionOntologyDownload.aspx.

where $x_t \in R^d$ and t is the length of sequence. After that, we use different networks to obtain the sentence representation individually. Finally, we apply the same classification layer to get prediction probabilities.

4.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) uses the convolution layers to learn local features of text. Our model of CNN, illustrated on Fig. 1 is similar to Kim [12].

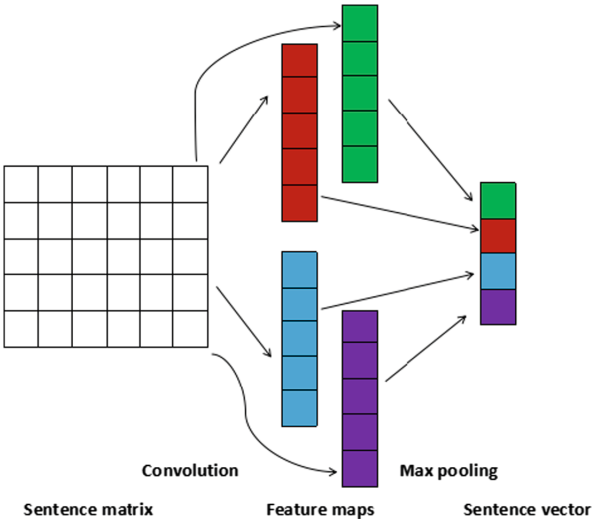

Fig. 1. CNN

We apply several filters of different sizes to the words representation matrix X to extract n-gram features, with general formulations:

$$c_i = f(WX_{i:i+h-1} + b) \tag{1}$$

We then feed the extracted feature maps to a max-pooling layer to get the most important features $c_{max} = \max_{i=1} c_i$. We combine all the c_{max} of each filter into one vector s to represent the whole sentence.

4.2 Long Short-Term Memory (LSTM)

For Attention and RCNN, we use Bi-LSTM to get contextual information of sentence. Let's first briefly introduce LSTM. Long Short-Term Memory network (LSTM) can avoid gradient vanishing and expansion, it is good at learning long-term dependencies

and modeling sequences [17]. There are three gates and a memory cell in the LSTM architecture. Formally, LSTM cell can be computed as follows:

$$i_k = \sigma\left(W^i x_k + V^i h_{k-1} + b^i\right) \tag{2}$$

$$f_k = \sigma\left(W^f x_k + V^f h_{k-1} + b^f\right) \tag{3}$$

$$o_k = \sigma\left(W^o x_k + V^o h_{k-1} + b^o\right) \tag{4}$$

$$c_k = f_k \odot c_{k-1} + i_k \odot \tan h\left(W^c x_k + V^c h_{k-1} + b^c\right) \tag{5}$$

$$h_k = o_k \odot \tan h(c_k) \tag{6}$$

Where W and V are the weighted matrix and b are biases. σ is the sigmoid function and $\odot$ is element-wise multiplication. h_k is the vector of hidden layer which is the final word representations for text.

Since words in a sentence have strong dependence on each other and bi-directional LSTM can better capture the contextual information in text, we employ the bi-directional LSTM network which consists of a forward and a backward LSTM to learn the representation of each word in a sentence.

$$\begin{aligned} \vec{h}_k &= \vec{\text{LSTM}}(x_k) \\ \overleftarrow{h}_k &= \overleftarrow{\text{LSTM}}(x_k) \\ h_i &= [\vec{h}_k, \overleftarrow{h}_k] \end{aligned} \tag{7}$$

4.3 Attention Based LSTM

Each word has different levels of importance on the representation of the sentence semantic. For example, ‘nice’ plays a more critical role than ‘have’ in summarizing the example sentence [a] from Sect. 1. Thus, we introduce an attention mechanism to capture the important information in a sentence. Our model of attention based LSTM is illustrated on Fig. 2.

After obtaining the hidden state h_t by Bi-LSTM, we use a non linear transformation layer to get u_t as a representation of h_t. We use dot product function between ut and a word level vector v to get the relative importance and a softmax transformation to get the final attention signal α_t. Then we can get the weighted representation of sentence s with the attention signal and it can be used as features for text classification.

