SentiNLP at IJCNLP-2017 Task 4: Customer Feedback Analysis Using a Bi-LSTM-CNN Model

Shuying Lin[1,3,4], Huosheng Xie[1] Liang-Chih Yu[2,3] and K. Robert Lai[3,4]
[1]College of Mathematics and Computer Science, Fuzhou University, Fuzhou, P.R. China
[2]Department of Information Management, Yuan Ze University, Taiwan
[3]Innovation Center for Big Data and Digital Convergence, Yuan Ze University, Taiwan
[4]Department of Computer Science & Engineering, Yuan Ze University, Taiwan
Contact: lcyu@saturn.yzu.edu.tw

Abstract

Analysis of customer feedback helps improve customer service. Much online customer feedback takes the form of online reviews, but the tremendous volume of such data makes manual classification impractical, raising the need for automatic classification to allow analysis systems to identify meanings or intentions expressed by customers. The aim of shared Task 4 of IJCNLP 2017 is to classify customer feedback into six tags. We present a system that uses word embeddings to express features of the sentence in the corpus, using the neural network as the classifier to complete the shared task. The ensemble method is then used to obtain a final predictive result. The proposed method ranked first among twelve submissions in terms of micro-averaged F1 and second for accuracy.

1 Introduction

Software companies receive tremendous amounts of online customer feedback every day, including product comments, bug reports, new feature requests, response complaints, capacity concerns and so on. The effective classification of this customer feedback can provide a foundation for improved customer service, but the huge amounts of data make manual classification impractical, raising the need for automatic classification to accurately identify customer meanings or intentions.

Sentence classification is a fundamental task for natural language processing (Collobert et al., 2011). The goal of this shared task is to classify the cross language customer feedback into six categories (comment, request, bug, complaint, meaningless, and undetermined). Each sentence will be assigned at least one tag. It can be treated as a multi-label classification problem.

In recent years, deep neural network models such as convolutional neural networks (CNN) (Cun et al., 1990), recurrent neural networks (RNN) (Goller and Kuchler, 1996), long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997), and their combinations (Wang et al., 2016) have achieved remarkable results in many NLP tasks, including sentence classification (Kim, 2014; Kalchbrenner et al., 2014), sentiment analysis (Irsoy and Cardie, 2014; Liu et al., 2015), sarcasm detection (Ghosh and Veale, 2016; Amir et al., 2016). The neural network models can automatically infer the features and can be used as a sentence classifier. The word embeddings (Mikolov et al., 2013a; 2013b; Pennington et al, 214; Yu et al., 2017) can provide word vector representation that captures semantic and syntactic information of words. The word vector is used to build the sentence matrix and then inject information into sentence classifier. The LSTM can provide the sentence sequence information in one direction. Forward and backward networks respectively capture past and future information. Therefore, we used the bi-directional LSTM for our model.

This paper presents a system to classify English customer feedback into six labels (comment, request, bug, complaint, meaningless, and undetermined). The system uses the word vector of the sentence as input, and the neural network as the classifier. The neural network uses the sentence matrix as input to classify the sentence into six labels. For multi-label classification, we construct a binary classifier for each label. The proposed model for sentence classification consists of two parts: a bi-directional LSTM and CNN. The bi-directional LSTM is used to capture the context of a sentence through sequential modeling. The sequence feature is used as the input for the CNN layer, which is then used to extract the most important feature to form the sentence representation. The logistic regression layer on the top is used to output the sentence labels.

