MESSAGE POLARITY CLASSIFICATION USING ENSEMBLE RECURRENT NEURAL NETWORKS WITH WORD EMBEDDINGS

T.Jhansi Rani[1], K.Anuradha[2], P. Vijayapal Reddy[3]
[1] GITAM University Hyderabad, [2]GRIET Hyderabad, [3]Matrusri Engg. College Hyderabad

Abstract

Recently, twitter sentiment classification using neural networks has become one of state-of-the-arts, which requires less feature engineering work compared with traditional methods. This paper shows a simple and effective ensemble Recurrent Neural Network method to boost the performance of neural models which collect several words embedding learned on large-scale corpus. The different word embeddings cover different words and encode different semantic knowledge, thus using them together can improve the generalizations and performances of neural models. The additional comparisons demonstrate the superiority of ensemble neural network model over the other models based on word embedding set.

Index Terms: Recurrent Neural Networks, Sentiment Classification, Word Embedding.

I. INTRODUCTION

Twitter sentiment classification has attracted a lot of attention [1, 2, 3], which aims to classify a tweet into three sentiment categories: negative, neutral, and positive. Tweet text has several features such as written by the informal language, hash-tags and emoticons indicate sentiments, and sometimes is sarcasm, which make decisions of tweet sentiment hard for machines. With releases of annotated datasets, more researchers prefer to use the Twitter sentiment classification as one of the challenging to evaluate their proposed models.

Traditional methods [4] for twitter sentiment classification use a variety of hand-crafted features including surface-form, semantic and sentiment lexicons. The performances of these methods often depend on the quality of feature engineering work, and building a state-of-the-art system is difficult for novices. Moreover, these designed features are presented by the one-hot representation which cannot capture the semantic specific of different features. To address this, [5] induced sentiment-specific low-dimensional, real-valued embedding features for twitter classification, which encode both semantics and sentiments of words. In the experiments, the embedding features and hand-crafted features obtain similar results, and also they are complementary for each other in the system. With the developments of neural networks in natural language processing, neural sentiment classification [6,7] has attracted a lot of attention recently and become the state-of-the-arts. These methods learn word embeddings from large-scale twitter corpus, and later tune neural networks by the tweets which have distant labels, and finally fine-tune the proposed models by the annotated datasets.

It is popular to learn word embeddings using in-domain data is an effective way to boost model performances [8,9]. However, collecting large-scale twitter corpus is often time-consuming. In this paper, different word embedding sets to boost the performances of Recurrent neural networks, which only include released different word embeddings sets [8]. A simple and effective ensemble method is proposed, which takes different word embedding sets as input to train neural networks and predicts labels of testing tweets by merging all output of neural models. Ensemble RNN ensemble method show its effectiveness, though most of used word embedding sets are not learned from twitter corpus, which can be explained that different embedding sets has different vocabularies and encode different parts of sentiment knowledge. Additional experiments to analyze Ensemble RNN model.

II. MODEL

The details of Ensemble RNN method, which is illustrated in Figure 1. The different word embedding sets into neural net-works and train these neural networks separately. When predicting the labels of tweets in testing set, sum label probabilities of all neural network to make final decisions

A. Neural Network

There are many neural networks (e.g., LSTM, RNN and GRU) for Ensemble RNN method, in which RNN [10] is used in Ensemble RNN method. RNN adaptive gated decay, which aims to capture longer-range, non-consecutive patterns in a weighted manner.

Given a sequence of words which are denoted as $\{x_i\}^{l}_{i=1}$, the corresponding word embeddings $\{x_i\}^{l}_{i=1}$ are derived using the embedding matrix E. Then, RNN obtains their corresponding hidden vectors $\{h_i\}^{l}_{i=1}$ using the convolution operation and gating mechanism. After obtaining hidden vectors, RNN uses a pooling operation to get fixed-sized vector presentation, which is fed into softmax layer to finish the prediction. The n-gram convolution operation and gating decay are described as follows:

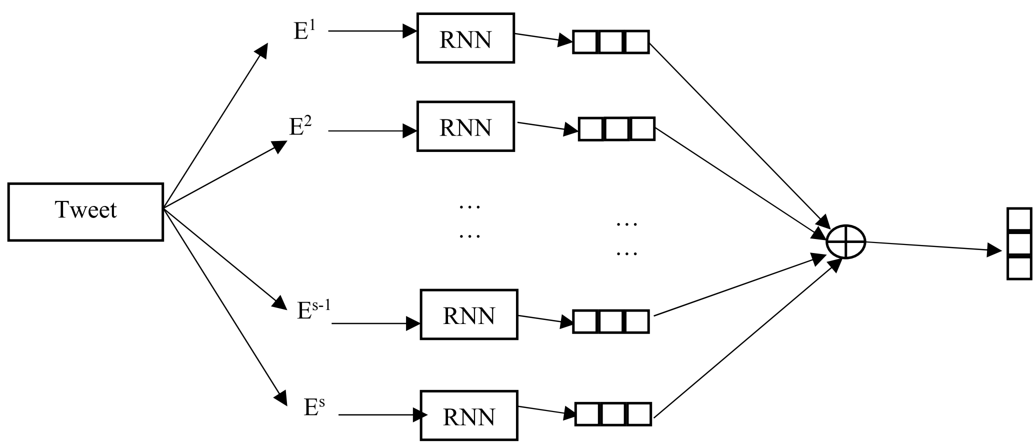

Fig:1 Recurrent Neural Networks with Word Embeddings

Sets	Corpus	Scale	Algorithm	Dimension	Vocab
gloveT	Twitter	27B	GloVec	200	1.2M
word2vecGN	Google News	100B	Word2Vec	300	3.0M

Table 1: Statistics of the embedding sets. Scale means the size of tokens in corpus, M and B refer to million and billion respectively. The embedding set word2vecGN are trained by Word2Vec.

$$\lambda_t = \sigma\left(W^{\lambda} x_t + U^{\lambda} h_{t-1} + b^{\lambda}\right)$$

$$c_t = \lambda_t \odot c_{t-1} + \left(1 - \lambda_t\right) \odot \left(W_1 x_t\right),$$

$$c^{(2)}{}_t = \lambda_t \odot c^{(2)}{}_{t-1} + \left(1 - \lambda_t\right) \odot \left(c^{(1)}{}_{t-1} + W_2 x_t\right)$$

$$\ldots\ldots,$$

$$c^{(n)}_t = \lambda_t \odot c^{(n)}_{t-1} + \left(1 - \lambda_t\right) \odot \left(c^{(n-1)}_{t-1} + W_n x_t\right)$$

$$h_t = sigm(c^{(n)}_t + b)$$

where W^{λ}, U^{λ}, b^{λ}, b and W_* are learnable parameters, σ is sigmoid function which rescales the value into (0, 1), is dot product, λ_t is gating value determining how much information of x_t and previous patterns is added into the hidden vector, $c_t^{(i)}$ refer to the vector for accumulated previous patterns which are ended with x_t include i consecutive tokens. When $\lambda_t = 0$, the convolution becomes a standard n-gram convolution.

The deep RNN can be built by adding several convolution layer on top of hidden vectors derived from the bottom convolution layer. Consider the RNN with d convolution layers, which outputs $\{h^d_i\}^l_{i=1}$.

Then, a last pooling operation is conducted on hidden vectors to obtain text representation r. Finally, text representation is fed into a softmax layer. The softmax layer outputs the probability distribution over |Y| categories for the distributed representation, which is defined as $p(r) = \mathrm{softmax}(W^{class}{}_k r)$.

The cross-entropy objective function is used to optimize the RNN model.

B.Prediction

Consider RNN models with different embedding sets as input. where s embedding sets which are denoted as $\{E^1, E^2, \cdots, E^s\}$, and feed them into s RNN models, then learn RNN models separately. Then predict sentiment label of testing tweet based on these learned RNN models, which are described by following functions:

$$p_1 = RNN_1(\{x_i\}^l_{i=1}, E^1),$$
$$p_2 = RNN_2(\{x_i\}^l_{i=1}, E^2),$$
$$\ldots,$$
$$p_s = RNN_s(\{x_i\}^l_{i=1}, E^s),$$
$$p' = \sum_{1 \le i \le s} p_i.$$
$$y = \mathrm{argmax}_{1 \le i \le |Y|} p'_i,$$

where y is the predicted label.

III. EXPERIMENT

A. Datasets and Settings

In this paper results are obtained by 2 embedding sets which are described in Table 1. Hence crawl and merge all annotated datasets of previous SemEvals, and split them into training, development, and testing sets with ratio 8:1:1, which are shown in Table 2 together with testing set of SemEval 2017. From the table, the testing set of SemEval 2017 has big differences on the category ratio (negative: neutral: positive), compared with the previous SemEval datasets.