$$u_t = \tanh(W_a h_t + b_a) \tag{8}$$

$$\alpha_t = \frac{exp\left(u_t^T v\right)}{\sum_t exp\left(u_t^T v\right)} \tag{9}$$

$$s = \sum_t \alpha_t h_t \tag{10}$$

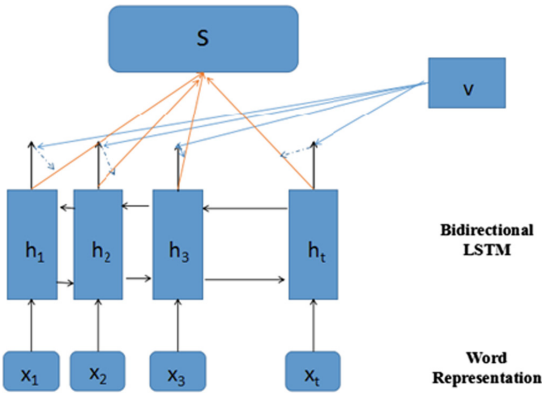

Fig. 2. Attention based LSTM model

4.4 Recurrent Convolutional Neural Network (RCNN)

LSTM is good at capturing the long contextual information. However, it can't sufficiently model the whole sentence since later words are more dominant than earlier words in LSTM. CNN is good at capturing local features while can't capture long gram features. Thus, we use RCNN to tackle these problems. In RCNN network we first apply a Bi-LSTM layer to capture the information h in text which is the same as the Bi-LSTM layer of attention model. Then we further apply a CNN which is identical to Sect. 4.1 on the hidden representation h_t to get the sentence representation s.

4.5 Final Predictions

After getting the sentence vector s, we apply a MLP (Multi-Layer Perception) layer and softmax function to produce the prediction probability.

$$p = \text{softmax}(W_2\sigma(W_1 s + b_1) + b_2) \tag{11}$$

We have five emotions, so we separately train our model for each emotion. Each model is an individual binary classifier. Cross-entropy loss is used as the objective function for training. We also weight the loss function by inverse the frequency of each class for handling the imbalanced data problem.

We have three models for each emotion. Soft voting strategy is used to ensemble these models to reduce variance and improve performance. That means, given a sentence and an emotion, we can get three 2-dimensional output probabilities and average them to identify whether the text expresses the emotion or not.

4.6 Parameters

In our experiments, all word vectors are initialized by word2vec, and out of vocabulary words are initialized by sampling from the uniform distribution $U(-0.01,0.01)$. The dimension of word vectors is 640.

We employ Adam [18] with initial learning rate of 0.0005 for the optimization of models. After the first 3 epochs, we reduce learning rate decay by a factor of 10. We use early stopping which means stop training when the performance of development set doesn't improve in 5 epochs and dropout with rate of 0.2 to avoid overfitting. We set the batch size to 50. For LSTM based models, hidden output size is 300. For CNN based models, filter windows of 2, 3, 4 with 250 feature maps each.

We use the development sets to tune the hyper parameters and select the best model based on performance on the development set, and evaluate on the test set.

4.7 Results Analysis

We conduct the experiment on the dataset of NLPCC 2018 Evaluation Task 1 which contains 6000 instances for training, 728 for development set and 1200 for test set. Each post contains five emotion labels, i.e. happiness, sadness, fear, anger and surprise. The results of system are evaluated by macro-averaged F1 Score.

We compare the performance of different configurations on Table 1. It shows that the basic neural networks significantly outperforms baseline (SVM model). The ensemble model which incorporates three models achieves the best performance. The reason is that our ensemble model is capable of extracting local features, focusing on silent parts and capturing contextual information which will contribute to sentence representation and text classification.

Table 1. Performance results of different models on the test data.

Models	Happiness	Sadness	Anger	Fear	Surprise	MacF1
Baseline	0.587	0.500	0.390	0.108	0.128	0.342
Ensemble NN	**0.715**	0.521	0.541	0.166	0.396	**0.468**
CNN	0.671	0.507	0.493	0.177	**0.404**	0.450
RCNN	0.709	0.541	0.532	0.171	0.344	0.459
ATTENTION	0.685	**0.543**	0.495	0.191	0.336	0.450
CNN + RCNN	0.694	0.513	**0.543**	0.193	0.380	0.464
CNN + ATT	0.679	0.530	0.534	**0.203**	0.387	0.466
RCNN + ATT	0.686	0.525	0.530	0.166	0.360	0.453

We have also found that single model CNN and Attention based LSTM get similar effect. While RCNN achieves the best performance among the single models, because the RCNN model fully considers local features and contextual information. Moreover, systems which ensemble two single models also outperform the other individual models, which demonstrate the effectiveness of ensemble model. Due to the small number of instances and some posts are also hard to classify by manual annotation, we can see the F-score of some emotions is below 0.4.

