YNU-HPCC at IJCNLP-2017 Task 4: Attention-based Bi-directional GRU Model for Customer Feedback Analysis Task of English

Nan Wang, Jin Wang and **Xuejie Zhang**
School of Information Science and Engineering
Yunnan University
Kunming, P.R. China
Contact:xjzhang@ynu.edu.cn

Abstract

This paper describes our submission to IJCNLP 2017 shared task 4, for predicting the tags of unseen customer feedback sentences, such as comments, complaints, bugs, requests, and meaningless and undetermined statements. With the use of a neural network, a large number of deep learning methods have been developed, which perform very well on text classification. Our ensemble classification model is based on a bi-directional gated recurrent unit and an attention mechanism which shows a 3.8% improvement in classification accuracy. To enhance the model performance, we also compared it with several word-embedding models. The comparative results show that a combination of both word2vec and GloVe achieves the best performance.

1 Introduction

Understanding and being able to react to customer feedback is the most fundamental task in providing good customer service. The goal of task 4 of the custom feedback analysis of IJCNLP-2017 is to train classifiers for the detection of meaning in customer feedback provided in English, French, Spanish, and Japanese. This task can be considered a short-text classification task, which has recently become popular in many areas of natural language processing, including sentiment analysis, question answering, and dialog management. The feature representation of a short text is a key to classification, which is usually extracted as features based on uni-gram, bi-gram, n-gram, or other combination patterns of the bag-of-words (BoW) model.

Deep neural networks (Hinton and Salakhutdinov, 2006) and representation learning (Bengio et al., 2003) have recently brought new ideas to resolving the data sparsity problem, and various neural models for learning word representations have been proposed (Bengio et al., 2003; Collobert et al., 2011; Huang et al., 2012; Mikolov et al., 2013b). Mikolov et al. (2013b) showed that meaningful syntactic and semantic regularities can be captured in pre-trained word embedding. The embedding model measures the word relevance by simply using the cosine distance between two embedded vectors.

Using pre-trained word embedding, neural networks have achieved remarkable results, including a convolutional neural network (CNN) (Collobert et al., 2011) and recurrent neural network (RNN) (Mikolov et al., 2010). Furthermore, several advanced architectures such as long short-term memory (LSTM) (Hochreiter and Urgen Schmidhuber, 1997) and a gated recurrent unit (GRU) (Cho et al., 2014) have been proposed owing to their better ability to capture long-term dependencies. They are equipped with gates to balance the information flow from the previous and current time steps dynamically. In addition, neural processes involving attention have been extensively studied in the field of computational neuroscience (Itti et al., 1998; Desimone and Duncan, 1995). Recent studies have shown that attention mechanisms are flexible techniques, and that new changes can be used to create elegant and powerful solutions. Yang et al. (2016) introduced an attention mechanism using a single matrix and outputting a single vector. Instead of deriving a context vector in terms of the input, a summary is calculated by referring to the context vector learning as a model parameter. Raffel and Ellis (2015) proposed a feed-forward network model with an attention mechanism, which selects the most important element from each time step using learnable weights depending on the target. In addition, Parikh et al.

Proceedings of the 8th International Joint Conference on Natural Language Processing, Shared Tasks, pages 174–179,
Taipei, Taiwan, November 27 – December 1, 2017.

(2016) introduced an attention mechanism for two sentence matrices, which outputs a single vector, and built an alignment (similarity) matrix by multiplying learned vectors from each matrix, computing the context vectors from the alignment matrix, and mixing with the original signal.

In the present study, we used a uni-gram and bi-gram as features for a support vector machine (SVM) and naïve bayes as baseline methods. A deep learning method was also implemented for better text classification results. We created our model using a bi-directional gate recurrent unit (Bi-GRU) with an attention mechanism, and compared the results with different word-embedding models (word2vec, GloVe, and their concatenate modes). We found that our model using word2vec or GloVe slightly outperformed the baseline methods, whereas the ensemble model using both word2vec and GloVe achieved better performance in comparison to the other models.

