ISCLAB at SemEval-2018 Task 1: UIR-Miner for Affect in Tweets

**Meng Li[1, 2], Zhenyuan Dong[1], Zhihao Fan[1], Kongming Meng[1], Jinghua Cao[1],
Guanqi Ding[1], Yuhan Liu[1], Jiawei Shan[1], Binyang Li[1*]**

[1] School of Information Science and Technology, University of International Relations
[2] University of Pittsburgh
[*] Corresponding author
mel165@pitt.edu; {byli, zydong, zhfan, kmmeng, jhcao, gqding, jwshan}@uir.edu.cn

Abstract

This paper presents a UIR-Miner system for emotion and sentiment analysis evaluation in Twitter in SemEval 2018. Our system consists of three main modules: preprocessing module, stacking module to solve the intensity prediction of emotion and sentiment, LSTM network module to solve multi-label classification, and the hierarchical attention network module for solving emotion and sentiment classification problem. According to the metrics of SemEval 2018, our system gets the final scores of 0.636, 0.531, 0.731, 0.708, and 0.408 in terms of Pearson Correlation on 5 subtasks, respectively.

1 Introduction

Recently, social media platforms are becoming more and more popular, such as Twitter microblogging, Facebook, and so on. Through these platforms, online users would like to share their opinions and emotions. Therefore, the analysis about the information on “affect” in the social media has attracted much interest from both academia and industries.

However, the short texts are usually consisted of informal expressions with much casual forms and emoticons, it brings great challenges for such research.

For this purpose, SemEval organized the evaluation of sentiment analysis on Tweet. This year comes the fifth edition that consists of new genres, including emotion intensity regression task, emotion intensity ordinal classification task, sentiment intensity regression task, sentiment degree ordinal classification task, and emotion classification task (Mohammad et al., 2018).

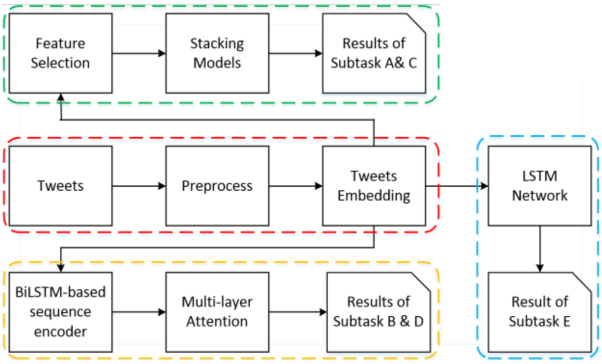

Figure 1: System architecture.

We participated in SemEval 2018 task 1 for English, i.e. Affect in Tweet. Our system considers EI-reg and V-reg (subtask A and C) as regression problems to get the emotion intensity and sentiment intensity by using regression models, while regards EI-oc and V-oc (subtask B and D) as categorization problems to classify each tweet into its corresponding emotion category and sentiment category by implementing hierarchical attention networks. Moreover, subtask E, i.e., E-c, is considered as a multi-label classification task.

This paper is organized as follows. Section 2 overviews the framework of our system. Section 3 describes the methods for subtask A and C. Section 4 describes the hierarchical attention networks for subtask B and D. Subtask E will be introduced in Section 5. Section 6 presents the evaluation results. Section 7 will conclude this paper.

2 System Overview

The architecture of UIR-Miner is shown in Figure 1. UIR-Miner system is comprised of 4 modules:

(1) Preprocessing module: involves data cleaning, topic classification, and tweets embedding.

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 286–290
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

(2) Regressor module: creates an ensemble regressor model by using different basic models simultaneously to calculate the emotion intensity and sentiment intensity, i.e. subtask A and subtask C;
(3) Classification module: constructs an LSTM network with multi-layer attention mechanism for emotion and sentiment categorization, i.e. subtask B and subtask D;
(4) Multi-label Classification module: builds a LSTM network for subtask E.

2.1 Preprocessing

Our system will firstly preprocess the Tweets data, and the main steps are as follows.

- Delete the unrelated texts, including the id, some mentions, stop words, and some meaningless punctuation combinations.
- Normalize synonymous words, like replacing “cant” and “can’t” with “cannot”.
- Extract emoticons from tweets through regular expressions, and then maintain the emotional ones.

