deepSA2018 at SemEval-2018 Task 1: Multi-task Learning of Different Label for Affect in Tweets

[1]Zi-Yuan Gao and [2]Chia-Ping Chen
Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung, Taiwan
[1]m053040030@student.nsysu.edu.tw
[2]cpchen@cse.nsysu.edu.tw

Abstract

This paper describes our system implementation for subtask V-oc of SemEval-2018 Task 1: affect in tweets. We use multi-task learning method to learn shared representation, then learn the features for each task. There are five classification models in the proposed multi-task learning approach. These classification models are trained sequentially to learn different features for different classification tasks. In addition to the data released for SemEval-2018, we use datasets from previous SemEvals during system construction. Our Pearson correlation score is 0.638 on the official SemEval-2018 Task 1 test set.

1 Introduction

In recent years, people began to study how to create computational systems that process and understand the human languages. Today, people share their thoughts on social networks of the Internet, e.g. Facebook, Line, Twitter and so on. Thus, if the messages in the textual contents of social networks can be extracted and summarized automatically via algorithms, it is possible to learn what people are interested in or are concerned with, and use such information to predict future market trends.

Here we continue our previous works on the task 4 of SemEval-2017: Sentiment Analysis in Twitter (Rosenthal et al., 2017). SemEval-2017 subtask 4A is similar to task 1 of SemEval-2018: Affect in Tweets (Mohammad et al., 2018). They are challenging tasks as the messages on Twitter, called tweets, are short and informal. Furthermore, in addition to noisy or incomplete texts, the emotional content of a tweet can be ambiguous and subjective.

Affect in Tweets is an expanded version of WASSA-2017 shared task (Mohammad and Bravo-Marquez, 2017). The best system in WASSA-2017 is an ensemble of three sets of approaches, including feed-forward neural network, multi-task deep learning and sequence modeling using CNNs and LSTMs (Goel et al., 2017). They attempt to use the idea of multi-task learning to explore the notion of generalized or shared learning across different emotions. In this paper, we extend the idea with different label methods.

The rest of this paper is organized as follows. In Section 2, we introduce our system. In Section 3, we describe the details of training and experimental settings. In Section 4, we present the evaluation results along with our comments.

2 System Description

2.1 Baseline System

Using RNN has become a very common technique for various NLP tasks. There are many units for RNN-based model like simple RNN, gated recurrent units (GRU) (Chung et al., 2014), and long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997). For the baseline, we use LSTM as unit for its long-range dependency.

Figure 1 shows the architecture of our baseline system. Our baseline system contains an input layer, an embedding layer, Bi-LSTM layers and an output layer. At the input layer, the words of tweet are pre-processed, and they are treated as a sequence of words $w_1, w_2, ...w_n$. Each word is represented by a one-hot vector, and the size of input layer is equal to the size of word list.

At the embedding layer, each word is converted to a word vector. We use pre-trained word vector which are stored in a matrix. Words are mapped to word vectors by the word embedding matrix. A word not in the word embedding matrix is represented by a zero vector.

A Bi-LSTM layer contains h units. We use bidirectional (Schuster and Paliwal, 1997) structure

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 226–230
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

to gather two-way contextual information at each point. The hidden states from the first word to the penultimate word in a tweet are connected to the hidden states of the next word. The state values in both directions are combined with sum. Only the last Bi-LSTM states of the last word are connected to the output layer. Finally, the network output is converted to probability by a soft-max function.

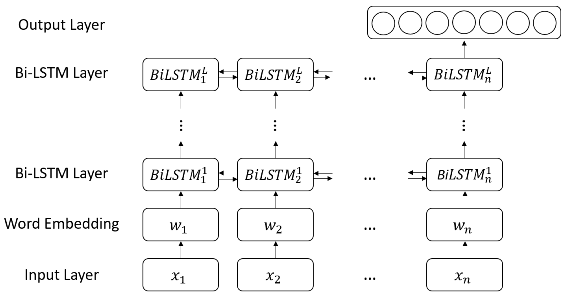

Figure 1: LSTM-RNN architecture.

2.2 Multi-task Learning

Multi-task learning has been used with success in applications of machine learning, from natural language processing (Collobert and Weston, 2008) and speech recognition (Deng et al., 2013). By sharing representations with related tasks, a model tends to generalize better on the original task (Ruder, 2017). In this work, different labels for the same data are exploited in multi-task learning.