2 Bi-GRUATT

Our model is based on a bidirectional GRU (Bahdanau et al., 2014) with an attention mechanism (Raffel and Ellis, 2015). GRU was designed to have more persistent memory, thereby making it easier to capture long-term dependencies than an RNN. Irsoy and Cardie (2014) showed that such a bi-directional deep neural network maintains two hidden layers, one for the left-to-right propagation, and the other for the right-to-left propagation. We chose the Bi-GRU model because it could obtain full information through two propagations. In addition, attention mechanisms allow for a more direct dependence between the states of the model at different points in time. In this section, our model is described using the following four steps: embedding, encoding, attending, and prediction. The model architecture is shown in Fig. 1.

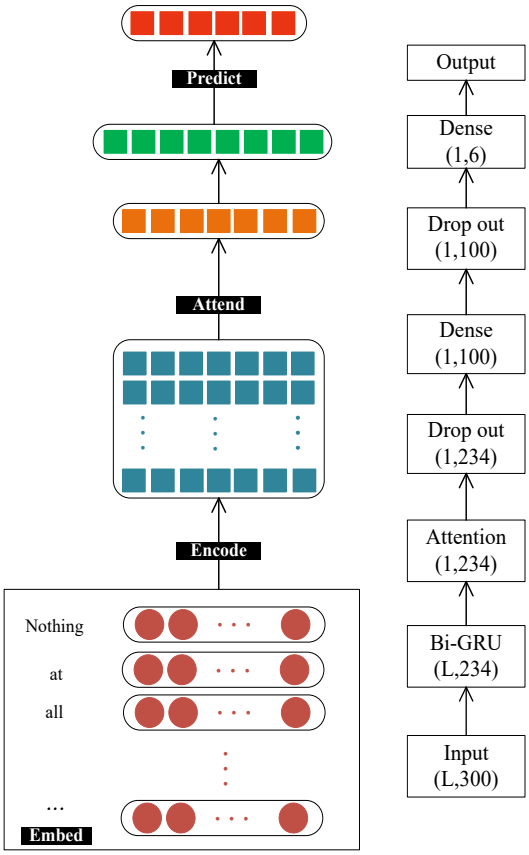

Figure 1: Architecture of Bi-GRUATT model. (the left-hand side ignores dropout layers which marked up on the right hand side, and L is the maximum length of all training texts.)

Embedding. We took size L tokens of text as input, where L was the maximum length of all training texts. In this English training dataset, L is 117. In addition, every word in the text was embedded into a 300-dimensional vector through the pre-trained embedding model. For those words that cannot be recognized in the pre-trained model, the same dimensional vector of zeros was replaced. This was also used for padding out the sentence when it was shorter than L.

Mikolov et al. (2013a) proposed word2vec, which allows training on larger corpora, and showed how semantic relationships emerge from such training. Pennington et al. (2014) proposed the GloVe approach, which maintains the semantic capacity of word2vec while introducing statistical information from a latent semantic analysis (LSA), which shows improvement in semantic and syntactic tasks. We tested word2vec and GloVe on pre-trained embedding models, and combined these two vectors, converted from each model, to a new 600-dimensional vector to obtain the advantages of both word2vec and GloVe.

Encoding. Through the given sequence of word vectors, the encoding step computes a representation of a sentence matrix, where each row represents the meaning of each token in the context of the rest of the sentence. We used a bi-directional GRU to summarize the contextual information from both directions of a sentence text, and obtained a full sentence matrix vector by concatenating the sentence matrix vector forward and backward at each time step. Similarly to the L-

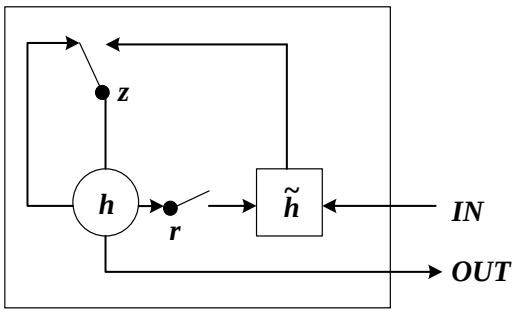

Figure 2: Illustration of gated recurrent units. (r and z are the reset and update gates, and h and $\widetilde{h}$ are the activation and the candidate activation.)

STM unit, the GRU has gating units that modulate the flow of information inside the unit, however, without having a separate memory cells (Chung et al., 2014). As is shown in Fig. 2. A GRU has two gates(a reset gate r, and an update gate z) rather than three gates in LSTM. Intuitively, the reset gate determines how to combine the new input with the previous memory, and the update gate defines how much of the previous memory to keep around. We ultized Bi-GRU instead of Bi-LSTM for its better performance on this task. For the activation function, softsign was used.