2.2 Word embedding

In the preprocessing, we used the pre-trained word embedding by Glove (Penningto et. al, 2014), in which each word e_{it} will be represented by a 200-dimensional vector w_{it}, $i \in [1, L]$, $t \in [1, T]$. Here, i denotes the location of the sentence in the tweet and L is the maximum number of sentences for each tweet, t denotes the location of the word in the sentence and T is the maximum number of words for each sentence. Set $T = 140$ and $L = 5$.

3 Subtask A and C

This section will describe the methods for subtask A and C. Given a tweet and an emotion E (or a sentiment V), determine the intensity of E (or V) that best represents the mental state of the tweeter—a real-valued score between 0 and 1. We consider both of subtask A and C as a regression problem.

On the whole, we use a stacking framework to enhance the accuracy of final prediction. The original features are selected as input into the stacking model, including hashtags, emoticons, and *n-gram* features. Then, the stacking model is divided into two layer, the base layer and the stacking layer. In the base layer, we choose four basic regressors due to their excellent performance. In the stacking layer, we still use SVM model, especially, the NuSVR model, which can control its error rate. Finally, we get the final result of intensity value.

3.1 Feature Selection

Since there are many irregular expressions in tweet, we combine the features, including emoticon, hashtag, and special punctuations. In our system, we mainly select the following features:

- Hashtags: the number of hashtags in one tweet;
- Ill format: the presence of ill format with some characters replacing by *;
- Punctuation: the number of contiguous sequences of exclamation marks, question marks, and both exclamation and question marks; whether the last token contains an exclamation or question mark;
- Emoticons: the presence of positive and negative emoticons at any position in the tweet; whether the last token is an emoticon;
- OOV: the ratio of words out of vocabulary;
- Elongated words: the presence of sentiment words with one character repeated more than two times, for example, ‘cooool’;
- URL: whether the tweet contains a URL.
- Reply or Retweet: is the current tweet a reply/ retweet tweet.

3.2 Stacking Model

To avoid overfitting, we test 6 basic models to construct our stacking model.

- B: Bayesian Ridge (Hsiang, T.C 1975)
- G: Gradient Boosting Regressor (Jerome H. Friedman, 2001)
- K: Kernel Ridge (Zhang Y et. al, 2013)
- L: Lasso Regressor (Tibshirani et al., 1996)
- M: MLP Regression (Pal and Mitra, 1992)
- R: Random Forest Regressor (Ho, 1995)
- S: SVR (Vapnik 1995)

To achieve the best performance, we also compare different combinations of our basic models with the metrics of Mean Squired Error (MSE) in the stacking method, and the experimental result is shown in Table 1.

- Baseline: we use SVR as the Baseline;
- Stacking1: B+K+S;
- Stacking2: M+K+R;
- Stacking3: B+K+R+S;
- Stacking4: B+G+K+M;
- Stacking5: G+K+L+ S;
- Stacking6: B+G+K+S.

Since Stacking 6 achieves the best performance, we use the same setting in our system.

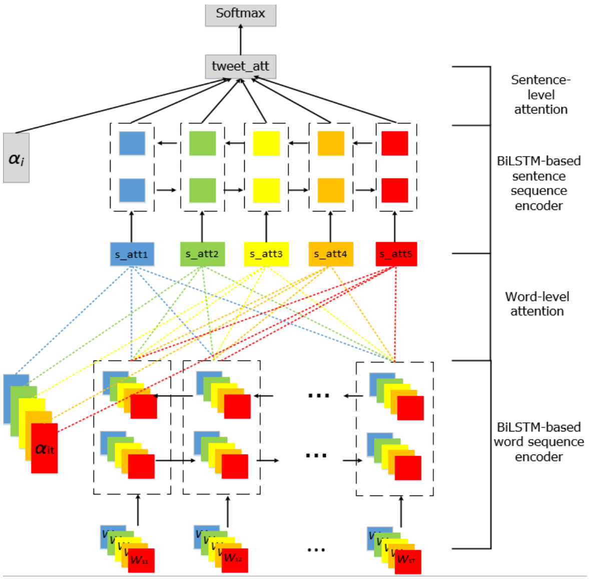

Figure 2: BiLSTM network with multi-layer attention mechanism.

Table 1: Evaluation on different combinations in stacking method.

