EICA at SemEval-2017 Task 4: A Convolutional Neural Network for Topic-based Sentiment Classification

Maoquan Wang, Shiyun Chen, Yufei Xie, Jing Ma, Zhao Lu
Department of Computer Science and Technology,
East China Normal University, Shanghai, P.R.China
{maoquanwang,yufeixie,liangma,lzhao}@ica.stc.sh.cn, csyecnu@outlook.com

Abstract

This paper describes our approach for SemEval-2017 Task 4 - Sentiment Analysis in Twitter (SAT). Its five subtasks are divided into two categories: (1) sentiment classification, i.e., predicting topic-based tweet sentiment polarity, and (2) sentiment quantification, that is, estimating the sentiment distributions of a set of given tweets. We build a convolutional sentence classification system for the task of SAT. Official results show that the experimental results of our system are comparative.

1 Introduction

With the rapid growth of social media such as Twitter, sentiment classification towards the user generated texts has attracted increasing research interest. The objective of sentiment classification is identifying the sentiment of a text into binary polarity (Positive vs. Negative) or single-label multi-class (e.g., Very positive, Positive, Neutral, Negative, Very negative). Feature representation is one of key points for this kind of classification, which generally falls into two categories: (1) traditional feature engineering (Liu, 2012; Mohammad et al., 2013), such as sentiment lexicon, n-grams, dependency triple, etc. (2) deep learning methods (Zhao et al., 2015; Yang et al., 2016), which use exquisitely designed neural network to encode input texts and to get text feature representation. Recently, deep learning approaches emerge as powerful computational models for text sentiment classification, and have achieved new state-of-the-art result in some datasets.

SemEval-2017 provides a universal platform for researchers to explore the task of twitter sentiment analysis. In this paper, we explore Task 4 (Rosenthal et al., 2017), which includes five subtasks: subtask A, B and C are related to the task of sentiment classification, and subtask D and E are related to sentiment quantification (that is distributions of sentiments). Considering the length limitations of tweets, we view the subtasks of SAT as sentence-level sentiment analysis. We design a convolutional neural network for topic-based sentiment classification.

2 System Description

In this section, we describe the neural network architecture of our system. As shown in Figure 1, our system consists of six layers, an input layer, a convolutional layer, a max-pooling layer, a topic embedding layer, a concatenate layer, and an output layer.

Input layer. A tweet text can be denoted as a sentence sequence $\mathbf{x}$ with n words, $\mathbf{x} = [w_1, w_2, \cdots, w_i, \cdots, w_n]$. To obtain word vector of word w_i , we look-up word embedding matrix $\mathbf{E}$, where $e(w_i) \in \mathbf{R}^d$, $\mathbf{E} \in \mathbf{R}^{|V| \times d}$, $|V|$ is the vocabulary size. Then, we get an input matrix $\mathbf{X} = [e(w_1); \cdots; e(w_n)]$, where $\mathbf{X} \in \mathbf{R}^{n \times d}$.

Convolution layer. The convolution action has been used to capture n-gram information (Collobert et al., 2011), and n-gram has been shown useful for twitter sentiment analysis (Dos Santos and Gatti, 2014). In this layer, a set of m filters is applied to a sliding window of length h over each tweet matrix $\mathbf{X}$, and a feature $\mathbf{c_i} \in \mathbf{R}^{n-h+1}$ is generated from a window of words $e(w)_{i:i+h}$ by:

$$\mathbf{c_i} = f(F_k \cdot e(w)_{i:i+h} + b) \quad (1)$$

where f is an activation function, and $b \in \mathbf{R}$ is a bias term. The vectors $\mathbf{c} = [\mathbf{c}_1 \oplus \cdots \oplus \mathbf{c}_m]$ are then aggregated over all m filters into a feature map matrix. We consider m is 3, and h is chosen in $\{3, 4, 5\}$.

Max-pooling layer. In order to get a fixed dimension vector, we exploit pooling techniques to

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 737–740,
Vancouver, Canada, August 3 - 4, 2017.