Attending. The attending step reduces the matrix representation generated by the encoding step into a single vector. The formula used by the attention mechanism to produce a single vector can be described as follows:

$$e_t = tanh(Wh_t) \quad (1)$$

$$\alpha_t = softmax(e_t) \quad (2)$$

$$c = \sum_{t}^{T} \alpha_t h_t \quad (3)$$

where h_t denotes the hidden state at each time step, and T is the total number of time steps in the input sequence. Vectors in hidden state sequence h_t are fed into the learnable function α_t to produce a probability vector α. Based on the weighting given by α, vector c is computed as the weighted average of h_t.

The characteristic advantage of an attention mechanism over other reduction operations is that the attention mechanism takes an auxiliary context vector as input. The context vector is crucial because it indicates which information to discard, and thus a summary vector is tailored to the network using it. The activation function of a dense layer during the attending step is tanh.

	Train	**Dev**	**Test**
Comment	1758	276	285
Complaint	950	146	145
Request	103	19	13
Bug	72	20	10
Meaningless	306	48	62
Undefined	22	3	4
Total	**3065**	**500**	**500**

Table 1: Customer feedback classification of English dataset distribution of the shared task.

Prediction. After the text has been reduced to a single vector, we can learn the target representation as a class label, real value, or vector. The activation function of a dense layer during the prediction step is softmax.

3 Experiment

For comparison, we used SVM and Naïve Bayes models as the baseline methods to evaluate our system performance. The following subsections describe the baseline methods and the bidirectional GRU approach with an attention mechanism. We then tested the system results using different word embedding models individually and in combination.

3.1 Datasets

For this shared task, we got four languages (English, French, Spanish and Japanese) of customer feedback datasets. In this paper, only English datasets were used. The training data published by the organizers included sets of sentences annotated with six tags, comments, requests, bugs, complaints, and meaningless or undetermined statements. Each sentence has at least one tag assigned to it, and might be annotated with multiple tags. The distribution of all datasets is shown in Table 1. The total number is not equal to the sum of each category because several samples have multiple labels. We took these multi-labeled samples as separate samples with the same text and different labels.

For a better performance with deep learning, we additionally crawled user comments from the

Booking[1] and Amazon APP[2] websites, and trained the word2vec (Mikolov et al., 2013b) word-embedding model on about 26,148,855 tokens. GloVe (Pennington et al., 2014) was also used to develop the system using pre-trained word vector glove.840B.300d, which was trained on 840B tokens and is publicly available[3].

3.2 Implementation Details

The two baseline methods were implemented using scikit-learn (Pedregosa et al., 2011) in Python. Instead of a simple whitespace tokenizer, we used Unitok[4] as a full tokenizer because of its better performance. Baseline methods were used one-vs-all SVM method with linear kernel and multinomial Naïve Bayes method. All parameters were adjusted using a grid search function. We experimented with uni-gram and bi-gram separately, and in combination, using the word level as features.

We implemented our model using the Python Keras library with a TensorFlow backend. The recurrent dropout rate of the GRU was set to 0.2, and two other layers with dropout rates of 0.3 and 0.5 were added before the dense layer during the attending and prediction steps, respectively, to avoid overfitting of the system. The model was trained using rmsprop with a mini-batch size of 32 to minimize the loss of function of a categorical cross entropy.

We found that the provided training data were imbalanced. The smallest number of class samples was only 22, which accounts for 0.7% of the entire training dataset. The largest number of class samples was 1,758, which accounts for 54.7% of the training data. For this reason, an additional parameter sample weight was set for balancing the data. The loss was multiplied by the sample weight to improve the accuracy of a small number of classes.

Finally, the epoch was set to depend on an early stop, which relied on a validation set to determine when to stop the training. The epoch was fixed at around 20.

