YNU_DYX at SemEval-2019 Task 9: A Stacked BiLSTM Model for Suggestion Mining Classification

Yunxia Ding, Xiaobing Zhou*, Xuejie Zhang
School of Information Science and Engineering
Yunnan University, Yunnan, P.R. China
*Corresponding author:zhouxb@ynu.edu.cn, ynudyx@gmail.com

Abstract

In this paper we describe a deep-learning system that competed as SemEval 2019 Task 9-SubTask A: Suggestion Mining from Online Reviews and Forums. We use Word2Vec to learn the distributed representations from sentences. This system is composed of a Stacked Bidirectional Long-Short Memory Network (SBiLSTM) for enriching word representations before and after the sequence relationship with context. We perform an ensemble to improve the effectiveness of our model. Our official submission results achieve an F_1-score 0.5659.

1 Introduction

Suggestions in the Oxford Dictionary are defined as ideas or plans for consideration. Some of the listed synonyms of suggestions are proposal, proposition, recommendation, advice, hint, tip, clue. In general, other types of text and suggestions are easily distinguished by the definition of the suggestion (Negi and Buitelaar, 2015).

Suggestion mining can be defined as the extraction of suggestions from unstructured text, where the term 'suggestions' refers to the expressions of tips, advice, recommendations etc. We often see comments on products in product forums which are recommended or not recommended, and some users will consider whether to purchase the product based on these comments. Suggestion mining is also defined as automatic extraction of recommendations from a given text. These texts that express user suggestions can usually be found in social media platforms, blogs, or product online forums (Negi and Buitelaar, 2017; Negi et al., 2016).

Suggestion mining remains a relatively young area compared to Sentiment Analysis (Negi and Buitelaar, 2017), due to the lack of a large number of tagged datasets. SemEval 2019 Task 9-SubTask A is mainly a binary classification, identifying sentences which express suggestions in a given text. And we need to classify each sentence of given text, the categories being suggestions or non suggestions. This is similar to the polarity analysis of emotions, as positive or negative instances, respectively.

In the past, classification problems in natural language processing were solved by traditional methods, such as sentiment analysis (Nielsen, 2011; Go et al., 2009; Bollen et al., 2011; Mohammad et al., 2013; Kiritchenko et al., 2014) which were handled by classifiers such as Naive Bayes (McCallum et al., 1998)and SVMs (Gunn et al., 1998). However, deep neural networks achieve increasing performance compared to traditional methods, due to their ability to learn more abstract features from large amounts of data, producing state-of-the-art results in sentiment analysis.

The SubTask-A is part of SemEval 2019 Task9: Suggestion Mining from Online Reviews and Forums, and is concerned with classifying suggestions forum for Windows platform developers—suggestions or non suggestions. There are 34 teams who participated in the task(Negi et al., 2019).

In this paper we describe our system designed for this task. First, we model the sentence and establish the vector representation of the sentence through Word2Vec (Mikolov et al., 2013a), a Stacked Bidirectional Long-Short Memory Network(SBiLSTM) for enriching word representations with context. Finally, the sentence representation is projected into the label space through a Dense Layer.

The rest of the paper is organized as follows: Section 2 provides the details of the proposed model; Data Processing and analysis are discussed in section 3. Experiments and results are described in Section 4. Finally, we draw conclusions in Sec-

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1272–1276
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

tion 5.

2 System Description

2.1 Network Architecture

First we use the embedding layer to get a distributed representation of the words, then feed the results of the embedding layer to the first BiLSTM layer. Using the LSTM model (Hochreiter and Schmidhuber, 1997) can better capture long-distance dependencies and learn what information to remember and what information to forget by training the model. BiLSTM captures the semantic information of sentences from both forward and reverse directions. In order to get more fine-grained sentence information, we use 2-layer BiLSTM. The features obtained from the first BiLSTM are then put into the next BiLSTM (Graves and Schmidhuber, 2005; Graves et al., 2013). The final result is obtained by the softmax used as the activation function in the Dense layer. The model architecture is show in Figure 1.

