Yuan at SemEval-2018 Task 1: Tweets Emotion Intensity Prediction using Ensemble Recurrent Neural Network

Min Wang, Xiaobing Zhou*
School of Information Science and Engineering
Yunnan University, Yunnan, P.R. China
*Corresponding author, zhouxb.cn@gmail.com

Abstract

This paper describes the performing system for SemEval-2018 Task 1 subtask 3 - Given a tweet, determine the intensity of sentiment or valence (V) that best represents the mental state of the tweeter—a real-valued score between 0 (most negative) and 1 (most positive). The proposed system gets features in tweets from the existing emotional dictionary and represents the word using word embedding, then utilizes the joint representations as the inputs of the bidirectional long short-term memory (BiLSTM) to learn and get the regression result. To boost performance we ensem- ble several BiLSTMs together. We ranked 6th in subtask 3 among all teams. Our approach achieves the Pearson(All instances) score 0.836 and Pearson(gold in 0.5-1) score 0.667, we outperform the baseline model of this task by 25.1% and 21.8% of Pearson(All instances) and Pearson(gold in 0.5-1) scores respectively.

1 Introduction

Sentiment analysis (SA) is a field of knowledge which deals with the analysis of people's opinions, sentiments, evaluations, appraisals, attitudes and emotions towards particular entities (Liu, 2012). EmoInt (Mohammad and Bravo-Marquez, 2017) is a shared task hosted by WASSA 2017, aiming to predict the emotion intensity in tweets. SemEval 2018 Task 1 subtask 3 (Mohammad et al, 2018) is similar to EmoInt, however the goal of subtask 3 is to detect valence or sentiment intensity, in which scores are floating point values between 0 and 1, representing low and high intensities of the emotion being expressed, respectively. Obviously we don't know in advance whether twitter's emotional intensity is positive or negative, but in EmoInt task we can determine whether twitter emotions are positive or negative based on one of four datasets: anger, fearness, joy, sadness. This is still a challenging task and remains active areas of research. These setbacks are: extensive usage of hashtags, slang, abbreviations, and emoticons. And tweets are usually typed on mobile devices like mobile phone, laptop or iPad which can result in a substantial amount of typos.

Existing methods for modeling emotion intensity rely vastly on manually constructed lexicons, which contain information about intensity weights for each available word (Mohammad and Bravo-Marquez, 2017a; Neviarouskaya et al., 2007). The intensity for the whole tweet can be deduced by combining individual scores of words, which is easy and ignores the word order compositionality of the language. Building such lexicons is a labour-intensive procedure. We can learn from these models the skills of combining feature extraction and classification or regression stages given a sufficient amount of training data.

Some deep learning methods are used to process the same question. Deep neural architectures for emotion intensity prediction in tweets (Goel et al., 2017) and character- and word-level recurrent neural models for tweet emotion intensity detection (Lakomkin et al., 2017).

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 205–209
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

In our work, we firstly clean tweets, then build lexical features and find optimal combinations of features to produce a final vector representation of a tweet, next train a neural network regression model and finally get the tweet's intensity scores. In addition, we adjust our models' parameters and through the ensemble models to get the best performing results.

2 Data cleaning

We use the dataset provided by the official organizers to train our system, there are 1181 labeled training tweets, 449 labeled dev tweets. Test set are unlabeled 17874 tweets and the gold labels were given only after the evaluation period. Before training model or predicting test set we firstly clean the tweets, this is imperative. We utilize the following prep- rocessing steps.

(1) **Hashtags** are crucial markers for determining sentiment. The "#" symbol is removed and the word itself is retained. Eg, a hashtag like "#the_best_one", finally we get "the best one".
(2) **Username mentions,** we replace it with "usename".
(3) **Shortening,** we transform word "don't", "I've", "I'll" et al into "do" "n't", "'ve", "'ll".
(4) **Punctuations,** only "!" and "?" are retained, others like ";" ">" ")" "," "-" are deleted.
(5) **Numerical symbols**, considering that the data in the dataset is relatively standardized and there are few numbers, so we remove the all digitals and only keep English words.
(6) Extra spaces are removed and all words become lowercase letters.

3 Feature Extraction

In order to completely extract features from tweets, we consider two characteristics which are annotated lexicons and pre-trained word embedding.

