EmoNLP at IEST 2018: An Ensemble of Deep Learning Models and Gradient Boosting Regression Tree for Implicit Emotion Prediction in Tweets

Man Liu

liumanlalala@gmail.com

Abstract

This paper describes our system submitted to IEST 2018, a shared task (Klinger et al., 2018) to predict the emotion types. Six emotion types are involved: anger, joy, fear, surprise, disgust and sad. We perform three different approaches: feed forward neural network (FFNN), convolutional BLSTM (ConBLSTM) and Gradient Boosting Regression Tree Method (GBM). Word embeddings used in convolutional BLSTM are pre-trained on 470 million tweets which are filtered using the emotional words and emojis. In addition, broad sets of features (i.e. syntactic features, lexicon features, cluster features) are adopted to train GBM and FFNN. The three approaches are finally ensembled by the weighted average of predicted probabilities of each emotion label.

1 Introduction

Twitter is an active social networking platform. It is estimated that nearly 500 million tweets are sent per day[1]. As a short message where people can convey their emotions, twitter data is particular interesting for emotional detection. The task of WASSA 2018 Implicit Emotion Shared Task is aimed to predict the emotion underlying in the tweets. The emotional types that are supposed to predict are ”Anger, Fear, Sadness, Joy, Surprise, Disgust”. In each tweet, the emotional expression is implicit, that is, a certain emotional word is removed. The removed emotional words could be one of the following:”sad”, ”happy”, ”disgusted”, ”surprised”, ”angry”, ”afraid” or a synonym of one of them. For example, given the tweet ”It’s [#TARGETWORD#] when you feel like you are invisible to others.”, the system should predict the label of this tweet as ”Sadness”. Moreover, the system can not only be useful for implicit emotion detection but also for various NLP applications. For example, this system can be used to detect the emotions in movie reviews which do not have the sentimental word but actually express sentimental polarities. In this paper, we describe our approaches and experiments to solve this problem. Our system is an ensemble of three classification approaches combined with a weighted average of predicted probabilities. Whilst, two of the three approaches are neural network models and the other is a gradient boosting regression tree model (Section 3). The rest of the paper is structured as follows: Section 2 discusses in brief the dataset for the task. Section 3 gives an explanation about the details of various approaches used in our system. Section 4 shows the results and discussions about them. Finally, we draw the conclusion about our participation in the Section 5.

2 Data

The dataset provided in this task contains the tweet text and the target emotions which are the predicted labels. The gold labels of test set are given only in the evaluation period. In the training data, there are 153383 tweets for training, 9591 tweets in the development dataset and 28757 in the test dataset. We also make use of external dataset which contains 640 million tweets. The external data is used for training the word embedding as the input of the deep learning model.

3 Proposed system

Our system is an ensemble of three different models. We demonstrate the separate models followed by the ensemble method. We tokenize each tweet with the tokenization tool **tweetokenize** [2]. Since

[1] https://en.wikipedia.org/wiki/Twitter

[2] https://github.com/jaredks/tweetokenize

Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 201–204
Brussels, Belgium, October 31, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

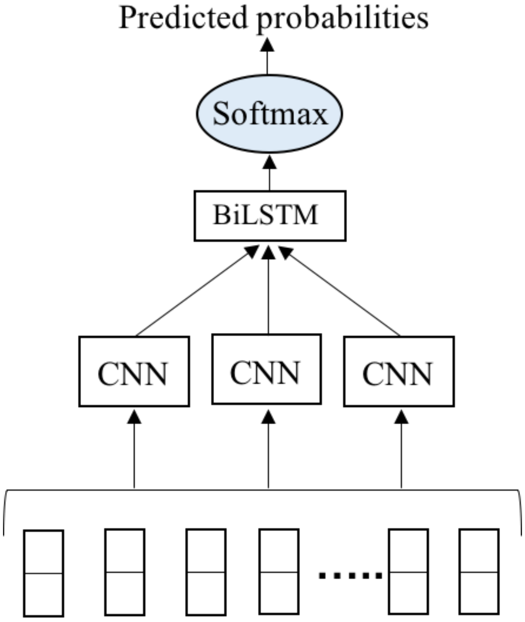

Figure 1: Architecture of ConBLSTM.

all the hashtags have been removed, we make no changes about hashtags.

3.1 Approach 1: convolutional BLSTM

To capture the sentence-level features and lexical information hiding in the tweets, we utilize a convolutional BLSTM model without any hand-crafted features. convolutional BLSTM has showed strong advancement in various NLP domains (Zeng et al., 2016), (Eger et al., 2017).

