A Topic BiLSTM Model for Sentiment Classification

Yanming Huang[1,2], Y.Jiang[1]
[1]School of Computer Science South China Normal University
Guangzhou, China
ym.huang@m.scnu.edu.cn,
ycjiang@scnu.edu.cn

Touhidul Hasan[2], Q. Jiang[*2]
[2]Shenzhen Institutes of AdvancedTechnolgy, CAS
Shenzhen, China
touhidul.hasan@siat.ac.cn
qs.jiang@siat.ac.cn

Chao Li [2,3]
[3]Shandong University of Science and Technology,
Qingdao, China
1008lichao@163.com

ABSTRACT
The Long Short Term Memory (LSTM) network is very effective for capturing sequence information which can help to analyze sentiments. However, it fails to capture the meaning of polysemous word under different contexts. In this paper, we propose topic information-based bidirectional LSTM (BiLSTM) model for sentiment classification. BiLSTM model learns topic information to obtain the sensitive representation of the polysemous word under given circumstance. The topic information is generated through a topic modeling via Latent Dirichlet Allocation (LDA). The topic information-based BiLSTM network allows the model to capture the meaning of the polysemous word and long sequence information automatically. The experimental results on real-world datasets demonstrate that the proposed method outperforms the task of benchmark sentiment classification on SemEval 2013 and IMDB.

CCS Concepts
• **Computing methodologies→Artificial intelligence→Natural language processing→Information extraction**

Keywords
sentiment classification; bidirectional LSTM; LDA

1. INTRODUCTION
Sentiment classification aims to classify the sentiment polarity of a sentence as positive, negative, and neutral, where it receives much attention in natural language processing (NLP) and has many applications such as products recommendation from customer reviews and public opinion analysis [1, 2]. Most existing works on sentiment classification focus on feature engineering and designing a variety of features, for instance, bag-of-words, PMI unigram lexicons, and ngrams [3, 4]. Recently, word embedding and neural network-based models such as recurrent neural networks (RNN) and long short term memory (LSTM) become popular for sentiment classification and these methods extract individual features automatically [5, 6, 7].

LSTM is a specially designed memory cell from RNN, and it copies the state between time steps in a non-linear fashion. The LSTM memory cell is good at capturing the long distance information. However, it only captures the single embedding features [8]. Though the context-sensitive feature is helpful to distinguish the sentiment of the sentence, still LSTM cannot determine the semantic of the polysemous word. For example, two tweets from SemEval 2013 [9] dataset with word *offensive* are given below:

•Brook Lopez the 2nd best *offensive* center in the NBA he not ass. **(POSITIVE)**

•Monday before I leave Singapore, I am going to post something that might be *offensive*. **(NEGATIVE)**

the word *offensive* is used as a positive word in the first tweet, but as a negative in the second tweet.

In this paper, we propose the topic information-based BiLSTM model which capture long-range sequence and learn topic-enriched information to represent the sentiment of the polysemous words. The topic information is learned by the LDA model [10]. Unlike [11], we obtain the topic distribution as the semantic feature to join the BiLSTM network. We apply Sentiment-Specific Word Embedding (SSWE) to capture the sentiment information of a text [8, 7]. The similarity matrix is constructed from the training set to solve the data imbalance problem, and it learns the training and new data jointly with shared representation which allows our model to obtain a better generalization. The convolutional neural network learns word vector using w2v, and glove [6, 7]. Finally, SSWE is used to encodes the sentiment information of the text into the continuous representation of words.

The remainder of this paper is structured as follows. We review related works in Section 2. In Section 3, we present the details of the topic information-based BiLSTM network. The experimental results are discussed in Section 4, and the paper is concluded in Section 5.