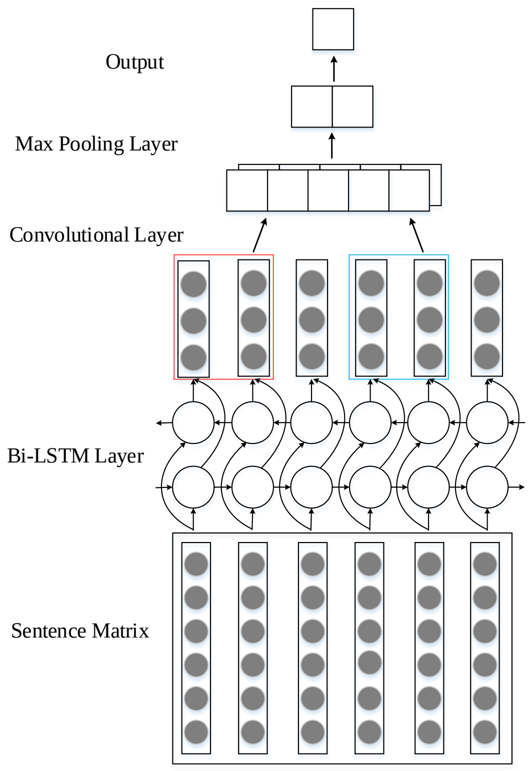

Figure 1: System architecture of the proposed bi-LSTM -CNN model.

The rest of this paper is organized as follows. Section 2 describes the model for sentence classification. Section 3 briefly introduces the ensemble method used in this paper. Section 4 reports experimental results and analysis. Section 5 presents conclusions and suggests directions for future work.

2 Model for Sentence Classification

The model aims to classify a sentence into six labels according to its sentence texts. Figure 1 shows the framework of the bi-directional LSTM-CNN model for sentence classification. In the input layer, the sentence is transformed into a sentence matrix based on the word vector. The word vectors of vocabulary words are trained from a large corpus using the Glove (Pennington et al., 2014) toolkit. The sentence matrix is fed into a forward LSTM network and a backward LSTM network. The representation sequence is then averaged over all timesteps and then concatenated to produce the final sequence representation. The sequence is the input of the CNN layer. The CNN extract the sequence feature information followed by logistic regression layer whose target is the class label for given sentence.

For a given corpus, we store the word vector in a look up matrix $\mathrm{M} \in \mathrm{R}^{d \times |V|}$, where |V| is the vocabulary size of the given texts and d is the dimensionality of the word vector. For the sentence $S = \{s_1, s_2, \dots, s_n\}$, n is the length of the sentence. Let |V| denote the vocabulary of words, while d is the dimensionality of word vectors. The sentence matrix representation $\mathrm{X}=\{x_1, x_2, ..., x_\mathrm{n}\}$, x_i is the word vector of word s_i in accordance with the look up matrix M .

2.1 Recurrent Neural Network

To capture the relationship between the input sequences, we can use the RNN layer to transform word vector into the sentence feature representation.

Due to the vanishing gradients and exploding gradients problem (Pascanu et al., 2012), the LSTM (a certain type of RNN) is introduced for sequence modeling (Tang et al., 2015; Tai et al., 2015). The LSTM consists of three gates: input gate *i*, forget gate *f* and output gate *o*. The gate mechanisms work collaboratively to learn the long-term dependencies. At each time step t, the LSTM is calculated as follows:

$$\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
g_t &= \tanh(W_g x_t + U_g h_{t-1} + b_g) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t * c_{t-1} + i_t * g_t \\
h_t &= o_t * \tanh(c_t)
\end{aligned} \tag{1}$$

where x_t is the input from lower layer. h_{t-1} is the hidden state at time step *t-1*. The i_t , f_t , o_t are respectively called the *input*, *forget* and *output* gates. $W \in R^{\omega \times d}$ and $U \in R^{\omega \times d}$ are the weight matrices for different gates for input x_t and hidden state h_{t-1}. *b* denotes bias vectors. Here * is the element-wise multiplication, and the $\sigma(\cdot)$ and $\tanh(\cdot)$ are the element-wise activation function. The d can be the dimension of the word vector or the size of the hidden state in the lower layer.

A forward network and a backward network can respectively capture the past and future information. For sentence classification, it is beneficial to combine the past and future information in sequence modeling. Therefore, we use bi-directional LSTM to obtain the contextual information in sentence sequence modeling.

2.2 Convolutional Neural Network

The convolutional is used to extract n-gram features. We use a convolution filter $\mathrm{F} \in R^{\omega \times d}$ to obtain the feature map $\mathrm{Y} \in \mathrm{R}^{n-\omega+1}$, The *j*-th element y_j is given by:

$$y_j = f\left(W \bullet x_{j:j+\omega\text{-}1} + b\right) \tag{2}$$

where *f* is the Relu activation function, *W* is the weight matrix of convolution filter, *b* is a bias, *ω* is the length of the filter, and *d* is the dimension of the word vector. The convolutional layer uses multiple filters in parallel to obtain feature maps. It also can use convolutional filters of various length to extract different feature information.