Input features: We trained word embedding on 470 million emotion related tweets using GloVe method. The 470 million emotion related tweets are filtered from 640 million tweets. The filtering process are based on a pre-built emotion word list which was extracted from NRC Word-Emotion Association Lexicon (Mohammad and Turney, 2013). With NRC Word-Emotion Association Lexicon, we extract the word which has at least one positive emotion. In addition, emojis are incorporated into the pre-built emotion word list because emojis as another expression of emotions are helpful for the emotion prediction. The emoji list is extracted from python package *emoji*[3]. Then the filtered tweets are used to train the word embedding. Considering the experiment efficiency and performance, we finally select the trained word embedding with dimension 100.

Architecture: After loading the pre-trained GloVe word embedding, we apply 3 convolutional layers with filter sizes 3, 5, 7. We concatenate the respective vectors and feed them into the forward and backward LSTM layers. The output of the BLSTM layer is put into the softmax layer to compute the probabilities of each emotion. The final predicted emotion type is the one with the max probabilities. Figure 1 is used to illustrate the architecture of this model.

Training: The epoch and filter number in each layer in the final experiment are chosen based on the result of pre-experiments. The epoch is 100 and the filter number in the 3 convolutional layers are 512. The output dimension of BLSTM is 128 which is also selected based on pre-experiments.

3.2 Approach 2: Feed-forward neural neural network

Inspired by the previous work of Pranav et al. 2017 (Goel et al., 2017), which is a system to predict the emotional intensity, we choose feed forward neural network due to its advantage of efficiency and effectivity in the classification task. We spell out the architecture as follows:

Input features: Given a tokenized sentence with words $\{w_0, w_1, w_2..., w_n\}$, the first step is to extract word-level and character-level representations by vectorizing word and character ngrams. The next step is to extract a fixed length sentiment feature representation. Each tweet is represented as a sentence vector by concatenating broad sets of character-level representations, word-level representations and sentiment feature representations. To handle with the problem that the length of each tweet vector varies, we utilize the SciKit-Learn tool **DictVectorizer**[4] and **CountVectorizer**[5]. We adopted features including character ngram feature, POS feature, cluster feature, negation feature, word ngram feature, counting feature and lexicon feature. Details of these features are explained in Liu (2018). A variety of sentiment lexicons are explored in the lexical features[6]. All these features are concatenated as the input features. Dimension

[3] https://pypi.org/project/emoji/

[4] http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.DictVectorizer.html

[5] http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[6] NRC Emotion Lexicon; NRC Hashtag Sentiment Lexicon; MaxDiff Twitter Lexicon; MPQA Effect Lexicon; MPQA Subjectivity Lexicon; Harvard Inquirer Lexicon; Bing Liu Lexicon; Loughran McDonald Lexicon; Amazon Laptop Review Lexicon; Sentiment140 Lexicon.

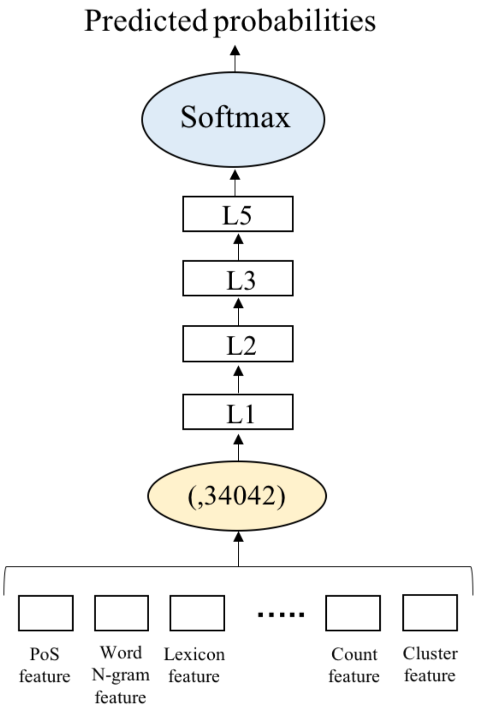

Figure 2: Architecture of feed forward neural network (FFNN). 34042 shows the number of input tweets.

of each sentence vector is not fixable and dependent on the corpus. For the training dataset with 153383 samples, the dimension of each sentence vector is 41298.

Architecture: Firstly, sentence vectors are fed into the input layer and then passed to four hidden layers (L_1, L_2, L_3, L_4). L_1, L_2 and L_3 are all followed by dropout *(p=0.5)* to avoid over-fitting and co-adaptation of features (Srivastava et al., 2014). Activation functions in the hidden layers are *Relu* (Maas et al., 2013). L_4 is followed by a softmax neuron which predicts the probability of each emotion. We use Figure 2 to illustrate the architecture of this network.