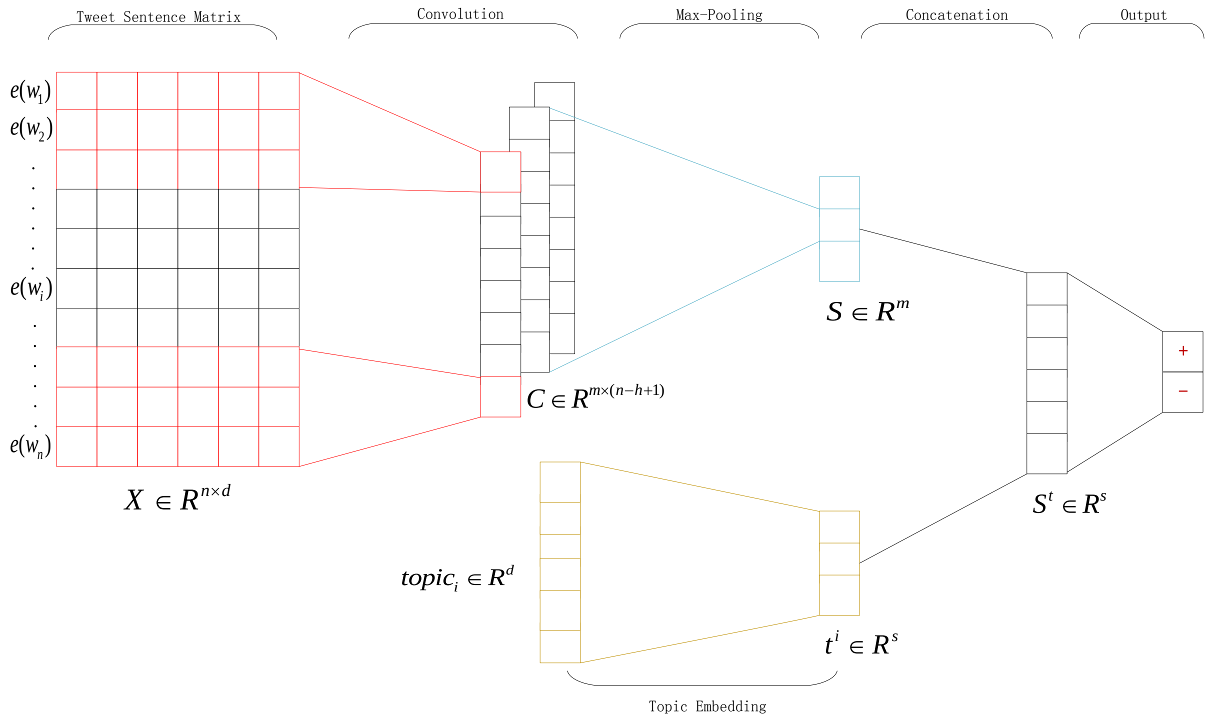

Figure 1: The framework of the simple CNN for topic-based sentiment classification.

get sentence representation $\mathbf{S} \in \mathbf{R}^m$, and we adopt max pooling function.

Topic embedding layer. To make the best use of topic information, we propose to learn an embedding vector $\mathbf{t}_i$ for each topic:

$$\mathbf{t}_i = tanh(W^{(1)} avg(e(\mathbf{w}_1), \cdots, e(\mathbf{w}_k))) \quad (2)$$

where $w_1, \cdots, w_k$ are topic words, $\mathbf{t_i} \in \mathbf{R}^s$, $avg(\cdot)$ is a element average function, and $W^{(1)} \in \mathbf{R}^{s\times d}$.

Concatenation layer. We use a concatenation layer to get tweet representation which can be formed as:

$$\mathbf{S^t} = tanh(W^{(2)} [\mathbf{S} \oplus \mathbf{t_i}]) \quad (3)$$

where $\oplus$ is the concatenation operator, $W^{(2)} \in \mathbf{R}^{s\times(s+m)}$.

Output layer. Finally, we use a softmax layer to get the class probability:

$$P_i = \frac{exp(W_{y_i}^T \mathbf{S^t}^{(i)} + b_{y_i})}{\sum_{j=1}^{C} exp(W_j^T \mathbf{S^t}^{(i)} + b_j)}. \quad (4)$$

Where $\mathbf{S^t}^{(\mathbf{i})}$ denotes the tweet representation with sentiment class y_i. W_j is jth column of parameter $W \in \mathbf{R}^{2s\times C}$ and C is number of categories.

Training process. The training goal is to minimize the cross-entropy loss over the training set T:

$$L(\theta) = -\sum_{x\in T}\sum_{i=1}^{C} P_i^g(x) \cdot logP_i(x) + \frac{\lambda}{2} \parallel \theta \parallel^2 \quad (5)$$

where C is the number of classes, x represents a tweet, θ is the model parameters, $P^g(x)$ is the goal probability, which has the same dimension as the number of classes, and only the corresponding goal dimension is 1, with all others being 0.

We use mini-batch gradient descent algorithm to train the network, with the batch size is 32 and a learning rage of 0.01. We also use Adadelta (Zeiler, 2012) to optimize the learning of θ, which is a effective method to train the neural networks. We initialize all the matrix and vector parameters with uniform samples in $\left(-\sqrt{6/(r+c)}, +\sqrt{6/(r+c)}\right)$ (Glorot and Bengio, 2010), where r is the rows numbers , and c is the column numbers.

Pre-training Word Embedding We adopted the word2vec tool[1] to obtain word embedding with

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

the dimensionality of 100, trained on 238M tweet from Sentiment140[2].