Figure 2 shows our multi-task learning framework. The overall system is divided into five models. The Three-class model is trained first, and its trained parameters are used to initialize the parameters in other models. Then we train the Negative, Neutral, Positive class models, and their trained parameters are used to initialize the parameters of the Seven class model. The final output is obtained from the Seven class model.

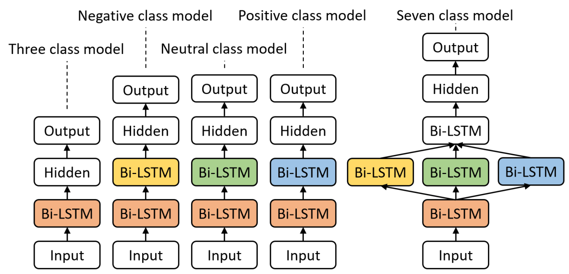

Figure 2: Multi-task learning of sentiment classification.

Three class model In Three class model, the tweets are converted to the word vector and used as the input to Bi-LSTM layer. The output layer has three units for three classes $\{-1, 0, 1\}$.

Negative class model The Negative class model has one more Bi-LSTM layer than Three class model. The output layer has four units for four classes $\{-3, -2, -1, other\}$.

Neutral class model The Neutral class model has the same architecture as the Negative class model. The output layer has two units for two classes $\{0, other\}$.

Positive class model The Positive class model has the same architecture as the Negative class model. The output layer has four units for four classes $\{other, 1, 2, 3\}$.

Seven class model The Seven class model combines the Bi-LSTM layers of the Negative class, Neutral class, and Positive class models. Further, it has one additional Bi-LSTM layer. The output layer has seven units for seven classes $\{-3, -2, -1, 0, 1, 2, 3\}$. Note that attention mechanism (Luong et al., 2015; Wang et al., 2016) is incorporated in this model.

3 Training

3.1 Data

We use the dataset provided for the SemEval-2018 shared task (Mohammad et al., 2018), which includes a new dataset and the datasets provided for SemEval-2017 (Rosenthal et al., 2017). Table 1 summarizes the statistics of these datasets.

3.2 Different Labeling

The SemEval-2017 dataset consists of three-class data, which is different from the new SemEval-2018 dataset. In order to exploit SemEval-2017 dataset, we modify the data labels. In the baseline system, we change the label to ±1, ±2, or ±3. Adding a lot of data lead to imbalance problem, so we apply two methods of data balance. Method 1 is that adding data to positive and negative classes randomly such that they have same size respectively. Method 2 is that adding data to all classes randomly such that they have 3,000 tweets. Table 1 shows the numbers of data points after these different labeling methods.

dataset	labels	Negative			Neutral	Positive			total
		-3	-2	-1	0	1	2	3	
train-18	-	129	249	78	341	167	92	125	1,181
train-17	-	8,581			18,186	15,219			41,986
train-all	to ±1	129	249	8,659	18,527	15,386	92	125	43,167
train-all	to ±2	129	8,830	78	18,527	167	15,311	125	43,167
train-all	to ±3	8,710	249	78	18,527	167	92	15,344	43,167
train-all	bal-method 1	3,013	3,012	3,012	18,527	5,201	5,201	5,201	43,167
train-all	bal-method 2	3,000	3,000	3,000	3,000	3,000	3,000	3,000	21,000
dev-18	-	69	95	34	105	58	35	53	449

Table 1: Statistics of our different labeling methods and datasets. train-18 and dev-18 are from SemEval-2018 Task 1. train-17 is from SemEval-2017 task 4. train-all means the merger of the train-18 and train-17 datasets.

3.3 Pre-processing

We begin with basic pre-processing methods (Yang et al., 2017), e.g. splitting a tweet into word, replacing URLs and USERs with normalization patterns <URL> and <USER>, and converting uppercase letters to lowercase letters. As tweets are informal and complex, the basic pre-processing is too simple to convey enough important information.

Tweets often have emoticons and hashtags, which could be instrumental to sentiment analysis. Thus, we use text processing tool[1] (Baziotis et al., 2017) to improve text normalization, including sentiment-aware tokenization, spell correction, word normalization, word segmentation (for splitting hashtags). and word annotation.

3.4 Early Stopping

The early stopping method is used to prevent overfitting when the loss of a development set ceases to decrease for a few epochs. We randomly take 20% of SemEval-2018 train data as the development set for early stopping and the remaining 80% data as the train set.

3.5 Settings

The maximum length for any tweet in the used datasets is $n = 99$. The embedding is based on a publicly available set of word vectors learned from 400 million tweets for the ACL WNUT 2015 shared task (Baldwin et al., 2015).