Dataset	Number of Tweets	Ratio of positive/negative/neutral Tweets
Previous SemEvals	50032	1.5/4.7/3.8
SemEval 2017 Test	12284	3.2/4.8/2.0

Table 2: Datasets statistics

For the model settings, all RNN models have same configurations but different word embedding sets. Then set dimensions of hidden vectors to 250 and depths d to 2. To avoid model over fitting, the dropout and regularization as follows: (1) the regularization parameter is set to 1e-5; (2) the Dropout rate is set to 0.3, which is applied in the final text representation. All parameters are learned by Adam optimizer [12] with the learning rate 0.001. Note that, all word embedding sets are fixed when training. All models are tuned by the development set in Training.

B. Result Analysis

The results on datasets are observed from previous SemEvals, which are described in Table 3. Then report the performances of Ensemble RNN method in Table 4.

Embeddings	Accuracy	Recall-Negative	Recall-neutral	Recall-positive	Recall-average
gloveT	65.6	62.8	60.2	72.8	65.7
Word2vec	65.5	68.2	65.3	70.5	65.7
Ensemble	69.6	69.1	68.2	75.0	73.5

Table 3: Results on datasets previous SemEval data sets

Embedding	Accuracy	Negative	neutral	Positive	Avg
GloveT	62.8	69.5	56.7	67.2	64.3
Word2vec	61.9	67.4	59.0	59.8	62.1
Ensemble RNN model	65.7	68.7	2.4	67.8	66.6

Table 4: Results for message polarity classification with five scale point

From the Table 3, the gloveT performs worst though it is trained on in-domain twitter dataset. Then include that the quality of corpus is also important as the size of corpus and domain in twitter sentiment classification. Additionally, obtain infer that word2vec of Google news outperforms others in recall of negative category, and gloveT is best in recall of positive category. Different embedding sets propose different characteristics. Additionally, the ensemble method obtains a significant improvement of 4%.

In the Table 4, Ensemble RNN method with best in SemEval 2017, and report the result of individual embedding sets. This method outperforms other systems in accuracy, but performs worse in Recall-Average, especially in Recall-Negative. Compared with the median system, method has improvements of about 5% in both accuracy and Recall-Average. Different from the results in Table 3, the word2vec performs worse among these embedding sets, while the gloveT obtains best performances. Additionally, that gloveT performs best both in Recall-Negative and Recall-Positive, and word2vec performs best in Recall-Neutral. Compared with the embedding baselines, ensemble RNN method obtains improvements of 2.7% and 1.5% in accuracy and R Average respectively, which demonstrates the effectiveness of the proposed method.

IV. CONCLUSION

Proposed an effective ensemble method to boost the neural twitter sentiment classification. By using different embedding sets, the system can cover more words and encode more sentiment information. The results on datasets of previous SemEval and show the effectiveness of Ensemble RNN method. Moreover, error analysis required to conduct to propose the main challenges for ensemble RNN method.

REFERENCES

[1] Li Dong, Furu Wei, Yichun Yin, Ming Zhou, and Ke Xu. 2015. Splusplus: A feature-rich two-stage classifier for sentiment analysis of tweets. SemEval-2015 page 515.

[2] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. In Pro-ceedings of the 10th International Workshop on Se-mantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016. pages 1–18.

[3] Sara Rosenthal, Na Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twit-ter. In Proceedings of the 11th International Work-shop on Semantic Evaluation. Association for Computational Linguistics, Vancouver, Canada, SemEval ’17.

[4] Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrccanada: Building the state-of-the-art in sentiment analysis of tweets. In Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013, Atlanta, Georgia, USA, June 14-15, 2013. pages 321–327.

[5] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Baltimore, Maryland, pages 1555–1565. http://www.aclweb.org/anthology/P14-1146 .

[6] Aliaksei Severyn and Alessandro Moschitti. 2015. UNITN: training deep convolutional neural network for twitter sentiment classification. In Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACLHLT 2015, Denver, Colorado, USA, June 4-5, 2015. Pages 464–469.

[7] Jan Deriu, Maurice Gonzenbach, Fatih Uzdilli, Aurelien´ Lucchi, Valeria De Luca, and Mar-tin Jaggi. 2016. Swisscheese at semeval-2016 task 4: Sentiment classification using an en-semble of convolutional neural networks with distant supervision. In Proceedings of the 10th International Workshop on Semantic Evalu-ation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016. pages 1124–1128.

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their com-positionality. In

Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.

[9] Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016. pages 2979–2985.

[10]Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi S. Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Llu´ıs Marquez`. 2016. Semi-supervised question retrieval with gated convolutions. In NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016. pages 1279–1289.

[11]Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29

[12]Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. CoRR abs/1412.6980. http://arxiv.org/abs/1412.6980.2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. pages 1532–1543.