Method	Metrics				
	Ang	Fear	Joy	Sad	Ave
Baseline	9.774	8.390	9.055	9.086	9.076
Stacking1	9.404	7.926	8.352	8.629	8.578
Stacking2	9.596	7.849	8.192	8.520	8.539
Stacking3	9.351	7.900	8.206	8.536	8.500
Stacking4	9.557	7.715	8.045	8.454	8.443
Stacking5	9.381	7.790	8.170	8.387	8.432
Stacking6	**9.300**	**7.766**	**7.794**	**8.334**	**8.298**

4 Hierarchical Attention Networks for Subtask B and D

This section will introduce our hierarchical attention model for subtask B and D. Given a tweet and an emotion category E (or a sentiment category V), classify the tweet into one of the ordinal classes of intensity of E (or V) that best represents the mental state of the tweeter. Note that, the number of category of E is 4, while that of V is 7. In our system, we consider both of subtask B and D as a classification problem.

Each tweet contains several sentences that are comprised by several words. In order to better represent the semantics of emotion or sentiment, we utilize the hierarchical structure of a tweet to capture the contextual information of both intra and inter-tweet. The architecture is shown as Figure 2.

We build a hierarchical model which contains two layers, word layer and sentence layer. Since words and sentences are highly sensitive to the contexts, recurrent neural networks based on bidirectional long short-term memory (BiLSTM) (Hochreiter and Schmidhuber, 1997) are implemented on both layers to get tweets' representations. Furthermore, since the words in one sentence or different sentences in a given tweet can indicate different emotion intensity or sentiment intensity. To better represent the semantics, attention mechanisms are added to both layers respectively (Xu et. al., 2015). We then use softmax as the activation

4.1 BiLSTM-based Word Encoder

A word level BiLSTM (Hochreiter and Schmidhuber, 1997) is used to represent each word. The BiLSTM consists of the forward LSTM and the backward LSTM. Forward LSTM reads the sentence s_i from e_{i1} to e_{iT} and represents the word e_{it} as $\overrightarrow{LSTM}(w_{it}), t \in [1, T]$. Backward LSTM reads the sentence s_i from e_{iT} to e_{i1} and represents the word e_{it} as $\overleftarrow{LSTM}(w_{it}), t \in [T, 1]$. Then word e_{it} can be annotated by combining both forward information and backward information, $h_{it} = [\overrightarrow{LSTM}(w_{it}), \overleftarrow{LSTM}(w_{it})]$. The equations are listed as follows:

$$i_t = \sigma(W_i w_{it} + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_f w_{ft} + U_f h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma(W_o w_{ot} + U_o h_{t-1} + b_o) \quad (3)$$

$$u_t = \tanh(W_u w_{ut} + U_u h_{t-1} + b_u) \quad (4)$$

$$c_t = i_t \odot u_t + f_t \odot c_{t-1} \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

where i_t, f_t and o_t are the input gate, forget gate and output gate, σ is the logistic sigmoid function, $\odot$ denotes elementwise multiplication, *tanh* is the network output activation function, and *softmax* is used for categorization. To better support Twitter, we input the word embedding with 200 dimensions, and the max number of words in a sentence as 140.

4.2 Word Layer Attention

Different weights α_{it} are given to different words. Attention mechanism (Xu et. al., 2015) is added to the word layer and the sentence s_i can be represented as s_att_i.

$$u_{it} = \tanh(W_w h_{it} + b_w) \quad (7)$$

$$\alpha_{it} = \frac{\exp(u_{it}^T u_w)}{\sum_t exp\,(u_{it}^T u_w)} \quad (8)$$

$$s\_att_i = \sum_t \alpha_{it} h_{it} \qquad (9)$$

More specifically, after putting h_{it} into a fully-connected layer, we get u_{it}. Then calculate weight α_{it} with a word level context u_w. Finally, we can get the sentence vector through an attention layer by calculating the sum of $\alpha_{it} h_{it}$.

4.3 Sentence Layer Attention

Similarly, a sentence level BiLSTM (Hochreiter and Schmidhuber, 1997) can be used to represent sentence s_i by adding sentence level context information,

$h_i = [\overrightarrow{LSTM}(s_{att_{it}}), \overleftarrow{LSTM}(s_{att_{it}})]$.

