YNUDLG at SemEval-2017 Task 4: A GRU-SVM Model for Sentiment Classification and Quantification in Twitter

Min Wang[1], Biao Chu[1], Qingxun Liu[1], Xiaobing Zhou[1]
[1]School of Information Science and Engineering, Yunnan University, Yunnan, P.R. China
{1147192499, 372744991, 1243305076, zhouxb.yn}@qq.com

Abstract

Sentiment analysis is one of the central issues in Natural Language Processing and has become more and more important in many fields. Typical sentiment analysis classifies the sentiment of sentences into several discrete classes (e.g.,positive or negative). In this paper we describe our deep learning system(combining GRU and SVM) to solve both two-, three- and five- tweet polarity classifications. We first trained a gated recurrent neural network using pre-trained word embeddings, then we extracted features from GRU layer and input these features into support vector machine to fulfill both the classification and quantification subtasks. The proposed approach achieved 37th, 19th, and 14rd places in subtasks A, B, and C, 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). Typical approaches to sentiment analysis is to classify the sentiment of a sentence into several discrete classes such as positive and negative polarities, or six basic emotions: anger, happiness, fear, sadness, disgust and surprise (Ekman,1992). SA is widely considered to be one of the most popular and challenging, competitive and the hot research area in computational linguistics. There are many ways to tackle the sentiment classification problems, such as random forest, support vector machine (SVM), Bayes classifier. In addition, there are many challenges, such as analysis of noise texts (e.g. oral language) in natural language processing tasks, despite numerous notable advances in recently years (e.g., Breck et al,. 2007; Yessenalina and Cardie, 2011; Socher et al,. 2011). Based on this, our way is to extract features with Gated Recurrent Unit (GRU) and classify sentences by SVM using these features.

Task 4 subtask A is to classify a tweet's sentiment as positive, negative, or neutral. Subtask B (Tweet classification according to a two-point scale) requires classifying a tweet's sentiment towards the given topic. Similar to B, subtask C is a five-point scale (Nakov et al., 2016). Unlike typical classification approaches, ordinal classification can assign different ratings (e.g., very negative, negative, neutral, positive and very positive) according to the sentiment strength (Taboada et al., 2011; Li et al., 2011; Yu et al., 2013; Wang and Ester, 2014).

This paper presents a system that combine GRU and SVM to process subtasks A, B and C. Our system uses a GRU neural network with word embeddings (Mikolov et al,. 2013) that are slightly fine-tuned (Yoon Kim et al,. 2014) on each training set. The word embeddings were obtained by training GloVe (Jeffrey Pennington et al,. 2014) on 2 billion tweets that we crawled for this purpose. These word vectors are then used to build sentence vectors through a recurrent convolutional neural network.

The proposed gated recurrent neural network consists of the GRU layer and SVM classifier. The choice based on the following two reasons: (1) it is more computational efficient than Convolution Neural Network (CNN) models (Lai et al., 2015); (2) unlike CNN, it also can extract long semantic patterns without tuning the parameter when training the model. Our system architecture is composed of a word embedding layer, drop out layer, GRU layer, a hyperbolic relu layer, SVM classifier and softmax layer.

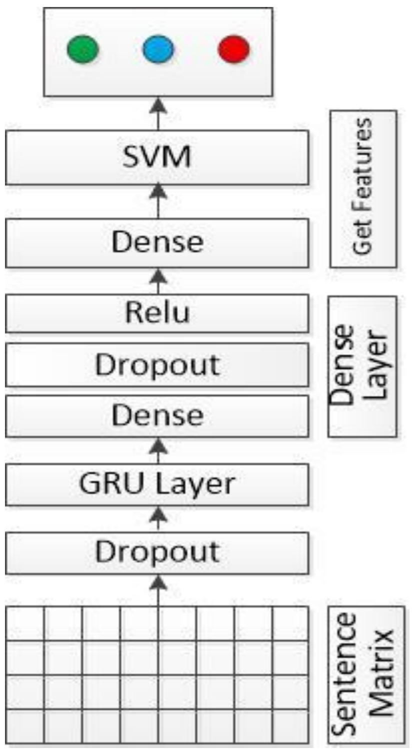

Figure 1: The architecture of the GRU + SVM system.

By capturing features from GRU layer, we obtain training and test data features, and integrate them with given labels as inputs, so SVM classifier can train the parameters.