Table 2 shows the performance of our system (DUTIR_938) compared with the top, the third and the median ranked team. Our system ranks second among the nineteen teams. However, for the comparison with the top system, our system is over 9%

lower in Sadness and Fear emotion detection. We think the word embedding layer and quality of sentiment words translation are the most important factors. Some sentiment related words can't be mapped to correct vector space even with the fine tuning of word embeddings and will result in misclassification.

Table 2. Performace of the top-3 and median-ranked system offered by the organizer

Team	Happiness	Sadness	Anger	Fear	Surprise	MacF1
DeepIntell	0.734	0.616	0.543	0.264	0.418	0.515
DUTIR_938	0.715	0.521	0.541	0.166	0.396	0.468
Shining	0.710	0.652	0.540	0.292	0.139	0.467
Yang_NEU	0.568	0.432	0.351	0.207	0.255	0.363
Baseline	0.587	0.500	0.390	0.108	0.128	0.342

5 Conclusion and Future Work

In this paper we use bilingual sentiment lexicons and an English-Chinese dictionary to build a connection between two languages and then we explore a method which ensembles several neural networks to detect emotion. Our method can also reduce the impact of data imbalance and boost the performance. Our submission result ranks the second place in all of teams and shows the effectiveness of our method.

We find that the translation of bilingual sentiment related words plays an important role in code-switching text emotion detection, for the future work, we will try to utilize more external knowledge to better bridge the gap to between two languages.

Acknowledgements. This work is supported by National Natural Science Foundation of China (61562080, 61632011, 61572102, 61702080).

References

1. Lee, S., Wang, Z.: Emotion in code-switching texts: corpus construction and analysis. In: Eighth Sighan Workshop on Chinese Language Processing, pp. 91–99 (2015)
2. Wang, Z., Lee, S.Y.M., Li, S., et al.: Emotion analysis in code-switching text with joint factor graph model. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(3), 469–480 (2017)
3. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Isa-belle, P. (ed.) Proceedings of the EMNLP 2002, pp. 79–86. ACL, Morristown (2002)
4. Yang, M., Zhu, D., Chow, K-P.: A topic model for building fine-grained domain-specific emotion lexicon. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, 22–27 June 2014, Baltimore, MD, USA, vol. 2: Short Papers, pp. 421–426 (2014)
5. Li, S., Huang, L., Wang, R., Zhou, G.: Sentence-level emotion classification with label and context dependence. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, 26–31 July 2015, Beijing, China, vol. 1: Long Papers, pp. 1045–1053 (2015)

6. Abdul-Mageed, M., Ungar, L.: EmoNet: fine-grained emotion detection with gated recurrent neural networks. In: Meeting of the Association for Computational Linguistics, pp. 718–728 (2017)
7. Zhou, H., Chen, L., Shi, F., Huang, D.: Learning bilingual sentiment word embeddings for cross-language sentiment classification. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2015) (2015)
8. Ling, W., Xiang, G., Dyer, C., Black, A.W., Trancoso, I.: Microblogs as parallel corpora. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4–9 August 2013, Sofia, Bulgaria, vol. 1: Long Papers, pp. 176–186 (2013)
9. Wang, Z., Lee, S., Li, S., et al.: Emotion detection in code-switching texts via bilingual and sentimental information. In: Meeting of the Association for Computational Linguistics and the, International Joint Conference on Natural Language Processing, pp. 763–768 (2015)
10. Wang, Z., Yue, Z., et al.: A bilingual attention network for code-switched emotion prediction. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, pp. 1624–1634 (2016)
11. Zhang, L., Wang, S., Liu, B.: Deep learning for sentiment analysis: a survey. Wiley Interdiscip. Rev. Data Min. Knowl. Disc. (2018)
12. Kim, Y.: Convolutional neural networks for sentence classification. In: Empirical Methods in Natural Language Processing, pp. 1746–1751 (2014)
13. Lai, S., Xu, L., Liu, K., et al.: Recurrent convolutional neural networks for text classification. In: National Conference on Artificial Intelligence, pp. 2267–2273 (2015)
14. Yang, Z., Yang, D., Dyer, C., et al.: Hierarchical attention networks for document classification. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489 (2016)
15. Mikolov, T., Chen, K., Corrado, G., et al.: Efficient estimation of word representations in vector space. Comput. Sci. (2013)
16. Liu, B., Hu, M., Cheng, J., Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of the 14th International World Wide Web conference (WWW-2005), Chiba, Japan, pp. 10–14 (2005)
17. Graves, A.: Long short-term memory. In: Graves, A. (ed.) Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence, vol. 385. Springer, Heidelberg (1997)
18. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. Comput. Sci. (2014)