2.2 Word Embedding

Word embedding is unarguably the most widely known technology in the recent history of NLP. It converts words into a distributed representation that can solve dimensional disaster problems (Bengio et al., 2003). And it projects words from high-dimensional space to a lower-dimensional vector space through hidden layers and performs semantic feature extraction (Kim, 2014). This technology has a wide range of applications in NLP. It is well-known that using pre-trained embedding helps, as well.

Word embeddings can better measure the similarity between words, and are also dense vector representation of words that capture semantic and syntactic information. So in this task we try to use the Word2Vec (Mikolov et al., 2013b) and Glove (Pennington et al., 2014) to get the vector representations of the words.

3 Data Processing and Analysis

There are two categories: suggestion and non-suggestion in the data set given in the shared task. And the organizer provides the third version data sets: a total of 8,500 sentences in training and 592 sentences in trial and 833 sentences in test.

3.1 Data Processing

We perform a series of specification processing on the text in the dataset.

- All characters are converted to lowercase.
- Contraction normalization, like replacing “don’t” and “dont” with “do not”, “cant” and “can’t” with “can not” and so on.
- All hyperlinks are replaced by “url”.

After the above processing, we find that some words in the text have not been segmented correctly. For example, the correct form of “supportedcultures” should be “supported cultures”. There are many such words in the dataset, and if we don’t deal with them, there will be a lot of unknown words in the vocabulary. In order to solve this problem, we use *Ekphrasis* (Baziotis et al., 2017), a tool geared towards text from social networks, such as Twitter or Facebook. *Ekphrasis* performs tokenization, word normalization, word segmentation (for splitting hashtags) and spell correction.

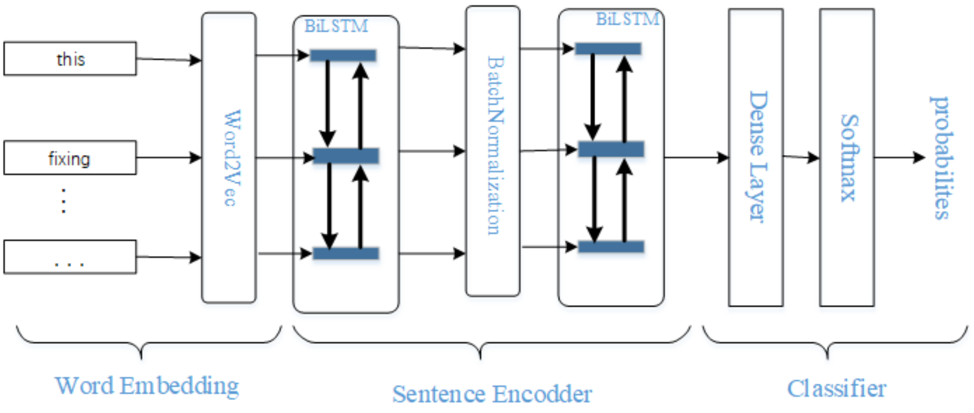

Figure 1: Our system architecture

3.2 Data Analysis

Sentence length: In order to determine the length of the training set sentence in the input model, after the data processing is finished, we analyze the length of each sentence. First, we find that the longest sentence is 495, the shortest is 0, and the sentence length is shown in Figure 2.

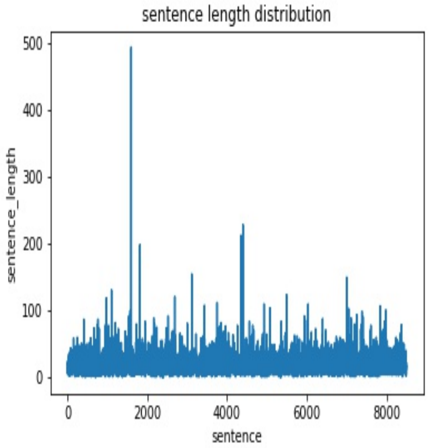

Figure 2: Training set sentence length distribution

If the sentence is too long, the calculation time will increase. If it is too short, the extra information will be lost. Therefore, according to the sentence length distribution map, the length of the sentence in the input model is finally determined to be 75.

Training set label: Table 1 shows the label distribution for the dataset.

	Train set	Trial set	Test set
Suggestions	2085	296	87
Non-suggestions	6415	296	746

Table 1: Number of sentences in each dataset.

