THU_NGN at IJCNLP-2017 Task 2: Dimensional Sentiment Analysis for Chinese Phrases with Deep LSTM

Chuhan Wu[1], **Fangzhao Wu**[2], **Yongfeng Huang**[1], **Sixing Wu**[1] and **Zhigang Yuan**[1]
[1]Tsinghua National Laboratory for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
{wuch15,wu-sx15,yuanzg14}@mails.tsinghua.edu.cn, yfhuang@mail.tsinghua.edu.cn
[2]Microsoft Research Asia, wufangzhao@gmail.com

Abstract

Predicting valence-arousal ratings for words and phrases is very useful for constructing affective resources for dimensional sentiment analysis. Since the existing valence-arousal resources of Chinese are mainly in word-level and there is a lack of phrase-level ones, the Dimensional Sentiment Analysis for Chinese Phrases (DSAP) task aims to predict the valence-arousal ratings for Chinese affective words and phrases automatically. In this task, we propose an approach using a densely connected LSTM network and word features to identify dimensional sentiment on valence and arousal for words and phrases jointly. We use word embedding as major feature and choose part of speech (POS) and word clusters as additional features to train the dense LSTM network. The evaluation results of our submissions (1st and 2nd in average performance) validate the effectiveness of our system to predict valence and arousal dimensions for Chinese words and phrases.

1 Introduction

Sentiment analysis is an important task in opinion mining for both academic and business use. Traditional sentiment analysis approaches mainly intend to identify the positive or negative sentiment polarities of text. This field has been widely researched and has many effective approaches based on rules or statistical methods. However, analyzing only the polarities of sentiments is rough and can't differ sentiment distinctions in fine-grained. In order to go further in fine-grained sentiment analysis, some approaches were proposed to address this problem in more categories or in real-value, such as dimensional sentiment analysis. Evaluating sentiment in valence-arousal (VA) space was first proposed by Ressel (1980). As shown in Figure 1, the valence dimension represents the degree of positive or negative sentiment, while the arousal dimension indicates the intensity of sentiment. Based on this two-dimensional representation, any affective state can be represented as a point in the VA coordinate plane by determining the degrees of valence and arousal of given words (Wei et al., 2011; Malandrakis et al., 2011; Yu et al., 2015; Wang et al., 2016) or texts(Kim et al., 2010; Paltoglou et al., 2013).

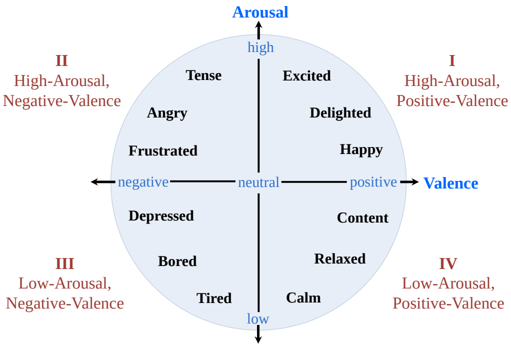

Figure 1: Two-dimensional valence-arousal space.

External VA resources like lexicons are necessary to VA sentiment evaluation. However, there is a lack of these resources especially for Chinese, and it's usually difficult to construct them manually. Thus in order to get large scale lexicons in a reasonable cost, the objective of the shared task DSAP is to automatically acquire the valence-arousal ratings of Chinese affective words and phrases. Some typical approaches to word-level VA rating task are based on statistical observations like linear regression (Wei et al., 2011) and kernel function (Malandrakis et al., 2011). However, these methods deeply rely on the affective lex-

Proceedings of the 8th International Joint Conference on Natural Language Processing, Shared Tasks, pages 47–52,
Taipei, Taiwan, November 27 – December 1, 2017. ©2017 AFNLP

icons and may ignore some high level sentiment features. After an effective word representation proposed by Mikolov et al. (2013), some methods based on word embedding were introduced to this task such as weighted graph (Yu et al., 2015; Wang et al., 2016). And in recent years, NN-based methods (Chou et al., 2016; Du and Zhang, 2016) were employed for this task and show better performance.