2. RELATED WORK
Traditional methods focus on two topics: lexicon-based approaches and corpus-based methods. The typical work of former [12, 13, 14] use some base dictionaries and pre-existing linguist resources. The latter focuses on hand-crafted discriminative features and combining them to make a classifier. Ngram features and Support Vector Machine are widely used to make the sentiment classification [1]. The work in [9] uses diverse sentiment lexicons and manual-designed feature, building the top performance in the Twitter sentiment classification track of SemEval 2013. However, ngram features fail to capture the long-distance information. Increasing the context-window size easily leads to model over-fitting. Manually designed features are expensive and time-consuming.

ICIAI 2018, March 9–12, 2018, Shanghai, China

ACM ISBN 978-1-4503-6345-7/18/03…$15.00
DOI: https://doi.org/10.1145/3194206.3194240

Deep learning approaches can learn automatic features, and it achieves state-of-the-art performances [15]. Semi-supervised recursive autoencoder is introduced for predicting sentiment distributions without using any pre-defined sentiment lexicon or

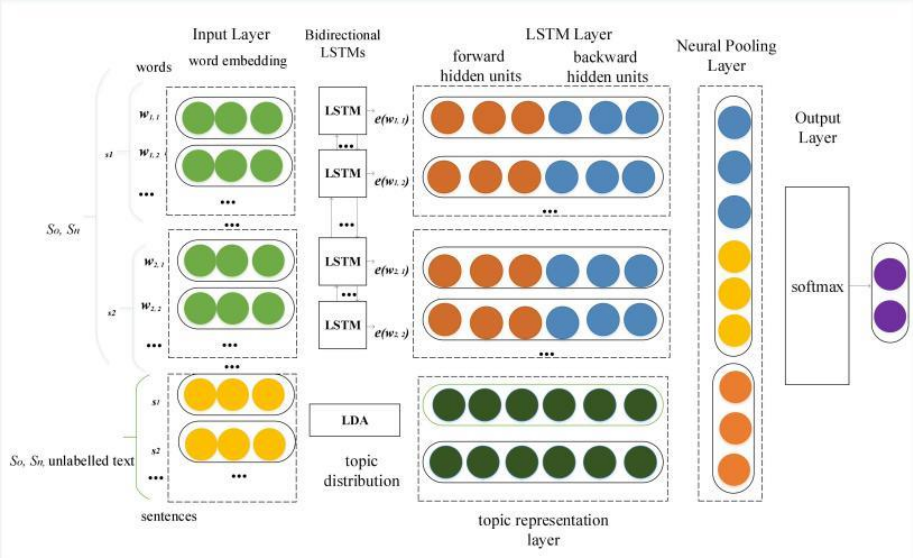

Figure 1. System architecture of the proposed model

polarity shifting rules [16]. The work in [17] develops neural networks to learn word vectors from tweets containing positive and negative emotions. A tree long short-term memory (LSTM) is developed for improving semantic representations [8]. However, LSTM fails to represent the sentiment of the polysemous words.

3. TOPIC BILSTM MODEL

An overview of the proposed model is shown in Figure 1. We first design the methodology that generates the new data, and together it represents as a word sequence. We follow the sentiment specific word embedding to determine word vectors of vocabulary words in the training data [17]. The hidden unit is introduced to get the high-level representation of forward and backward LSTMs. The topic-enrich information learns from LDA in topic representation layer. Long sequence and topic features are concatenated in LSTM layer, where neural pooling layer extracts features, and predict in softmax layer.

3.1 Input Layer

The input layer of the network is mapped words into real-valued vectors for processing by subsequent layers. However, the collected dataset is an imbalance. The model trained from the dataset that has low precision in minority class and overfitting in majority class. As described in the previous strategy, we design a way that has balance data and allows the entire network to learn the shared representation.

Let the new text S_n together with original text S_o to get balance dataset $S = \{S_n, S_o\}$. For each given text S_{io}, we replace several words randomly with words from similarity matrix and obtain the new text S_{in}. We generate a random number for each word in the given training data, and the word is chosen to replace the original text S_{io}. The Cosine distance (threshold 0.8) is used to construct the similarity matrix $D \in R^{|V|\times|V|}$, where V is the size of the vocabulary.