[1] https://www.booking.com/

[2] https://www.amazon.com/Best-SellersAppstoreAndroid/zgbs/mobile-apps/ref=zg_bs_nav_0

[3] https://nlp.stanford.edu/projects/glove/

[4] http://corpus.tools/wiki/Unitok

Methods	Acc	F1-Score	
		Micro	Macro
word2vec			
SVM	63.0	65.4	40.3
Naive Bayes	62.6	64.8	36.1
Bi-GRU (with weight)	61.2	62.2	41.2
Bi-GRUATT (no weight)	64.0	66.3	35.1
Bi-GRUATT (with weight)	**65.0**	**67.3**	**44.1**
GloVe			
Bi-GRUATT (no weight)	**71.0**	73.0	47.0
Bi-GRUATT (with weight)	64.0	66.1	47.7
word2vec+GloVe			
Bi-GRUATT (no weight)	**71.0**	**73.2**	48.8
Bi-GRUATT (with weight)	68.6	70.7	**49.9**

Table 2: Comparative results of methods with different word embedding.

3.3 Results

We validated the performance based on the development dataset, and used the same model weight on the test dataset to output the test results. For this shared task, the micro-average (Micro) and macro-average (Macro) were used in the evaluation along with the accuracy (Acc). The results of the baseline and our proposed models based on the word2vec embedding are shown in Table 2.

We found that using the combination of a unigam and bi-gram performed better than only a unigram or bi-gram individually for both the SVM and Naïve Bayes models. The attention mechanism enhanced 3-5% for the three evaluation indicators, and showed remarkable improvement with the parameter sample weight for the macro F1-score.

Different word-embedding models were employed in our experiment, the comparative results of which are also presented in Table 2. GloVe showed a clearly better performance over word2vec embedding, with a 7% improvement in accuracy owing to the training on larger corpora. Moreover, using the parameter sample weight to balance the training data, the model was trained to be biased toward small sample classes. As a result, the macro F1-score of Bi-GRUATT with the sample weight increased, whereas the micro F1-score decreased. And the combination of word2vec and glove achieved the best performance.

The model using word2vec and GloVe showed different performance on different tags. For example, as shown in Table 3, compared to Bi-GRUATT with weight of word2vec, F1-score of tags complaint, bug, meaningless increased 3-8% in the model with GloVe, however, decreased

Tags	Bi-GRUATT(with weight)		
	word2vec	GloVe	word2vec+GloVe
comment	79.0	76.0	80.5
complaint	61.5	64.8	63.0
bug	30.8	38.5	38.5
meaningless	48.5	54.4	54.5
request	42.9	37.5	57.1
undetermined	-1	-1	-1

Table 3: Comparative F1-score of each tag with different word embedding.

0.05-3% in tags comment and request. By combining word2vec and GloVe together, we not only got the higher score of their each tag, but also advanced the score of each tag.

The task attracted a total of 139 submissions of four languages from 12 teams. Our Bi-GRUATT model with a combination of word2vec and GloVe achieved the best result in terms of accuracy, and ranked sixth in micro F1-score, third in macro F1-score out of 56 submissions for the English language.

4 Conclusion and Future Work

In this paper, we presented our implemented solutions to IJCNLP task 4, with the goal of classifying six classes for customer feedback sentences of English. We promoted a bi-directional GRU with an attention mechanism, and presented two other baseline methods (SVM and Naïve Bayes) for comparison. The experiment results showed an improvement of around 8% when using the Bi-GRUATT model with GloVe and word2vec relative to the baseline methods; in addition, a sample weight parameter for imbalanced data achieved a good macro-average score, but with a decline in accuracy and micro-average score.

As future work, we will attempt a multi-label classification of this task, and test the performance of our model in other languages, such as Spanish, French, and Japanese.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No.61702443 and No.61762091, and in part by Educational Commission of Yunnan Province of China under Grant No.2017ZZX030. The authors would like to thank the anonymous reviewers and the area chairs for their constructive comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3:1137–1155.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *SSST@EMNLP*, pages 103–111. Association for Computational Linguistics.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR*, abs/1412.3555.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Robert Desimone and John Duncan. 1995. Neural Mechanisms of Selective Visual Attention. *Annual Review of Neuroscience*, 18(1):193–222.

G. E. Hinton and R. R. Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507.

Sepp Hochreiter and J Urgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *ACL (1)*, pages 873–882. The Association for Computer Linguistics.

Ozan Irsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *EMNLP*, pages 720–728. Association for Computational Linguistics.

Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. In *Interspeech*, pages 1045–1048.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *EMNLP*, pages 2249–2255. The Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Colin Raffel and Daniel P. W. Ellis. 2015. Feed-forward networks with attention can solve some long-term memory problems. *CoRR*, abs/1512.08756.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *HLT-NAACL*, pages 1480–1489. The Association for Computational Linguistics.