Training: Parameters are optimized in the neural network by performing 5-fold cross validation. We use a batch size of 128 and 100 epochs. The optimization algorithm is Adam (Kingma and Ba, 2014) with the default parameter setting in Keras[7].

3.3 Approach 3: GBM

The input features of this models are same with those of the previous model feed forward neural network. Input features are concatenated and fed into the model Gradient Boosting Regression Tree (GBM). GBM previously shows efficiency and power to take use of broad sets of sentimental features in predicting the type of emojis (Liu, 2018). In this task, despite of implication of the emotion words, GBM still has a comparable performance. From the previous experiments, we choose two hyperparameters to tune and use 300 trees and 64 leaves per tree in this model. Besides, we set learning rate to 0.1 and minimal number of data in one leaf to 20. The tool we used to build GBM model is lightGBM (Ke et al., 2017).

3.4 Ensemble of the three approaches

Out final submitted system is ensembled by the previously described three approaches. We compute the weighted average of the predicted probabilities and determine the final label using the max probability. Due to the time limit, our submitted system does not use the global optimization of ensemble weights. After submitting, we tune the weights of each model with intensive experiments. The final weights for our system are: 2 (FFNN), 2 (ConBLSTM) and 1 (GBM), while the weights of submitted system are 4 (FFNN), 2 (ConBLSTM) and 1 (GBM).

4 Result and discussion

We compare the results achieved by our individual approaches, the ensemble system and the WEKA Baseline system which is the official baseline model for this task. The official score of our submitted system is 0.621. Table 1 and 2 shows the results of our systems with the best weight settings on development and test dataset. From table 1, we can find the ensemble model achieves the best performance compared with the single model and FFNN+BLSTM ensemble model. Approach 1 (ConBLSTM) achieves the lowest scores among the three approaches. Table 2 illustrates that among all the individual emotions, our system performs best on ”Joy” which has the most labels in both the development dataset and the test dataset.

5 Conclusion and future work

In this paper, we propose an ensemble system to predict the implicit emotion of tweets. Three approaches are exploited: convolutional BLSTM, feed forward neural network and gradient boosting regression tree. To ensure the replicability, each approach is detailed about the architecture and the input features. In the future work, we will carry

[7] https://keras.io

	Development			Test		
System	P	R	F1	P	R	F1
Feed Forward NN	0.61	0.61	0.61	0.62	0.61	0.62
ConBLSTM	0.55	0.55	0.55	0.55	0.55	0.55
GBM	0.62	0.62	0.62	0.62	0.62	0.62
Feed Forward NN + BLSTM	0.61	0.61	0.61	0.63	0.63	0.63
Ensemble Model	0.64	0.63	0.63	0.64	0.64	0.64
Baseline	0.60	0.60	0.60	0.60	0.60	0.60

Table 1: Test and Development results on our system.

	Development			Test		
Label	P	R	F1	P	R	F1
Anger	0.57	0.57	0.57	0.53	0.53	0.53
Disgust	0.65	0.62	0.64	0.66	0.66	0.66
Fear	0.64	0.68	0.66	0.65	0.70	0.67
Joy	0.78	0.72	0.75	0.76	0.72	0.74
Sadness	0.65	0.58	0.61	0.65	0.57	0.61
Surprise	0.56	0.64	0.60	0.56	0.64	0.60

Table 2: Results on different labels.

out experiments with different dimensions of word embedding as well as different embedding methods, i.e. word2vector. Due to the time limit, the experiments lack feature analysis with which the final result may be improved. We plan to experiment different sets of features to find out which feature sets are more helpful. Finally, we would test more ensemble methods as well as the effectiveness of each approach in the ensemble system.

References

Steffen Eger, Erik-Lân Do Dinh, Ilia Kuznetsov, Masoud Kiaeeha, and Iryna Gurevych. 2017. Eelection at semeval-2017 task 10: Ensemble of neural learners for keyphrase classification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 942–946.

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.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Roman Klinger, Orphée de Clercq, Saif M. Mohammad, and Alexandra Balahur. 2018. Iest: Wassa-2018 implicit emotions shared task. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Brussels, Belgium. Association for Computational Linguistics.

Man Liu. 2018. Emonlp at semeval-2018 task 2: English emoji prediction with gradient boosting regression tree method and bidirectional lstm. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 390–394.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. 29(3):436–465.

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.

Ying Zeng, Honghui Yang, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2016. A convolution bilstm neural network model for chinese event extraction. In *Natural Language Understanding and Intelligent Applications*, pages 275–287. Springer.