3 Experiments

3.1 Datasets

Since only tweet IDs are provided by organizers, Some tweets are no longer available on Twitter due to tweets miss or system errors. Subtask B and D share one dataset, while subtask C and E share the other dataset. An overview statistics of the data available for download are given in Tables 1, 2, and 3, respectively.

dataset		positive	neutral	negative	total
train	2013train	3,632	4,564	1,453	9,649
	2013test	1,473	1,513	559	3,545
	2015test	1,033	983	363	2,379
	2016train	3,078	2,036	861	5,975
	2016dev	842	765	390	1,997
	2016test	7,033	10,302	3,221	20,556
dev	2013dev	573	737	339	1,649
	2014test	982	669	202	1,853
	2015train	170	252	66	488
	2016devtest	994	681	323	1,998
test	2017test	2,375	5,937	3,972	12,284

Table 1: Statistics of datasets for subtask A, English. The data was divided into train, dev and test sets.

dataset		positive	negative	total	topics
train	2015train	144	56	200	44
	2016train	3,579	754	4,333	60
	2016dev	985	339	1,324	20
	2016test	8,202	2,333	10,535	100
dev	2015test	863	260	1,123	137
	2016devtest	1,417	264	1,417	20
test	2017test	2,458	3,695	6,153	125

Table 2: Statistics of datasets for subtask B and D, English. The data was divided into train, dev and test sets.

dataset		-2	-1	0	1	2	total	topics
train	2016train	87	665	1,651	3,139	433	5,975	60
	2016dev	43	296	675	930	53	1,997	20
	2016test	136	2,191	10,034	7,814	381	20,556	100
dev	2016devtest	31	232	582	1,005	148	1,998	20
test	2017test	177	3,505	6,149	2,323	130	12,284	125

Table 3: Statistics of datasets for subtask C and E, English. The data was divided into train, dev and test sets.

[2] http://help.sentiment140.com/for-students/

3.2 Tweet Preparation.

We preprocessed all of our datasets as follows:

- The tweet text was lowercased.
- URLs and mentioned usernames were substituted by replacement tokens $<$ **LINK** $>$ and $<$ **MENTION** $>$ respectively. We also map numbers to a generic **NUMBER** token.
- All words that appear less than 5 times in the training were removed.
- Recovered the elongated words to their original forms, e.g., "gooooooood " to "good".
- The NLTK[3] twitter tool was employed to tokenize tweets.

3.3 Result on Test Data

Subtask A. For this subtask, there is no topic information, so we removed the Concatenate and Topic Embedding parts in Figure 1. We report the result of our system in Table 4.

Metric	Our score	Best score	Rank
ρ	0.595	0.681	23/37
F_1^{PN}	0.599	0.677	24/37
Acc	0.555	0.651	24/37

Table 4: Our score and rank compared to the best team's result for Subtask A "Message Polarity Classification" , English.

As shown in Table 4, we obtained poor performance in Subtask A. In order to further analysis our system performance on three-point scale(positive, negative, neutral), we show the detail results in Table 5

Our system did not distinguish the positive and negative class, but it performed well in neutral class. The unbalanced train data distribution may influence our system: 49%(positive), 31%(neutral) , 20%(negative).

Subtask B and C. The results of our system for Subtasks B and C are reported in Table 6 and Table 7, individually. For these two subtasks, the organizers make available alternative metrics. We found that the choice of the scoring metric influences results considerably, for example, in Subtask C, our system ranked second by MAE^{μ} while ranked $8th$ in MAE^{M}.

[3] http://nltk.org/

Team		P	R	F1
EICA	+	0.5086	0.6371	0.5656
	-	0.6137	0.4907	0.5453
	=	0.6351	0.6561	0.6454
DataStories	+	0.6259	0.7023	0.6619
	-	0.5929	0.8291	0.6914
	=	0.7471	0.5115	0.6073
BB_twtr	+	0.6851	0.6522	0.6682
	-	0.5848	0.8776	0.7019
	=	0.7518	0.5144	0.6109

Table 5: More detail metric in task A. EICA is our team name, DataStories and BB_twtr are rank 1 teams which have same ρ score. +: positive. -: negative. =: neutral

Metric	Our score	Best score	Rank
ρ	0.790	0.882	14/23
F_1^{PN}	0.775	0.890	14/23
Acc	0.777	0.897	16/23

Table 6: Our score and rank compared to the best team's result for Subtask B "Tweet classification according to a two-point scale" , English.

4 Conclusion

In this paper, we used a simple convolution neural network to accomplish sentiment analysis towards sentence level (i.e., subtask A) and topic level (i.e., subtask B, C), without using any user information. In future work, we will focus on developing advanced neural network to model sentence with the aid of user information. we also would like to ensemble deep leaning based classifier with handcrafted features based classifier to improve the system performance, in the next SemEval competition.

Acknowledgements

This research was supported in part by Science and Technology Commission of Shanghai Municipality (No.16511102702).

References

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

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

Metric	Our score	Best score	Rank
MAE^M	0.823	0.481	8/15
MAE^μ	0.509	0.554	2/15

Table 7: Our score and rank compared to the best team's result for Subtask C "Tweet classification according to a five-point scale" , English

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*. volume 9, pages 249–256.

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

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242* .

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Vancouver, Canada, SemEval '17.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*. pages 1480–1489.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. *arXiv preprint arXiv:1504.05070* .