3.1 Annotated Lexicon

For extracting lexicon features, we follow the procedure as per the baseline system provided in the WASSA Emotion Intensity Task. The knowledge sources that have been used are: MPQA subjective lexicon (Wilson et al., 2005), Bing Liu lexicon (Ding et al., 2008), AFINN (Nielsen, 2011), Sentiment140 (Kiritchenko et al., 2014), NRC Hashtag Sentiment Lexicon (Mohammad and Kiritchenko, 2015), NRC Hashtag Emotion Association Lexicon (Mohammad et al., 2013), NRC Word-Emotion Association Lexicon(, 2013), NRC-10 Expanded Lexicon (Bravo Marquez et al., 2016) and the SentiWordNet (Esuli and Sebastiani, 2007). Two more features are calculated on the basis of emoticons (obtained from AFINN (Nielsen, 2011)) and negations present in the text. We use several of the above lexicons as following:

- **Emoji Valence (EV):** This is a hand classified lexicon of Unicode emojis, rated on a scale of -5 (negative) to 5 (positive).
- **SentiWordNet (SWN):** Calculates positive and negative sentiment score using SentiWordNet, which is an opinion mining resource available through NLTK.
- **Depeche Mood (DM)** (Staiano and Guerini, 2014)**:** This is a lexicon comprised of about 37,000 unigrams annotated with real-valued scores for the emotional states *afraid*, *amused*, *angry*, *annoyed*, *don't care*, *happy*, *inspired* and *sad*.
- **Emoticon Sentiment Lexicon:** Note that this is a sentiment lexicon drawn from emoticons, and is not an emotion lexicon.
- **NRC-Emoticon-AffLexNegLex-v1.0:** Each line of this lexicon represents a real-valued sentiment score: score = PMI(w, pos) - PMI(w, neg), where PMI stands for Point-wise Mutual Information between a term w and the positive/negative class.
- **NRC-Hashtag-Sentiment-Lexicon-v1.0** (Moh-ammad and Turney, 2013)**:** The lexicon is an association of words with positive (negative) sentiment generated automatically from tweets with sentiment-word hashtags.
- **NRC-Hashtag-Sentiment-AffLexNegLex-1.0:** The same lexicon as Sentiment 140, but here tw- eets with only emotional hashtags are considered during training.

3.2 Word Embedding

The text can be converted into word embedding, which represents each word of the text with a d dimensional vector (Mikolov et al., 2013). Considering that we have to deal with tweets, we use GloVe word embedding trained on 2 billion tweets from twitter (Pennington et al., 2014), vectors of 100, 200 and 300 dimensions are provided as part of the pre- trained model. For this work, we use the 300 dimensional vectors of 42B tokens. We also considered GoogleNews- vectors-negative300 in our expe-riments but the effects was not as good as the GloVe word embedding.

4 Model Training

Based on the application of features extractions and word embedding, we can represent each word in a tweet as a high dimensional space vector, and the dimension of the vector is $d + l$. d represents the dimension of GloVe word embedding 300 and l stands for the length of the additional lexical dictionary. After representing the tweets, we need to train models. Since the task requires the computation of a real valued emotion intensity score for the tweets in the test set, we explore several regression methods. Our system is implemented in Keras and we finally choose the best single BiLSTM model, which contains two layers of BiLSTM following the embedding layer and, we add a dropout layer. Some parameters of our model are: dropout probability 0.25 and 0.5 respectively; units of the BiLSTM layers are 512 and 256 respectively; units of the full connection layer is 256. The complete model structure is shown below Figure 1:

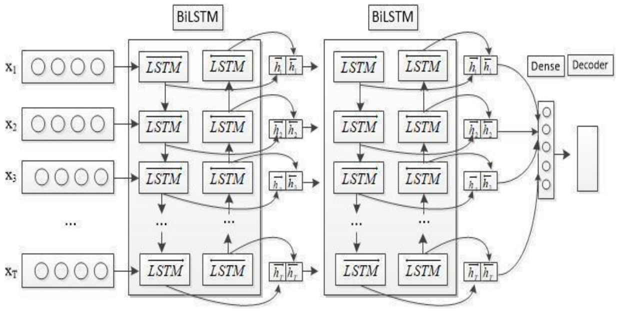

Figure 1: A two layer bidirectional LSTM model.

5 System tuning

When training model on Keras so there only some parameters need to change, we tune the parameters such as the choice of loss function, dropout probability, dimension of the BiLSTM layer. As for feature combination we use all the annotated lexicons mentioned in section 3.1 so as to control the variables and we don't consider the impact of different dictionary combinations on the results, which may be discussed in the future work. Note that all of our tuning processes are done on the development set, each time we finished a model we record the results. Ensembling of some models is universal used method to improve the performance of the overall system by combining predictions of several classifiers. Our system ensembles ten exactly the same BiLSTMs models and average the results, it turns out that the ensemble result is better than that of a single model. That is to say when we ensemble the model, the weight of each single BiLSTM is the same.

6 Experiment and results

All our experiments have been developed using Keras deep learning library with Theano backend, and with CUDA enabled. And all our experiments are performed on a computer with Intel Core(TM) i3 @3.4GHz 16GB of RAM and GeForce GTX 1060 GPU. After testing many neural network models, we finally find the best results on LSTM and BiLSTM models. Table 1 shows the results of a single layer LSTM changing the loss function and word embedding, we can learn that MAE loss function can get the best result with Glove word embedding, in general the performance on Glove word embedding is better than word2vec embedding. Table 2 shows the results of a single BiLSTM changing the loss function and integrating ten models under different loss functions and different word embedding we can learn that MAPE loss function can get the best result with Glove word embedding, in general the performance on Glove word embedding is better than word2vec embedding. Table 3 is the result of double layers BiLSTM changing the loss function and integrating ten models under different loss functions and different word embedding we can learn that MAPE loss function can get the best result with Glove word embedding, in general the performance on Glove word embedding is better than word2vec embedding.