However, deep learning methods haven't been applied to such word-level and phrase-level task yet. Motivated by the successful applications of deep learning such as DenseNet proposed by Huang et al. (2016), we propose a densely connected deep LSTM network to predict VA ratings for words and phrases jointly. We segment all words and phrases and pad them to the same length for joint training. In network training, we use word embedding as the representation of word and add POS embedding and word clusters as additional features. The evaluation results of our system (1st and 2nd of two runs) in this task show the effectiveness of our method.

2 Densely Connected LSTM with Word Features

2.1 Network Architecture

Due to the feature self-extraction ability of deep network, features in different level can be learned by different layers. If we concatenate these features, layers can learn linguistic features of different levels at the same time. Since the input is sequential data, the layers of network can be implemented with LSTM. The architecture of our dense LSTM network is shown in Figure 2. Output from every top LSTM layer will be concatenated together as the input for bottom layers. Thus for a N-layer dense LSTM, there will be $\frac{N(N-1)}{2}$ connections.

The input of our network is word embedding concatenated with additional features. The details of features will presents in the following subsections. In this dense network, we pad all N LSTM layers to the same length L. We mark the output hidden state of i-th LSTM layer as h_i. Thus, the fitting function of h_i can be represented as follows:

$$h_i = \mathcal{F}_i(h_1; h_2; ...; h_{(i-1)}) \tag{1}$$

where $\mathcal{F}_i$ denotes the fitting function of i-th LSTM layer. Finally, a linear decoder is used to dense the output from the bottom LSTM layer into VA ratings. So the final output y is:

$$y = W \cdot h_{NL} + b \tag{2}$$

Where h_{NL} denotes the last hidden state of the N-th layer. From Eq. 1 we can see that each layer can learn all levels of features from previous layers at the same time. Thus it may be easier for the network to learn a better representation by combining low-level semantic and high-level sentiment information.

2.2 Word Features

In our method, word embedding is the major feature for network's training, while POS embedding and the one-hot representation of word cluster are chosen as additional linguistic features. We concatenate these features of each word to feed the network. These features are described as follows.

2.2.1 Word Embedding

In our model, we embed each word into a v_1-dim vector. The word embedding model is trained on a mixed corpus including SogouCA News dump[1] and wiki dump[2]. Since the linguistic features for out-vocabulary word are missing, and the miss segments can also cause similar problem. So we use 500 collected sentences from the Internet to fix this problem in run2 submission. The embedding we use is Google Embedding (Mikolov et al., 2013) proposed by Mikolov et al. We use the open source word2vec tool[3] to train word embedding and get word clusters.

2.2.2 POS Embedding

We get the POS tags of words and phrases after parsing. Since the POS tags of words also carry rich linguistic information, we embed POS tags into a v_2-dim vector when training on the dataset instead of using one-hot representation.

2.2.3 Word Cluster

After getting the embedding of words, we cluster all words in the dictionary into k classes by K-means method. The selection of k is based on the 10-fold cross validation results in our experiment. The class of a word will be one-hot encoded and then merged directly with other features.

[1] https://www.sogou.com/labs/resource/ca.php

[2] http://download.wikipedia.com/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2