The feature maps are fed into the max-pooling layer to obtain the most salient information to form the feature vector. The feature vector is useful for determining the output result.

3 Ensemble method

In statistics, ensemble methods use multiple learning algorithms to obtain better predictive performance (Maclin and Opitz, 1999; Rokach, 2010). In this paper, the ensemble methods use multiple results produced by the same NN-based method in different training times to obtain better predictive performance. For multi-label classification, we make a variation of the voting rules in the ensemble method. The predictive outputs can be calculated in two steps: For each label, we assign the sentence to label *i* if the proportion of positive results in more than half of the total predictive result. After that, we assign it to label *j* with the most component predictions if the sentence was unannotated.

Class	Training set	Development set
Comment	1758	276
Complaint	950	146
Meaningless	306	48
Request	103	19
Bug	72	20
Undetermined	22	3

Table 1: Data distributions of sentence labels in customer feedback: comment, complaint, meaningless, request, bug, and undetermined

4 Experiments and Evaluation

Dataset. For the shared task on customer feedback analysis, several sentences of annotated and unannotated customer feedback in English were prepared, with a total of 3065 training texts, 500 development texts and 500 test texts. The training and development texts were annotated with six labels (comment, request, bug, complaint, meaningless, and undetermined). We evaluated the proposed bidirectional LSTM-CNN model by submitting the results of the test set to the IJCNLP 2017 Task 4 Customer Feedback Analysis. The distribution of the six labels shown in Table 1 indicates a data imbalance. Most of the data were assigned one of five class labels, and only a few were annotated as being undetermined. There are many pre-trained word vectors for English provided by word embeddings tool. We use the pre-trained word vectors trained on 840B tokens from common crawls and its dimensionality is 300 provided by Glove (Pennington et al., 2014) because of the high volume of the vocabulary. Word vectors are randomly initialized with uniform distribution sample if the word is not in the vocabulary of the pre-trained word vectors.

	Micro- F1	Acc
Scores	0.7557	0.708
Rank	1	2

Table 2: Results of ensemble method for the best of the five results produced by the neural network model for IJCNLP-2017 Task 4.

Let Y be the vector representation of the label of the sentence. The labels *ls*={ l_1, l_2, l_3, l_4, l_5, l_6 }and the sentence has multiple labels {l_1, l_2}. For each label, we use a binary classifier. Therefore, we can express the labels as: Y=(1, 1, 0, 0, 0, 0) to calculate the loss function through binary cross-entropy.

Experimental settings. The dataset containing the development set and training set is randomly shuffled and then re-divided for the 5-fold cross validation. The hyper-parameters of the neural network are chosen based on the performance on the development data. We obtain the final hyper-parameters by averaging the ten evaluation results of the development set in the 5-fold cross validation.

The epoch time is 10 to minimize the loss function. To avoid overfitting, we use the early stop mechanism and random dropout (rate of 0.25 or 0.5) (Srivastava et al., 2014). The nadam (Dozat, 2016) update rule is used to automatically tune the learning rate. The activation function in the top layer is a sigmoid function, which scales each label within a range 0 to 1. If the continuous result y_i is greater than 0.5, it is rounded up to 1, or set to 0 otherwise. The predicted result Y=(0, 0, 0, 0, 0, 0) for six labels is assigned to label i with the maximum value y_i of Y.

We first run experiments on CNN, LSTM and their combinations, including LSTM-CNN, CNN-LSTM, bi-LSTM and bi-LSTM-CNN. CNN1 and CNN2 have different hyper-parameters, a filter window of 3 with 256 feature maps in CNN1, and a filter window of 3 and 5 with 128 feature maps in CNN2.

We compare the results generated by the NN-based methods and select the best five results on the development set data in micro-averaged F1 to produce the ensemble result.