Each sentence represents as $S_i = \{w_{i1}, w_{i2}, \cdots, w_{ij}\}$, where w_{ij} is the j^{th} word of the sentence i. We map words into the input layer sequence $\{e(w_{i1}), e(w_{i2}), \cdots e(w_{ij})\}$, where e is the embedding of the word. We use matrix $E^{d\times|V|}$ to present word vectors, where d is the dimension of the word. We define the maximum length of the text as N. If the input length short than N, we append several random vectors with a uniform distribution $U(-0.01, 0.01)$.

3.2 Topic Representation Layer

The topic representation layer learns the topic distribution of the text. A document is a mixture of the topic information, and each document has a polynomial distribution of topics. We add unlabeled text to create the corpus, and Dirichlet prior Dir(α) is introduced to train topic distribution. For each topic, there is another multinomial distribution over words. Therefore, we apply Gibbs sampling for LDA model parameter estimation.

Concretely, each text is regarded as one document. Suppose there are T topics i.e., $t_1, t_2, \cdots, t_T$. The posterior probability of each topic given text s_i is computed as:

$$P_t(t_j|s_i) = \frac{C_{ij} + \alpha_j}{\sum_{k=1}^{T} C_{ik} + T\alpha_j} \quad (1)$$

where C_{ij} is the number of times that topic t_j is assigned to some word in text s_i, usually, it averaged over multiple iterations of Gibbs sampling. α_j is the j^{th} dimension of the hyper parameter of Dirichlet distribution that can be optimized during model estimation. To generate a better topic model, we multiply a random function with discrete uniform distribution. The new topic distribution as semantic feature concatenates to the final hidden state of the LSTM layers.

3.3 LSTM Layer

LSTM layer captures long sequence information. BiLSTM is an adaptation of the LSTM which incorporates a forward and backward LSTM layer to learn information from the previous layer. In Figure 1, one LSTM processes the sequence from left to right and the other from right to left. At each time step t, a hidden

forward network layer with hidden unit functions $\vec{h}$ are computed based on previous hidden unit $\overrightarrow{h_{t-1}}$.The input at the current step $e(w_t)$ and a hidden backward layer with hidden unit function $\overleftarrow{h}$ is computed based on hidden unit $\overleftarrow{h_{t+1}}$, and the current step $e(w_t)$. The dropout strategy is utilized to reduce overfitting which prevents complex co-adaptation on the training data. The forward hidden node sequence $\{\overrightarrow{h_1^1}, \cdots \overrightarrow{h_n^1}\}$ from the forward LSTM network and backward hidden node sequence $\{\overleftarrow{h_1^1}, \cdots \overleftarrow{h_n^1}\}$ from the backward LSTM network are concatenated both forward and backward hidden sequence to obtain a new hidden node sequence h^1.The topic distribution learns from topic representation layer, and it concatenates on the new hidden node sequence.

Each time step cell is calculated according to the following equations:

$$h_t^1 = f\left(e(w_{ij})_t\right), h_{t-1}^1 \tag{2}$$

where *f* denotes gated recurrent unit (GRU) [18], and h_{t-1} denotes the hidden output of the time step *t-1*. The GRU is calculated as follows:

$$z_t = \delta\left(U^z e(w_{ij})_t + W^z h_{t-1}^1\right) \tag{3}$$

$$r_t = \delta\left(U^r e(w_{ij})_t + W^r h_{t-1}^1\right) \tag{4}$$

$$\widetilde{h_t^1} = tanh\left(We(w_{ij}) + r_t U h_{t-1}^1\right) \tag{5}$$

$$h_t^1 = (1 - z_t) \otimes \widetilde{h_{t-1}^1} + z_t \otimes h_{t-1}^1 \tag{6}$$

here,z_t is a update gate,r_t is a reset gate,δ is a sigmoid function, *U* and *W* denote the weight, $\widetilde{h_t^1}$ is a candidate hidden layer, ⊗ represent a element-wise multiplication, h_t^1 is a final state.