We then add weights to different sentence. Take x_i as input and get $tweet\_att$ to represent each tweet through an attention layer.

$$u_i = \tanh(W_s h_i + b_s)$$

$$\alpha_i = \frac{\exp(u_i^T u_s)}{\sum_i exp\,(u_i^T u_s)}$$

$$tweet\_att = \sum_i \alpha_i h_i$$

More specifically, after putting h_i into a fully-connected layer, we get u_i. Then calculate weight α_i with a sentence level context u_s. Finally, we can get the tweet vector through an attention layer by calculating the sum of $\alpha_i h_i$.

5 Subtask E

This section will introduce neural network model for subtask E. Given a tweet, classify the tweet as "neutral or no emotion" or as one, or more, of eleven given emotions that best represent the mental state of the tweeter.

Each tweet will be classified with different numbers of labels. Since there exists eleven labels each of which may be suitable, considering one of these labels every time is reasonable. Our system will calculate a score for each of the eleven labels for each tweet, and select the top-3 as the final results.

We also used a LSTM network for this task, and get the classification result by using *softmax*. The other settings of this model is quite similar to that in Section 4 except for multi-label classification.

6 Experiment

In this section, we will report our evaluation results in SemEval 2018 based on the given dataset as well as the metrics. The statistics of the dataset is shown in Table 2.

Note that any other extra external resources, such as sentiment lexicon, emoticons, and annotated corpus, are not used in the evaluation except for the training dataset provided by the organization.

Table 2: Statistics of the dataset.

	Training set	Dev set	Test set
EI-reg	anger: 1701 fear: 2252 joy: 1616 sadness: 1533	anger: 388 fear: 389 joy: 290 sadness: 397	anger: 17939 fear: 17923 joy: 18042 sadness: 17912
EI-oc	anger: 1701 fear: 2252 joy: 1616 sadness: 1533	anger: 388 fear: 389 joy: 290 sadness: 397	anger: 1002 fear: 986 joy: 1105 sadness: 975
V-reg	1181	449	17874
V-oc	1181	449	937
E-c	6838	886	3259

Table 3 shows the results of our UIR-Miner for all the subtasks on both Dev set and Test set, and the final ranking.

Table 3: The results on different datasets.

	Score in Dev	Score in Test	Ranking
EI-reg	0.576	0.636	28/48
EI-oc	0.495	0.531	15/39
V-reg	0.729	0.781	21/38
V-oc	0.694	0.708	16/37
E-c	0.421	0.407	23/35

7 Conclusion

In this paper, we present a framework for SemEval 2018 Affect in Tweet task. After the preprocessing, we firstly propose an ensembling method to calculate the intensity score of emotion and sentiment. Then a LSTM network model with multi-layer attention mechanism is constructed for emotion and sentiment classification. According to SemEval 2018's metrics, our model runs got final scores of 0.636, 0.531, 0.731, 0.708, and 0.408 in terms of Pearson Correlation on 5 subtasks, respectively.

Acknowledgements

This paper is funded by the National Natural Foundation of China 61502115, 61602326, U1636103, U1536207, 61572043, 61672361, 61632011, the Hong Kong Applied Science and Technology Research Institute Project 7050854, and the Fundamental Research Fund for the Central Universityies 3262015T70, 3262017T12.

References

Alex J. Smola, Bernhard Schölkopf. 2004. A tutorial on support vector regression. In *2004 Kluwer Academic Publishers*, pages 199-222

Hsiang, T.C. 1975.A Bayesian View on Ridge Regression. In *Journal of the Royal Statistical Society*, page 267-268.

Zhang Y, Duchi J, Wainwright M. 2013. Divide and conquer kernel ridge regression. In *Conference on Learning Theory*, pages 592-617.

Jerome H. Friedman. 2001. Greedy function approximation: a gradient boosting machine. In Annals of Statistics, pages 1189–1232

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

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 Task1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018).*

Saif M. Mohammad and Svetlana Kiritchenko. 2018. Understanding emotions: A dataset of tweets to study interactions between affect categories. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference*.

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

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048-2057.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480-1489.

Tibshirani R, Bickel P, Ritov Y, et al. Least absolute shrinkage and selection operator[J]. 1996.

Ho T K. Random decision forests[C]//Document analysis and recognition, 1995, proceedings of the third international conference on. IEEE, 1995, 1: 278-282.

Pal S K, Mitra S. Multilayer perceptron, fuzzy sets, and classification[J]. IEEE Transactions on neural networks, 1992, 3(5): 683-697.