The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Aspect-Based Sentiment Analysis Using Bitmask Bidirectional Long Short Term Memory Networks

Binh Thanh Do
School of Information and Communication Technology
Hanoi University of Science and Technology, Hanoi, Vietnam

Abstract

This paper introduces a new method to classify sentiment polarity for aspects in product reviews. We call it bitmask bidirectional long short term memory networks. It is based on long short term memory (LSTM) networks, which is a frequently mentioned model in natural language processing. Our proposed method uses a bitmask layer to keep attention on aspects. We evaluate it on reviews of restaurant and laptop domains from three popular contests: SemEval-2014 task 4, SemEval-2015 task 12, and SemEval-2016 task 5. It obtains competitive results with state-of-the-art methods based on LSTM networks. Furthermore, we demonstrate the benefit of using sentiment lexicons and word embeddings of a particular domain in aspect-based sentiment analysis.

Introduction

Sentiment analysis is an important topic in natural language processing. Sentiment can be classified by three following polarities: positive, negative and neutral. Aspect-based sentiment analysis (ABSA) is a fundamental task in sentiment analysis. We have to find out the sentiment polarity of some specific aspects expressed in a comment or review. For example, in the review "The price is reasonable although the service is poor.", the expected sentiment polarity of "price" is positive, while the expected sentiment polarity of "service" is negative. ABSA can help consumers decide what to purchase because many consumers use the website to share their experiences about products, services, or travel destinations. Many companies are interested in this issue because they not only want to know what people ask about their reputation and products but also understand the needs of the market. Data is available in the newspaper, social networks. However, extracting the opinions from these is really a challenge. Therefore, many contests are held in recent years, such as task 4 in SemEval-2014 (SE-ABSA14) (Pontiki et al. 2014), task 12 in SemEval-2015 (SE-ABSA15) (Pontiki et al. 2015), and task 5 in SemEval-2016 (SE-ABSA16) (Pontiki et al. 2016). They attract interest of many research teams. Some research teams use traditional approaches which are sentiment lexicon, logistic regression, topic models (Lu et al. 2011) or Support Vector Machine (SVM) (Varghese and Jayasree 2013). Recently, the usage of LSTM (Hochreiter and Schmidhuber 1997) networks has been preferred (e.g., (Tang et al. 2015), (Ruder, Ghaffari, and Breslin 2016)) because of its effects in many natural language processing tasks. However, they face many challenges. The primary challenge is an extension of training data for deep neural networks. The second challenge is keeping the attention to aspects because of the complicated context of reviews.

In this paper, we introduce a bidirectional LSTM network using a bitmask layer to keep attention on the position of aspects. After the bitmask layer, sentiment polarity of all works in the review will be predicted through a softmax layer. In this way, we not only keep our model pays attention to aspects but also increase the number of training samples. In addition, we experiment with some word embedding approaches (e.g., word2vec (Mikolov et al. 2013), GloVe (Pennington, Socher, and Manning 2014)) of particular corpora and sentiment lexicon to improve classification accuracy.

We evaluate our approach on SE-ABSA14, SE-ABSA15, and SE-ABSA16 review datasets for restaurant and laptop domains. We compare our results with various baselines and state-of-the-art methods, which are mostly based on LSTM networks. The contributions of this paper are:

- Propose a neural network model to solve ABSA task and achieve some state-of-the-art results.
- Analyze the use of word embeddings trained by a particular corpora.
- Use sentiment lexicon to improve the classification accuracy.

Related work

Sentiment analysis with deep neural networks

In this section, we revise some approaches using deep neural networks to solve ABSA task.

Tang et al. (2015) develop a target-dependent long short term memory (**TD-LSTM**) to keep attention on the aspect phrases. Each aspect phrase is represented by means of LSTM hidden vector of each word in it. They upgrade TD-LSTM to a target-connection long short term memory (**TC-LSTM**) because they think TD-LSTM does not capture the connection between aspect and each context word when building the representation of a sentence. The difference is that in TC-LSTM the input at each position is the concatenation of word embedding and target vector, whereas in TD-LSTM the input only consists of the embedding vector of the current word. The target vector is the average of the vectors of words that aspect phrase contains. Both TD-LSTM and TC-LSTM get better results than using traditional LSTM.