The baseline system uses 4 hidden Bi-LSTM layers, with 300 neurons in each layer. Dropout method with probability 0.3 is used to prevent the model from overfitting (Srivastava et al., 2014).

In the multi-task learning approach, the numbers of neurons in the Bi-LSTM and hidden layers are $[200, 200]$, $[200, 150, 200]$, $[200, 150, 100]$, $[200, 150, 200]$, $[200, [150, 150, 150], 200, 200]$ for the 5 different class models, respectively.

4 Results

4.1 Baseline System

First, we compare the experiments of different labeling in baseline system to decide how to use the train-17 dataset. In baseline system, we use the basic pre-processing for text normalization. The results are shown in Table 2. The calculation of Pearson correlation coefficient (Pcc.) requires calculating the mean value of the data, which is often close to zero. From the results, labeling to more distant from zero get the higher Pcc. Therefore, we use labeling to ±3 method in the multi-task learning system.

train set	labels	Pcc.	Acc.
train-18	-	0.515	0.298
train-all	to ±1	0.572	0.261
train-all	to ±2	0.629	0.323
train-all	to ±3	**0.649**	**0.347**
train-all	bal-method 1	0.548	0.303
train-all	bal-method 2	0.553	**0.347**

Table 2: Results of different labeling. Pcc. means the pearson correlation coefficient (all classes). Acc. means the accuracy.

4.2 Multi-task Learning System

Table 3 shows the results of multi-task learning. With basic pre-processing for text normalization, the multi-task learning system is better than the

[1] github.com/cbaziotis/ekphrasis

model	training set	Pcc.	Pcc.(s-m)	Kappa	Kappa(s-m)	Acc.
baseline	train-18	0.515	0.567	0.499	0.534	0.298
baseline	train-all	0.649	0.712	0.628	0.700	0.347
multi-task	train-18	0.603	0.660	0.579	0.623	0.312
multi-task	train-all	0.689	0.760	0.671	0.753	0.350
multi-task*	train-18	0.622	0.667	0.616	0.653	**0.361**
multi-task*	train-all	**0.691**	0.770	0.665	0.757	0.323
multi-task*	train-all	**0.638**	0.698	0.606	0.643	-

Table 3: Results of multi-task learning. Final row is the official SemEval-2018 test set result and others are development set results. Here * means using the ekphrasis tool for pre-processing and s-m means some-emotion.

baseline system. When the basic pre-processing method is replaced by using ekphrasis tool, the performance is further improved. Finally, we submit the results from our best system for the unseen test set to SemEval-2018, getting 0.638 for Pcc. eventually. We note this is significantly lower than 0.691 on the development data.

5 Conclusion

The proposed method improves performance on SemEval-2018 over baseline systems without multi-task learning. External dataset can significantly improve the Pcc. performance, but not the Acc. performance. The possible reason is that all the labels of external dataset are marked as ±3, resulting in data imbalance problem. In the future, we will use skewness-robust weights to solve this problem and use more resources to improve the system as sentiment lexicons.

References

Timothy Baldwin, Marie-Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754. Association for Computational Linguistics.

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*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Li Deng, Geoffrey Hinton, and Brian Kingsbury. 2013. New types of deep neural network learning for speech recognition and related applications: An overview. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8599–8603. IEEE.

Pranav Goel, Devang Kulshreshtha, Prayas Jain, and Kaushal Kumar Shukla. 2017. Prayas at emoint 2017: An ensemble of deep neural architectures for emotion intensity prediction in tweets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 58–65.

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

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Saif M. Mohammad and Felipe Bravo-Marquez. 2017. WASSA-2017 shared task on emotion intensity. In *Proceedings of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, Copenhagen, Denmark.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

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*, SemEval '17, Vancouver, Canada. Association for Computational Linguistics.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Po-Yuan Shih. 2016. *Skewness-Robust Neural Networks with Application to Speech Emotion Recognition*. Ph.D. thesis, Masters thesis, National Sun Yat-sen University.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615.

Tzu-Hsuan Yang, Tzu-Hsuan Tseng, and Chia-Ping Chen. 2017. deepsa at semeval-2017 task 4: Interpolated deep neural networks for sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 616–620. Association for Computational Linguistics.

Yichun Yin, Yangqiu Song, and Ming Zhang. 2017. Nnembs at semeval-2017 task 4: Neural twitter sentiment classification: a simple ensemble method with different embeddings. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 621–625.