[3] https://code.google.com/archive/p/word2vec/

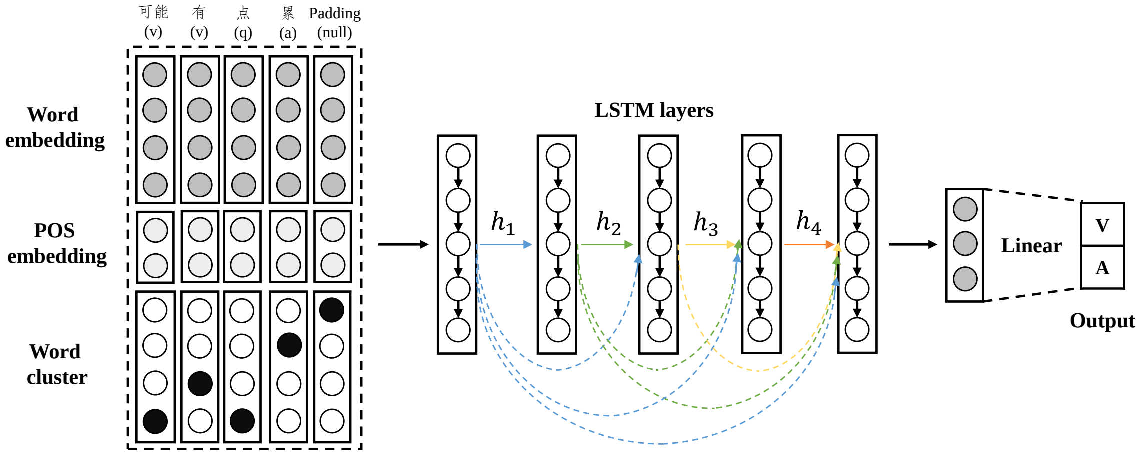

Figure 2: Dense LSTM network architecture. This figure shows a 5 layers dense LSTM as an example. The dashed lines represent the highway connection of different layers.

3 Experiment

3.1 Experiment Settings

3.1.1 Dataset and Metrics

In this task we use the dataset provided by the organizer which contains 2,802 words and 2,250 phrases for training, 750 words and 750 phrases for test. This dataset is based on the Chinese valence-arousal words (CVAW) dataset with 1,654 words and its later extensions. The annotations of valence and arousal are real-value from 1 to 9.

Prediction performance is evaluated by examining the difference between machine-predicted ratings and human annotated ratings. The evaluation metrics include Mean Absolute Error (MAE) and Pearson Correlation Coefficient (PCC), as shown in the following equations.

- Mean absolute error(MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|A_i - P_i|$$

- Pearson correlation coefficient(PCC)

$$PCC = \frac{1}{n-1}\sum_{i=1}^{n}(\frac{A_i - \bar{A}}{\sigma_A})(\frac{P_i - \bar{P}}{\sigma_P})$$

where A_i is the actual value, P_i is the predicted value, n is the number of test samples, $\bar{A}$ and $\bar{P}$ respectively denote the arithmetic mean of A and P, and σ is the standard deviation.

3.1.2 Preprocessing

Since we don't have enough corpus of traditional Chinese, we first translate the data into simplified Chinese. We use the ANSJ tool[4] to segment all words and phrases, because some words can also be splited into smaller subwords. Finally we pad all of them to the length of 5 for network training.

3.1.3 Network Training

In our experiment, the word embedding dimension v_1 is set to 300 and the POS embedding dimension v_2 is set to 50. The hidden states of every LSTM layer are 100-dim. The word cluster classes k is set to 250 for word-level prediction and 350 for phrase-level prediction. The objective function in our experiment is MAE and we use RMSProp optimizer to train the network. In order to prevent overfitting, we apply dropout after embedding and all LSTM layers. And specially in our run2 submission, we use a randomly selected dropout rate in every model, and the training samples are randomly selected from the whole training set. In order to suppress data noise and the randomicity introduced by dropout, we train our model for 100 times and ensemble all the model outputs by mean as the final predictions.