3.4 Neural Pooling Layer

Neural pooling functions subsample the output of the BiLSTM layer. Max pooling function and average function are utilized to capture the highest value and average value of each dimension respectively. This technique reduces the computational complexity of upper layers. The output of this layer h^2 is defined as:

$$h^2 = \left[\begin{bmatrix}\max(h_{i1}^1)\\ \cdots \\ \max(h_{in}^1)\end{bmatrix}', \begin{bmatrix}\text{avg}(h_{i1}^1)\\ \cdots \\ \text{avg}(h_{in}^1)\end{bmatrix}, \begin{bmatrix}\text{tp}(T_{i1}^1)\\ \cdots \\ \text{tp}(T_{in}^1)\end{bmatrix}\right] \tag{7}$$

3.5 Softmax Layer

The softmax layer predicts the sentiment of the text. The output of the neural pooling layer is the input of this layer. We use the softmax function to predict the final sentiment.

$$y' = softmax(w_s h^2 + b_s) \tag{8}$$

where $y'$is a predict label, $y' \in \{+1, -1\}$,w_s and b_s denote the weight and bias respectively.

The topic information-based BiLSTM model is trained by minimizing the cross-entropy between the predicted $y'$and actual y. Given the training set ={$(s_1, y_1), (s_2, y_2), \cdots (s_n, y_n)$},where s_i represents the i^{th} sentence and y_i represents the category of the i^{th} sentence, the loss function is defined as:

$$\text{L}(\theta) = \sum_{i=1}^{k} y' \times log(y) \tag{9}$$

4. EXPERIMENTS

We conduct experiments to evaluate the performance of the of topic information-based BiLSTM model against feature engineering and neural network based methods.

4.1 Datasets

In the experiment, we have used two datasets: the sentiment in Twitter in SemEval2013 [9] and IMDB movie reviews [15]. We run 2-classes (positive vs negative) classification. For the experiment the negative and positive sentiments denote by -1 and 1 respectively. In IMDB dataset, if the review score is 4 then it denotes as negative and if the review score is 7 then it denotes as positive. The details of the datasets are described in Table 1 and 2

Table 1. SemEval2013 Twitter dataset

	Positive	Negative	Total
Train	2642	994	3636
Dev	408	219	627
Test	1570	601	2171

Table 2. IMDB movie dataset

	Positive	Negative	Total
Train	25000	25000	50000
Test	25000	25000	50000
Unlabeled data			50000

4.2 Experimental Settings

For SemEval 2013 dataset, the feature engineering based methods, namely Support Vector Machine (SVM) + ngram [3], NRC-Canada [4], and Recursive Autoencoder (RAE) [16] are used for comparison.

For IMDB dataset, the feature engineering based method, namely NBSVM [19]; neural network based method, namely LSTM, Paragraph Vector 2-layer-MLP [20] are implemented for comparison.

4.3 Model Configuration and Training

On both datasets, we tokenize each text by removing @user, URLs, and change all words to lowercase. We do not remove stop-word and non-word tokens on both datasets because certain stop-word (e.g., negative words), non-word tokens (e.g., "!"and ":-)") are indicative of sentiment. We substituted all words (e.g., 5, 10) in the text by a unique label num, and those words are occurring less than that num are not taken into consideration.

In our network, publicly available word vectors are trained from [17] and used as pre-trained word embedding. The word vectors are updated during training. The dimension of the embedding is 50. We use ReLu as the activation function. AdaGrad optimization method is used to train the network through time and learning rate of 0.01. The dropout rate is set to 0.5.

For topic modeling, we collect 28M unlabeled tweets sentiment information from Twitter API and 50000 unlabeled data from the IMDB dataset to train the topic distribution respectively. After getting the topic document, we further use a random function

uniformly sampled from range (-0.01, 0.01) to obtain a better convergence.