	NN-based		Ensemble	
	Micro-F1	Acc	Micro-F1	Acc
CNN1	0.7223	0.668	0.7342	0.682
CNN2	0.7383	0.694	0.7495	0.702
LSTM	0.5947	0.57	0.6425	0.618
LSTM-CNN	0.6931	0.658	0.7181	0.68
CNN-LSTM	0.7155	0.68	0.7332	0.69
Bi-LSTM	0.7417	0.7	0.7455	0.704
Bi-LSTM-CNN	0.7309	0.69	0.7557	0.708

Table 3: Comparative results of ensemble method with all NN-based methods for sentence classification.

Evaluation metrics. IJCNLP 2017 Task 4 published the results for all participants assessed based on both accuracy and micro-averaged F1 measure. Given a binary classification, there are four basic outcomes: true positive (tp), true negative (tn), false positives (fp) and false negative (fn). The accuracy and F1 score (Powers, 2011) are evaluation measures B(tp, tn, fp, fn) used to evaluate the performance of a binary classification problem. Accuracy is the proportion of true results (both tp and tn) among the total test set. The micro-averaged F1 has a binary evaluation for its overall counts among all labels.

Results. A total of twelve teams participated in task 4. Table 2 shows the result of the ensemble method with the best of the five results produced by the bi-directional LSTM-CNN model. We obtain the best result by sorting the micro-averaged F1 for all labels. The ensemble result ranked first for micro-averaged F1 and second for accuracy.

Table 3 shows the best experimental results of ten runs for the neural network model for both accuracy and micro-averaged F1, along with rive runs for the ensemble method. We found that the ensemble method shows better performance, indicating that it can improve on all NN-based methods. The ensemble result in bi-LSTM-CNN achieves the best performance in all NN-based methods. The bi-LSTM yielded better performance without ensemble.

5 Conclusions

This study presents a neural network model to classify text-based customer feedback into six labels. We use the ensemble method to obtain the best of five results sorted by the micro-averaged F1. The use of the ensemble method can further improve performance in all NN-based methods. The bi-directional LSTM produces the sentence sequence feature, and the convolutional layer extracts the salient information from the sequence representation to classify the sentence into the multi-labels. Experimental results show that the bi-directional LSTM-CNN achieves best performance.

Future work will focus on exploring customer feedback in multiple languages and high-order correlations between labels should be taken into account to improve classification performance for both micro- and macro-averaged F1.

Acknowledgments

This work was supported by the Ministry of Science and Technology, Taiwan, ROC, under Grant No. MOST 105-2221-E-155-059-MY2 and MOST 105-2218-E-006-028.

References

Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mario J. Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. In *Proc. of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 167–177.

Ronan Collobert, Jason Weston, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(1):2493–2537.

Y. Le Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. 1990. Handwritten digit recognition with a backpropagation network. *In Advances in Neural Information Processing Systems*, pages 396–404.

Timothy Dozat. 2016. Incorporating nesterov momentum into adam. In *ICLR 2016 workshop*, page 107.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proc. of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA-16)*, pages 161-169.

C. Goller and A. Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Proc. of the IEEE International Conference on Neural Networks (ICNN-96)*, pages 347–352.

Sepp Hochreiter and J¨urgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ozan Irsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP-14)*, pages 720–728.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *Eprint Arxiv*, 1.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Eprint Arxiv.*

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP-15)*, pages 1433–1443.

R. Maclin and D. Opitz. 1999. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *Computer Science*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. On the difficulty of training recurrent neural networks. *Computer Science*, 52(3):III–1310.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP-14)*, pages 1532– 1543.

David M W Powers. 2011. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2:2229–3981.

Lior Rokach. 2010. *Ensemble-based classifiers.* Kluwer cademic Publishers.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL/IJCNLP-15)*, pages 1556–1566.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP-15)*, pages 1422–1432.

Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. 2016. Dimensional sentiment analysis using a regional CNN-LSTM model. In *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics (ACL-16)*, pages 225-230.

Liang-Chih Yu, Jin Wang, K. Robert Lai, and Xue-jie Zhang. 2017. Refining word embeddings for sentiment analysis. In *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP-17)*, pages 545–550.