3.2 Performance Evaluation

The evaluation results of our system are shown in Table 1. Our approach shows effectiveness in both word and phrase level and significantly outperforms these baselines. Besides, in the averaging performance, we reach 0.427 and 0.6245 of

[4] https://github.com/NLPchina/ansj_seg.git

MAE, 0.9345 and 0.7985 of PCC in the valence and arousal dimension respectively. And especially in phrase-level, our approach reach very low MAE (0.345 and 0.385) and very high PCC (0.961 and 0.911). We can see that our approach works better in phrase-level, which may indicate that our recurrent-NN based method makes better use of sequential information. And the word-level results in run1 are much lower than our cross validation results. It may due to the out-dictionary words or network overfitting. To solve these problems in run2, we use our collected sentences from the Internet to re-train the embedding, and ensemble models that trained by randomly selected data and dropout rate to reduce the risk of overfitting. The final evaluation results show the significant improvement made by these processes.

Model	Valence		Arousal	
	MAE	PCC	MAE	PCC
Word				
Baseline	0.984	0.643	1.031	0.456
MLP	0.728	0.802	0.955	0.577
CNN	0.765	0.772	0.992	0.537
LSTM	0.707	0.804	1.055	0.588
Our Run1	0.610	0.857	0.940	0.623
Our Run2	**0.509**	**0.908**	**0.864**	**0.686**
Phrase				
Baseline	1.051	0.610	0.607	0.730
MLP	0.831	0.763	0.449	0.872
CNN	0.512	0.911	0.471	0.861
LSTM	0.429	0.939	0.450	0.869
Our Run1	0.349	0.960	0.389	0.909
Our Run2	**0.345**	**0.961**	**0.385**	**0.911**

Table 1: Evaluation results of our two submissions and some baselines for comparison.

3.3 Influence of Network Depth

We compare the validation MAE performance of network with different depth, as shown in Figure 3. Note that we don't use word cluster here. When $N = 1$ this network is equal to a single LSTM layer, and when $N = 2$ it's equal to a 2-layer deep LSTM. From two figures, we can see that the dense LSTM network has better performance than a single LSTM or a standard deep LSTM, and a 5-layer dense LSTM network is the most suitable for this task. This result indicates that our dense LSTM network can learn a better representation of sentiment by combining different levels of features.

Model	**Valence**		**Arousal**	
	MAE	PCC	MAE	PCC
Word				
+POS	**0.516**	**0.898**	**0.719**	**0.748**
w/o POS	0.540	0.874	0.732	0.721
Phrase				
+POS	**0.335**	**0.968**	**0.387**	**0.918**
w/o POS	0.364	0.960	0.402	0.911

Table 2: Performance with POS embedding or not.

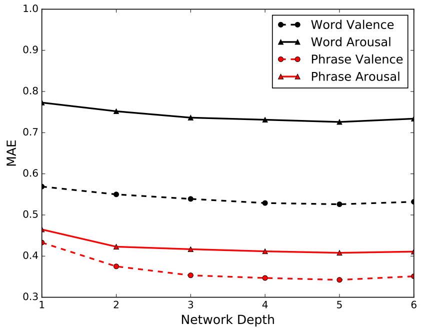

Figure 3: Validation MAE with different network depth.

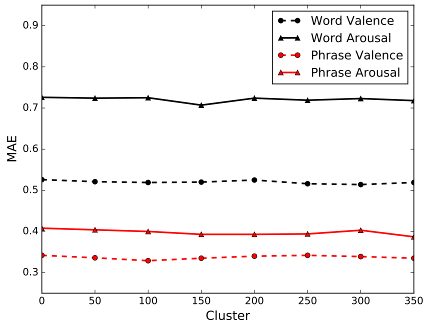

Figure 4: Validation MAE with cluster size k.

3.4 Influence of Word Features

3.4.1 POS embedding

We compare the validation performance of network with POS embedding and without. From Table 2, We can see that the significant improvement made by POS feature. This result indicates that POS tags of words contain very useful linguistic information and can improve the performance of deep model in DSA task.

3.4.2 Word cluster

We choose the cluster size k according to the validation performance and we use the network settings in run1. See Figure 4. The validation MAE results show that word cluster can improve the performance of network. We found the cluster size $k = 250$ is slightly better for word-level prediction while $k = 350$ is slightly better for phrase-level prediction.