Wang et al. (2016) propose another attention approach (**ATAE-LSTM**) for aspects in a review. They learn an embedding vector for each aspect, then concatenate it to the corresponding LSTM hidden vector of this aspect. After that, they combine this aspect attention vector with the last hidden vector of LSTM to predict the sentiment polarity. The downside of this method is that there are a lot of aspects. Therefore, many aspect embeddings are needed if we want to apply the model to reality life. Their model obtains comparable performances with TD-LSTM and TC-LSTM.

Ruder, Ghaffari, and Breslin (2016) show a hierarchical model of reviews (**HP-LSTM**) for ABSA. Firstly, they randomly initialize a representation vector for each aspect category. Then they extract features by a hierarchical bidirectional which are sentence-level LSTM and review-level LSTM. At the last layer, they concatenate the review-level embedding, which is extracted by LSTM, with the aspect category embedding vector. This model obtains some state-of-the-art results in the SE-ABSA16 datasets.

Tang, Qin, and Liu (2016) build a deep memory network (**MemNet**) consisting of multiple computational layers (hops), each of which contains an attention layer and a linear layer. Each word embedding combines with a location attention weight to form a vector, which is called an external memory. Then all context is computed as a weighted sum of each piece of external memory. Then the authors feed it to a softmax layer for aspect level sentiment classification. This model brings significant improvements in the SE-ABSA14 benchmark datasets. It seems the more memory layers, the higher accuracy is achieved.

Our work is in line with these methods, using word embeddings and a deep neural network to exploit the syntactic and semantic structures of reviews automatically. However, when they specialized in aspects embeddings and attention techniques, we use LSTM networks to summarize and capture context. The first difference is that our proposed method keeps attention on the last word of aspects to predict the sentiment polarity of it. For example, if the aspect is “battery life”, we focus on the information that was embedded with word “life”. For future convenience, we consider an aspect as a single word. The second difference of our method with other methods is that we predict the sentiment polarity of all words in the review, not just for aspect words. The non-aspect words have a “neutral” label. This method indirectly increases the number of training samples.

Sentiment embeddings

The context-based embedding learning algorithms (e.g., word2vec, GloVe) ignore the sentiment information of words. For example, vector representation of “good” and “bad” are close in the embedding space. Tang et al. (2014) proposed a method, sentiment-specific word embeddings (SSWE), which encodes the sentiment information into the vector representation of words so that it can separate “good” and “bad” in the embedding space. It is called sentiment-specific word embeddings (SSWE) or sentiment embeddings. Using SSWE feature performed competitive results in context-based sentiment analysis (Tang et al. 2016), but it gave a poor performance in ABSA (Tang et al. 2015).

Our approach

Word embeddings

One of the purposes of this paper is to use specific domain corpus to learn the word embeddings. For example, when we work on restaurants, we find some additional data about restaurant reviews and use it to learn the word embeddings of this domain. We expect the word embeddings trained on domain-specific corpora to capture semantic similarities between the words. Also, we compare particular domain embeddings with the standard pre-trained word embeddings, i.e., glove.42B.300d [1] is trained by GloVe, embed_tweets_en_200M_200D [2] (Deriu et al. 2017) is trained by word2vec, to evaluate which corpus and word embedding model are good for aspect-based sentiment analysis. Let us define some notations: $\mathbb{L} \in \mathbb{R}^{d\times|V|}$ is an embedding lookup table where d is the dimension of a word vector and $|V|$ is the vocabulary size. Each word is represented by a d-dimensional vector $[x_1\ x_2\ \ldots\ x_d]$ where $x_i \in \mathbb{R}$. Each review is represented by a set of d-dimensional vectors $\{w_1, w_2, ..., w_n\}$ where $w_i \in \mathbb{R}^d$ and n is the length of the review.