2 Combining GRU and SVM for Sente– nces Classification

2.1 Embedding Layer

The first layer in the network , we let xi∈Rk be the k-dimensional word vector corresponding to the i-th word in the sentence. A sentence of length n (padded where necessary) is represented as:

$$x1{:}n = x1 \oplus x2 \oplus \ldots \oplus xn, \qquad (1)$$

n is the maximum length of sentences and we set it to 50. When meeting short tweets we use fixed characters padding. Each word xi is represented by embedding vectors (w1,w2...wl) where l is set to 200. For word embeddings, we use pre-trained word vectors from GloVe (Pennington et al., 2014). GloVe is an unsupervised learning technology for learning word representation. The purpose of training is to use statistical information to find similarities among words and based on co-occurrence matrix and statistical information. We use them to provide pre-trained word vectors trained on 27B tokens from Twitter and with a length of 200. Words not presented in the set of pre-trained words are initialized randomly.

2.2 GRU Layer

The main layer in our model, the input to it is the sequence of length L and each word in it having k dimension. The gated recurrent network proposed in (Bahdanauetal., 2014) is a recurrent neural network (a neural network with feedback connection, see(Atiya and Parlos, 2000)) where the activation hj of the neural unit j at time t is a linear interpolation between the previous activation h_t^j (Chung et al., 2014):

$$h_t^j = (1 - z_t^j) h_{t-1}^j + z_t^j h_t^j \qquad (2)$$

Where z_t^j is the update gate that determines how much time the units update its content, h_t^j is the newly computed candidate state.

2.3 Dropout Layer

Dropout is a regularization technique for reducing overfitting in neural networks by preventing complex coadaptations on training data Dropout refers to randomly let the weight of some hidden layer of network does not work in model when training, those nodes does not work can temporarily thought is not part of the network but the weight of it is retained(just temporarily not update), because it may work again the next time throwing samples into it.

2.4 Relu Layer

This layer is to allow the network to make complex decision by learning non-linear classification boundaries. We used more efficient function Rectified Linear Units (ReLU).

2.5 Soft-Max Layer

The output of the GRU layer and dropout layer is passed to a fully connected softmax layer. This layer calculates the classes probability di- stribution:

$$P(y=j|\mathbf{x},\mathbf{s},\mathbf{b})=\mathrm{softmax}_j(\mathbf{x}^T\mathbf{w}+\mathbf{b}) = \frac{\exp(x^T w_j + b_j)}{\sum_{k=1}^{K}\exp(x^T w_k + b_k)} \quad (3)$$

where w_k and b_k are the weight vector and bias of the k-th class, respectively. For subtask A and B, the difference is three neurons and two neurons used (i.e., K=3 or K=2).

2.6 SVM classifier

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. Given a set of training examples, each training instance is marked as belonging to one or the other of the two categories, the SVM training algorithm to create a new instance will be assigned to one of two categories of models, making it a nonprobability binary linear classifier. Sklearn is a Python library of scientific computing and it provides several clustering algorithms. In our system, we from sklearn.svm import SVC module and redefine this function and parameters. As we all known, SVM is a binary classifier, but we also need process three and five classification problem. In the multiclass case, this is extended as per Wu et al. (2004). SVC and NuSVC implement the “one-against-one” approach (Knerr et al., 1990) for multi-class classification. If n_class is the number of classes, then n_class * (n_class - 1) / 2 classifiers are constructed and each one trains data from two classes. To provide a consistent interface with other classifiers, the decisi- on_function_shape option allows to aggregate the results of the “one-against-one” classifiers to a decision function of shape (n_samples, n_classes).

3 Data

The training and development datasets used in our experiments were all datasets from SemEval 2013-2016 that labeled. Before training, we processed the data with the follow procedures:

1). The texts were lowercased by string.stri- p().lower(),

2).We only retain punctuation, exclamation mark, question mark and comma,

3). Tokenize each tweet using blank in sentences,

4). Emoticons like (^.^) have been deleted,

5).Using the patterns described in Table 1 to normalize each tweet.