The system in this subtask are evaluated using the Pearson correlation coefficient, which computes a bivariate linear coefficient, and the secondary evaluation metrics, which is Pearson correlation for a subset of the test set that includes only those tweets with intensity score greater or equal to 0.5. We present the results of the system submitted to the competition leaderboard in Table 4. The score of our system is 0.836 (Pearson) and 0.667 (Pearson gold in 0.5-1). Note that the model we used on the test set is the best model on the development set, i.e., in Table 3 the third line.

Loss function	Pearson score
MSE(Glove)	0.804
MAE(Glove)	0.818
MAPE(Glove)	0.815
MSLE(Glove)	0.801
MSE(w2v)	0.801
MAE(w2v)	0.798
MAPE(w2v)	0.799
MSLE(w2v)	0.786

Table 1: Performance on development dataset. Single layer LSTM under different loss functions and different word embedding.

Loss function	Pearson score
MSE(Glove)	0.799
MAE(Glove)	0.820
MAPE(Glove)	0.822
MSLE(Glove)	0.801
MSE(w2v)	0.797
MAE(w2v)	0.810
MAPE(w2v)	0.799
MSLE(w2v)	0.784

Table 2: Performance on development dataset. Ensemble result of single layer BiLSTM under different loss functions and different word embedding.

Loss function	Pearson score
MSE(Glove)	0.805
MAE(Glove)	0.826
MAPE(Glove)	0.827
MSLE(Glove)	0.806
MSE(w2v)	0.796
MAE(w2v)	0.785
MAPE(w2v)	0.794
MSLE(w2v)	0.783

Table 3: Performance on development data-set. Ensemble result of double layers BiLSTM under different loss functions and different word embedding.

#	Team	P	P (gold 0.5-1)
1	SeerNet	0.873	0.697
2	TCS Research	0.861	0.680
3	PlusEmo2Vec	0.860	0.691
4	NTUA-SLP	0.851	0.688
5	Amobee	0.843	0.644
6	Yuan	0.836	0.667
7	nlpzzx	0.835	0.670

Table 4: Performance on test dataset. Final results in about test set on leaderboard and our system ranks 6th overall.

7 Conclusions

In this paper, we propose a deep learning framework to predict the emotion intensity in tweets. The proposed system is based on two layers BiLSTM and the last layer of model using a linear regression so that we can get the intensity score, which is a consecutive emotional value. Before training model we implement features extraction and represent the tweets by word embedding. Both single model and ensemble model are described in detail with a view of making our experiments replicable. The optimal parameters are mentioned along with our method of bringing the approaches together. Our submitted system beats the baseline system by about 25.1% on the test set. Our source code is in here https://github.com/ynuwm/SemEval-2018

Acknowledgments

This work was supported by the Natural Science Foundations of China No.61463050, No.61702443, No.61762091, the NSF of Yunn an Province No. 2015FB113, the Project of Innovative Research Team of Yunnan Province.

References

Bing Liu. 2012. Sentiment analysis and opinion mining Synthesis Lectures on Human Language Technologies. Morgan &Claypool publishers.

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. Santos C D, Tan M, Xiang et al. 2016. Attentive Pooling Networks.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2007. Textual affect sensing for sociable and expressive online comm- unication. Affective Computing and Intelligent Interaction pages 218–229.

Ozan Irsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In the Proceedings of the Conference on EMLNP. pages 720–728.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentimenttreebank. In the Proceedings of the Conference on EMLNP. volume 1631, pages 1631–1642.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann.2005. Recognizing contextual polarity in phraselevel sentiment analysis. In *HLT/EMNLP 2005,Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6- 8 October 2005, Vancouver, British Columbia, Canada*. The Association for Computational Linguistics, pages 347–354.

Finn Arup Nielsen. 2011. A new ANEW: evaluation of a word list for sentiment analysis in microblogs. In Matthew Rowe, Milan Stankovic, Aba-SahDadzie, and Mariann Hardey, editors, *Proceedings of the ESWC2011Workshop on 'Making Sense of Microposts': Big things come in small packages, Heraklion, Crete, Greece, May 30, 2011*. CEUR-WS.org,volume 718 of *CEUR Workshop Proceedings*, pages 93–98.

Mohammad Saif and Svetlana Kiritchenko. 2015. Using hashtags to capture fine emotion categories from tweets. Computational Inte- lligence 31(2):301–326.

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

Jacopo Staiano and Marco Guerini. 2014. Depeche mood: A lexicon for emotion analysis from crowd-annotated news. *arXiv preprint arXiv:1405.1605* .

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. Pages 1532–1543.

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 (SemEval2018), New Orleans, LA, USA.