Sentiment lexicons

Inspired by the sentiment embeddings (Tang et al. 2014), we use sentiment lexicon as an added feature for a word representation. The sentiment lexicon contains two main types: positive words (e.g., “great”, “awesome”, “good”), negative words (e.g., “poor”, “afraid”, “bad”). We create a one-hot vector to represent the sentiment polarity of a word. The first position of this vector expresses the positive words, the second position expresses the negative words, and the third position expresses the others. For example, the corresponding one-hot vectors of “good”, “bad” and “price” are [1 0 0], [0 1 0] and [0 0 1]. Then, we concatenate the one-hot vector into the word embedding. Therefore, a word is represented by a vector $[x_1\ x_2\ \ldots\ x_d\ s_1\ s_2\ s_3]$ where $x_i, s_i \in \mathbb{R}$ and $[s_1\ s_2\ s_3]$ is a one-hot vector. We have a new embedding lookup table $\mathbb{L}' \in \mathbb{R}^{(d+3)\times|V|}$. When concatenating the one-hot vector into the word embedding, we make the words, whose sentiment information are the same, closer to each other in the embedding space. In our experiment, we use an open source sentiment lexicon (around 6800 words), which has been collected by the authors (Hu and Liu 2004).

Long short term memory

Recurrent neural networks (RNNs) (Jain and Medsker 1999) are traditional models that have shown prospective promise in many NLP tasks. The output depends on the previous

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

[2] https://spinningbytes.com/resources/embeddings/

computations because RNNs have a "memory", which captures information on what has been calculated so far. But the regular RNNs cannot handle long-term dependencies because they consist of many biases and some useless past information when the sequence length of the input is large. Also, it is a cause of the vanishing/exploding gradient problem (Bengio, Simard, and Frasconi 1994). LSTMs (Hochreiter and Schmidhuber 1997) are designed to avoid useless information in the past. An LSTM cell at time t contains a forget gate f_t to forget history, an input gate i_t to add current information, and an output gate o_t to generate the output. A cell state c_t decides what information should be kept or erased from the context. The LSTM cell at time t is calculated as follows.

$$f_t = \sigma(W_f.[h_{t-1}; w_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i.[h_{t-1}; w_t] + b_i) \tag{2}$$

$$o_t = \sigma(W_o.[h_{t-1}; w_t] + b_o) \tag{3}$$

$$\tilde{c}_t = tanh(W_c.[h_{t-1}; w_t] + b_c) \tag{4}$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \tag{5}$$

$$h_t = o_t \odot tanh(c_t) \tag{6}$$

where w_t, h_t are the input word vector and hidden vector at time t; $W_f, W_i, W_o, W_c \in \mathbb{R}^{d' \times 2d}$ are weight matrices; $b_f, b_i, b_o, b_c \in \mathbb{R}^{d'}$ are bias vectors; d is dimension of word vector; d' is dimension of hidden vector; σ is sigmoid function; $\odot$ stands for element-wise multiplication.

Bitmask bidirectional long short term memory

A bidirectional long short term memory includes two LSTM layers: forward and backward LSTM. Forward LSTM processes the sequence in chronological order while backward LSTM processes the sequence in reverse order. In our experience, using either element-wise sum or concatenation to combine the forward with backward LSTM outputs gives the same result. Therefore, we choose element-wise sum to reduce the number of parameters at the next layer.

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{7}$$

As being shown in Fig. 1, our proposed model predicts the sentiment polarity for all words in the input review. Our key point is the bitmask layer m which is an n-dimensional vector of 0 and 1, where n is the length of the review. It has a value of 1 at positions of aspects and a value of 0 at positions of others. We keep the model look up the tth word input by multiplying the hidden vector h_t by the bit m_t.

$$h'_t = m_t.h_t \tag{8}$$

We use a softmax layer after the last predicted linear layer to normalize the output probability. The ith output of the softmax layer at time t is calculated as followings.

$$\hat{y}_{t,i} = \frac{exp(a_{t,i})}{\sum_{k=1}^{C} exp(a_{t,k})} \tag{9}$$

where C is the number of sentiment categories and $a_t = W_s.h'_t + b_s$ where $W_s \in \mathbb{R}^{C \times d'}, b_s \in \mathbb{R}^{C}$ are weight matrix and bias vector of softmax layer. In our experiment, the sentiment categories include positive, negative and neutral polarities. Therefore, C equals 3.