3.5 Model Ensemble

We ensemble the output predictions of models trained by different hyper-parameters and training data. The results is shown in Table 3. We can see that the model ensembling can improve the performance very siginicicantly. This may because the ensembled model have a better generalization ablility and is more stable with the data noise.

Model	Valence		Arousal	
	MAE	PCC	MAE	PCC
Word				
+ensemble	**0.469**	**0.927**	**0.688**	**0.767**
w/o ensemble	0.516	0.898	0.719	0.748
Phrase				
+ensemble	**0.286**	**0.975**	**0.348**	**0.930**
w/o ensemble	0.335	0.968	0.387	0.918

Table 3: The influence of model ensembling.

4 Conclusion

In this paper, we introduce a novel approach using a densely connected LSTM network with word features to DSAP shared task for Chinese words and phrases. We combine deep network and word features including POS and word cluster to address this task. In addition, we also use a random model ensemble strategy to improve the performance of our approach. The evaluation results (1st and 2nd averaging performance in two runs) show the effectiveness of our approach.

Acknowledgments

This research is supported by the Key Program of National Natural Science Foundation of China under Grant No. U1405254 and U1536201.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

Margaret M Bradley and Peter J Lang. 1999. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Technical report C-1, the center for research in psychophysiology, University of Florida.

Wei-Chieh Chou, Chin-Kui Lin, Yih-Ru Wang, and Yuan-Fu Liao. 2016. Evaluation of weighted graph and neural network models on predicting the valence-arousal ratings of chinese words. In *Asian Language Processing (IALP), 2016 International Conference on*, pages 168–171. IEEE.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Steven Du and Xi Zhang. 2016. Aicyber's system for ialp 2016 shared task: Character-enhanced word vectors and boosted neural networks. In *Asian Language Processing (IALP), 2016 International Conference on*, pages 161–163. IEEE.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06*, pages 417–422.

Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. 2016. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*.

Sunghwan Mac Kim, Alessandro Valitutti, and Rafael A Calvo. 2010. Evaluation of unsupervised emotion models to textual affect recognition. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 62–70. Association for Computational Linguistics.

Baoli Li. 2016. Learning dimensional sentiment of traditional chinese words with word embedding and support vector regression. In *Asian Language Processing (IALP), 2016 International Conference on*, pages 324–327. IEEE.

Nikos Malandrakis, Alexandros Potamianos, Elias Iosif, and Shrikanth Narayanan. 2011. Kernel models for affective lexicon creation. In *Twelfth Annual Conference of the International Speech Communication Association*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Georgios Paltoglou, Mathias Theunis, Arvid Kappas, and Mike Thelwall. 2013. Predicting emotional responses to long informal text. *IEEE transactions on affective computing*, 4(1):106–115.

JA Ressel. 1980. A circumplex model of affect. *J. Personality and Social Psychology*, 39:1161–78.

Jin Wang, Liang-Chih Yu, K Robert Lai, and Xuejie Zhang. 2016. Community-based weighted graph model for valence-arousal prediction of affective words. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):1957–1968.

Wen-Li Wei, Chung-Hsien Wu, and Jen-Chun Lin. 2011. A regression approach to affective rating of chinese words from anew. *Affective Computing and Intelligent Interaction*, pages 121–131.

Liang-Chih Yu, Lung-Hao Lee, Shuai Hao, Jin Wang, Yunchao He, Jun Hu, K Robert Lai, and Xue-jie Zhang. 2016. Building chinese affective resources in valence-arousal dimensions. In *HLT-NAACL*, pages 540–545.

Liang-Chih Yu, Jin Wang, K Robert Lai, and Xue-jie Zhang. 2015. Predicting valence-arousal ratings of words using a weighted graph method. In *ACL (2)*, pages 788–793.