Pattern	Examples	Normaliza
Usernames	@user1,@user2	UserName
Abbreviation	Don’t	Do not
Abbreviation	2c or a90	Delete
Repeated letters	ahahhhh	ahahh
Numbers	123	NUM
URLs	www.google.co	URL
Topic (Subtask B only)	Microsoft	Entity

Table 1: Normalization Patterns

Datasets	Total	Pos	Neg	Neu
Twitter2013	13454	5124	2097	6233
SMS2013	2093	492	394	1207
Twitter2014	1853	982	202	669
LiveJournal2014	1142	427	304	411
Tw2014Sarcasm	86	33	40	13
Twitter2015	2390	1038	365	987
Twitter2016	29632	11259	4483	3985

Table 2: Overview of datasets and number of tweets we d- owloaded. The data was divided into training, development and testing sets

4 Experiments and Results

All our experiments have been developed using Keras deep learning library with Theano backend, and with CUDA enabled. And all our experiments were performed on a computer with Intel Core(TM) i3 @3.4GHz 16GB of RAM and GeForce GTX 1060 GPU. The hyper-parameters of the network are chosen based on the performance on the dev-test data. We firstly carry out our system: put data into GRU model, as we known GRU can also train and test d- ataset at the same time we adjust some important parameters to make our GRU model to be the newest. Then we define a Theano function, and input para-meter is the GRU network portal, output is the GRU network dense layer before softmax layer. Using the Theano function, once we throw new data, we can get features that meeting our requirements. Last we throw these features into SVM so we can get classification results as we hope. After experiment we know our system performance well on two classification question, and poor on five classification. Some factors may cause this: small training data and in five-classification dataset many twitters belong to negative, neutral and positive; our deep model GRU maybe not actually called deep due to number of layers and our manually tuning.

4.1 Subtask A

Rank	System	$F1^{PN}$	ρ^{PN}	Acc
37	YNU-1510	0.201_{37}	0.340_{36}	0.387_{36}
baseline 1: all Positive		0.169	0.333	0.193
baseline 2: all Negative		0.244	0.333	0.323
baseline 3: all Neutral		0.000	0.333	0.483

Table 3: Result for Subtask A “Message Polarity classi– fication”, English. The systems are ordered by their F_1^{PN} score (higher is better).

4.2 Subtask B

Rank	System	ρ^{PN}	$F1^{PN}$	Acc
21	YNU-1510	0.516_{21}	0.499_{19}	0.499_{21}
baseline 1: all Positive		0.500	0.285	0.398
baseline 2: all Negative		0.500	0.376	0.602

Table 4: Results for Subtask B “Tweet classification a– ccording to a two-point scale”, English. The systems are ordered by their ρ^{PN} score (higher is better).

4.3 Subtask C

Rank	System	MAE^{M}	MAE^{μ}
14	YNU-1510	1.262_{14}	0.764_{14}
baseline 1: Highly NEGATIVE		2.000	1.895
baseline 2: NEGATIVE		1.400	0.923
baseline 3: NEUTRAL		1.200	0.525
baseline 4: POSITIVE		1.400	1.127
baseline 5: Highly POSITIVE		2.000	2.105

Table 5: Results for Subtask C “Tweet classification accor– ding to a five-point scale”, English. The systems are ordered by their MAEMscore (lower is better).

5 Conclusion

In this paper, we presented our GRU-SVM system used for SemEval 2017 Task4 (Subtasks A, B and C). The system used a gated recurrent layer as a core layer to extract features and then feed these features into SVM classifier.

Acknowledgments

This work was supported by the Natural Science Foundation of China No.61463050, and the NSF of Yunnan Province No. 2015FB113.

References

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

Paul Ekman. 1992. An argument for basic emotions. Cognition and Emotion, 6(3-4), 169-200.

Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In Manuela M. Veloso, editor, IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12,2007,pages 2683–2688.

Ainur Yessenalina and Claire Cardie. 2011. Compositional matrix-space models for sentiment a-nalysis. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 172–182.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive auto encoders for predicting sentiment distributions. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 151–161.ACL.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshopon Semantic Evaluation (SemEval 2013), pages 312–320, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016a. Evaluation measures for the SemEval-2016 task 4: ”sentiment analysis in Twitter”.

Preslav Nakov,Alan Ritter, Sara Rosenthal, VeselinStoyanov, and Fabrizio Sebastiani. 2016b. SemEval-2016 task 4: Sentiment analysis in Twitter. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, California, June. Association for Computational Linguistics.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2015. Character-aware neural language models. CoRR, abs/1508.06615.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semisupervised recursive autoencoders for predicting sentiment distributions. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 151–161.ACL.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv: 1409.0473.

Amir F Atiya and Alexander G Parlos. 2000. New results on recurrent network training: unifying the algorithms and accelerating convergence. NeuralNetworks, IEEE Transactions on, 11(3):697–709.