It can be seen from Table 1 that the label of the training set is extremely unbalanced, and the ratio of suggestion and non-suggestion reaches 1 : 3. In order to balance the training set data, we process those sentences labeled with the suggestions. We use the shuffle data enhancement method, which re-range the word order inside the sentences. We performed two shuffle operations, and the last data in the suggestions and non-suggestions in the final dataset were 6255 and 6415, respectively.

4 Experiments and Results

We use Python based neural network library, Keras[1], for the implementation. We train and validate our models on the training and validation sets provided by the organizer. The official evaluation metric is based on macro average F_1-*score* measure. More details about the data and the evaluation metrics can be found in the task description paper (Negi et al., 2019).

SBiLSTM: For the Stacked BiLSTM, the first layer BiLSTM *units* = 256, and the second layer BiLSTM *units* = 180.

Optimization: Optimization is carried out with Adaptive Moment Estimation(Adam) (Kingma and Ba, 2014), using the default learning rate 0.001, and hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$.

Loss Function: Usually the multi-classification problem uses *categorical crossentropy* as the loss function. But our system uses *binary crossentropy* in this binary classification.

As shown in Table 2, the number of unknown words in the dataset in Word2Vec are less than those in Glove. To reduce the number of unknown words in the embedding, making the context semantics better learned by the model. We randomly assign the vectors of unknown words. And we experimented with the embedding of the words Word2Vec and Glove, and found that the results of Word2Vec performed better than Glove.

Embedding	Ukw	Evaluation set F1
Glove	394	0.5422
Word2Vec	231	**0.5482**

Table 2: Comparison between Word2Vec and Glove on DBiLSTM models.

We compare the two network structures of Stacked LSTM and Stacked BiLSTM. As can be seen from the results in Table 3, the performance of the Stacked BiLSTM is better than that of LSTM.

Model	Embedding	Evaluation set F1
LSTM	Word2Vec	0.5612
BiLSTM	Word2Vec	**0.5637**

Table 3: Comparison of LSTM and BiLSTM.

Finally, we train the single model with the dropouts of 0.55, 0.60, 0.65, respectively. Each

[1] http://keras.io/

single model produces a soft probability, then we use the sum of the probabilities as the final prediction. We find that the performance of ensemble model is better than a single model.

Dropout	Evaluation set F1
0.55	0.5307
0.60	0.5214
0.65	0.5422
Ensemble	**0.5659**

Table 4: The model adopts Word2Vec, data enhancements and Stacked BiLSTM architecture. Dropout is recurrent-dropout in the BiLSTM layer.

5 Conclusion and Future Work

In this paper, we have presented a Stacked BiLSTM(SBLSTM) model for predicting the suggestion mining classification. The word embedding Word2Vec is used in our system, an ensemble method can significantly enhance the overall performance.

In the future, we will try to use language models to obtain the representation of sentences, and explore other NLP models to make the experimental results better. At the same time, we will also try to use transfer learning technology.

Acknowledgments

This work was supported by the Natural Science Foundations of China under Grant Nos. 61463050, 61702443 and 61762091, and the Project of Innovative Research Team of Yunnan Province under Grant No. 2018HC019.

References

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Johan Bollen, Huina Mao, and Alberto Pepe. 2011. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *Icwsm*, 11:450–453.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12).

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Steve R Gunn et al. 1998. Support vector machines for classification and regression. *ISIS technical report*, 14(1):5–16.

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Sapna Negi, Kartik Asooja, Shubham Mehrotra, and Paul Buitelaar. 2016. A study of suggestions in opinionated texts and their automatic detection. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 170–178.

Sapna Negi and Paul Buitelaar. 2015. Towards the extraction of customer-to-customer suggestions from reviews. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2159–2167.

Sapna Negi and Paul Buitelaar. 2017. Inducing distant supervision in suggestion mining through part-of-speech embeddings. *arXiv preprint arXiv:1709.07403*.

Sapna Negi, Tobias Daudert, and Paul Buitelaar. 2019. Semeval-2019 task 9: Suggestion mining from online reviews and forums. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*.

Finn Årup Nielsen. 2011. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.