4.4 Metrics

Performances are evaluated using F_{score} and Accuracy. It is defined as:

• F_{score}

$$\mathrm{P} = \frac{T_P}{T_P + F_P} \tag{10}$$

$$\mathrm{R} = \frac{T_P}{T_P + T_N} \tag{11}$$

$$F_{score} = \frac{2PR}{P + R} \tag{12}$$

• Accuracy

$$\text{Accuracy} = \frac{T_P + F_N}{T_P + F_N + T_N + F_P} \tag{13}$$

where T_P, F_N, F_P, T_N have denoted as the positive in the positive class, negative in the negative class, negative in the positive class, and positive in the negative class, respectively.

4.5 Comparative Results

Table 3 and 4 show the performance of the topic information-based BiLSTM model against several methods on two datasets.

From both datasets, neural network based methods perform well than feature engineering methods. Conversely, in SemEval 2013 dataset, SVM + unigram shows 74.50%, even high-order-ngram (up to 5gram) does not improve the result (+0.47%) significantly. Because ngram features only capture the context-window size information. In IMDB dataset, Paragraoh Vector 2-layer-MLP outperform NBSVM-bi (+3.28%).

The proposed Topic Information-Based BiLSTM model shows (85.02%) better performance than the NRC (84.75%) on SemEval 2013 dataset, and presents (95%) on IMDB dataset which is better than Paragraoh Vector 2-layer-MLP. BiLSTM model captures long sequence information. Besides, the proposed model is capable of addressing the problem by optimizing the classification result.

The experimental results demonstrate that the proposed topic information-based BiLSTM network is effective for sentiment classifications.

Table 3. Results on SemEval 2013 dataset

Model	Fscore
Recursive Autoencoder	75.42
SVM+unigram	74.50
SVM+5-gram	74.97
NRC-Canda (Top System in SemEval2013)	84.73
Topic Information-Based BiLSTM	**85.02**

Table 4. Results on IMDB dataset

Model	Accuracy
LSTM	89.1
NBSVM-bi	91.22
Paragraoh Vector 2-layer-MLP	94.5
Topic Information-Based BiLSTM	**95**

5. CONCLUSION

This study represents topic information-based BiLSTM model for sentiment classification. The traditional feature engineering and single word representation methods are not sufficient for sentiment analysis. The proposed topic Information-based BiLSTM network learns topic information automatically without using external resources or hand-crafted feature. Experimental results demonstrated that the proposed method outperforms conventional neural network and feature engineering studies.

ACKNOWLEDGMENTS

This research work was supported by Guangdong Foundation Grant No.2015A030310364 and No. 2015A080804019; National Natural Science Foundation of China Grant No.61702306.

6. REFERENCES

[1] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends R in Information Retrieval*, 2(1–2):1–135, 2008

[2] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[3] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

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

[5] Sepp Hochreiter and Ju ̈rgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

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

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

[8] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[9] Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 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*

Workshop on Semantic Evaluation (SemEval 2013), volume 2, pages 312–320, 2013.

[10] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[11] Bing Xiang and Liang Zhou. Improving twitter sentiment analysis with topic-based mixture modeling and semi-supervised training. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 434–439, 2014.

[12] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[13] Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pages 231–240. ACM, 2008.

[14] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[15] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[16] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning.Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing*, pages 151–161. Association for Computational Linguistics, 2011.

[17] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565, 2014.

[18] Kyunghyun Cho, Bart Van Merri ënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[19] S. I. Wang and C. Manning. Baselines and bigrams: Simple, good sentiment and text classification. In *Association for Computational Linguistics (ACL)*, 2012.

[20] James Hong and Michael Fang. Sentiment analysis with deeply learned distributed representations of variable length texts. Technical report, Technical report, Stanford University, 2015.