The equation (8) shows that h'_t is a zero vector when m_t equals 0 and h'_t has the same dimension as h_t. Therefore, by using a bitmask layer after the LSTM layer, we concentrate more on aspect words than non-aspect words. Also, we increase the number of training samples, especially for "neutral" samples because we predict the sentiment of all words in the context and the label of non-aspect words is "neutral".

Model training

The model is trained by minimizing the cross entropy plus an L_2 regularization term.

$$J(\theta) = -\frac{1}{|D|} \sum_{y_d \in D} \sum_{i=1}^{C} y_{d,i} log \hat{y}_{d,i} + \lambda ||\theta|| \tag{10}$$

where C is the number of sentiment categories, D is the collection of training labels, y_d is a one-hot vector where the element for the true sentiment is 1, $\hat{y}_d$ is the output of the last softmax layer, λ is the weight of L_2 regularization term, θ is the parameter set $\{W_f, W_i, W_o, W_c, W_s, b_f, b_i, b_o, b_c, b_s\}$. We use stochastic gradient descent algorithm to train and update parameters.

As far as we know, gradient descent algorithm is $\theta^{new} = \theta^{old} - \alpha \nabla_\theta J(\theta)$. At the tth word of the review, if m_t equals 0, h'_t is a zero vector. Hence, $a_t = W_s.h'_t + b_s = b_s$. Therefore, $\frac{\partial}{\partial W_s}(\sum_{i=1}^{C} y_{t,i} log \hat{y}_{t,i}) = 0$ and similar for $W_f, W_i, W_o, W_c, b_f, b_i, b_o, b_c$. So, the bit 0 in the bitmask only affects the bias vector b_s when using gradient descent algorithm. It means the bitmask can help increase the number of training samples without breaking the structure of LSTMs. Also, the variance of b_s makes our model avoid over-fitting.

Experiment

Datasets

We conduct experiments on review datasets of restaurant and laptop domains from SE-ABSA14, restaurant domain from SE-ABSA15 and SE-ABSA16. We removed a few examples having the "conflict label" in the SE-ABSA14 datasets. The number of each label in training and test set is shown in Table 1. In SE-ABSA14, we are given a set of aspects within a review, and each aspect has its own sentiment polarity. But in SE-ABSA15 and SE-ABSA16, we are given a set of aspects and their categories within a review, each category has its own sentiment polarity. An aspect category is defined as a combination of an entity type E and an attribute type A. For example, in the review "The wine list is interesting and has many good values", we are given two categories of the aspect "wine list", which are DRINKS#STYLE_OPTIONS and DRINKS#PRICES. This definition of aspect category makes the difference between entities and the particular facets that are being evaluated more explicitly. We decided to replace the aspect by its categories. For example, before: "I liked the **atmosphere** very much but the **food** was not worth the price", after: "I liked the **AMBIENCE#GENERAL** very much but the **FOOD#QUALITY FOOD#PRICES** was not worth the

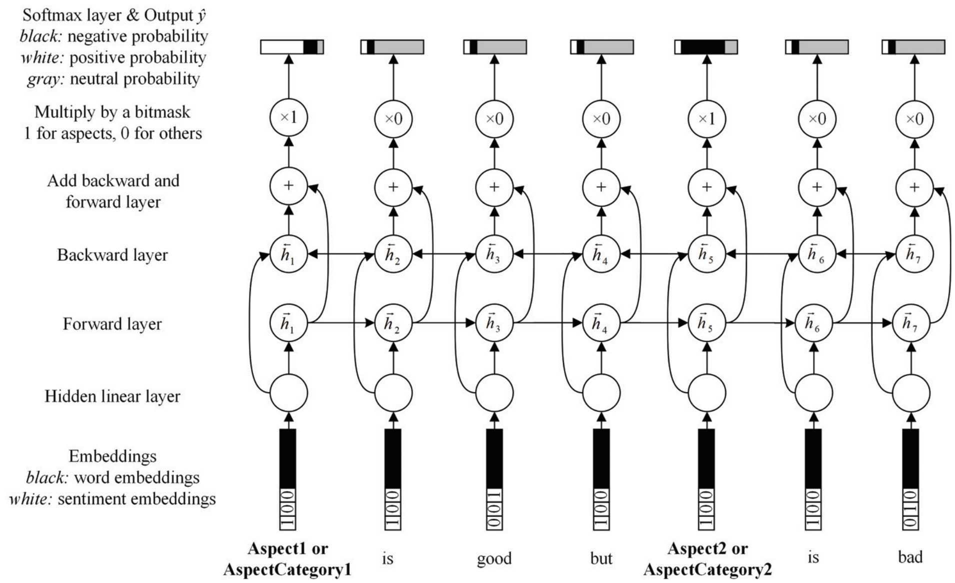

Figure 1: Bitmask bidirectional LSTM networks

SE dataset	Train			Test		
	Pos	Neg	Neu	Pos	Neg	Neu
SE14-Res	2159	800	632	730	195	196
SE14-Lap	980	858	454	340	128	171
SE15-Res	1198	404	53	457	347	45
SE16-Res	1660	751	101	613	206	44

Table 1: Statistics of the datasets

price". Sometimes, a review has no aspects, but it still has some hidden categories. We put these hidden categories into the end of the review then predict the opinion of it. For example, before: "After all that, they complained to me about the small tip", after: "After all that, they complained to me about the small tip SERVICE#GENERAL". When training the word embeddings, we consider an aspect category E#A as a word.

Data preprocessing and experimental setting

In the recurrent neural networks, if there are more sequence units, the more biases are added to its hidden vectors because $h_t = f_W(x_t, h_{t-1})$. Besides that, we will face the vanishing/exploding gradient problem when the length of the input is large. Therefore, we build a set of stop words for sentiment analysis then we remove them from the review. A stop word for ABSA task is a word that does not contribute the sentiment of the review such as *"a", "an", "the", "I", "I've", "me", "my", "myself"*. Furthermore, we remove some non alphabet characters, such as *. , : ; / ˜ - ? ! " ( )*.

Our proposed model contains an embedding layer, a hidden linear layer, a forward LSTM layer, a backward LSTM layer, a bitmask layer, and a softmax layer at last. We experiment with various word embedding approaches: word2vec on a specific domain corpus (100 dimensions, W.S), GloVe on a specific domain corpus (100 dimensions, G.S), GloVe on large common crawler (300 dimensions, G.L), and word2vec on 200 million English Tweets (200 dimensions, (W.L). We use laptop data of SE-ABSA and six categories of Amazon product reviews [3] to train word embedding for laptop domain. For the restaurant domain, we use data of SE-ABSA contest and Amazon fine food reviews.[4]. We use Xavier initialization (Glorot and Bengio 2010) for initial parameters, 0.5 for dropout probability, 0.0005 for weight decay, and set learning rate as 0.1. These parameters are chosen based on the size of our deep neural network. Also, we use dropout technique and early stopping to avoid over-fitting.

We compared our results with some baselines, recently state-of-the-art methods, and the top winners at SE-ABSA contests such as feature-based SVM (Kiritchenko et al. 2014), TDLSTM, ATAE-LSTM, HP-LSTM, MemNet. These methods are described in the related work section. We take their results from their paper. For future convenience, we use **BBLSTM** notation for our proposed model bitmask bidirectional long short term memory and **BBLSTM-SL** for BBLSTM using sentiment lexicon as a feature for words.

Experimental results

We apply our method to aspect-based sentiment classification to evaluate its effectiveness. In the Fig. 2, when the training accuracy score tends to increase to 100%, whereas the training loss tends to decrease to 0.0 after 500 epochs. It means that our model can fit the training data. The results of accuracy in Tables 2-5 are based on the observations on SE-ABSA14, SE-ABSA115 and SE-ABSA16 test set.

[3] http://times.cs.uiuc.edu/ wang296/Data/

[4] https://www.kaggle.com/snap/amazon-fine-food-reviews

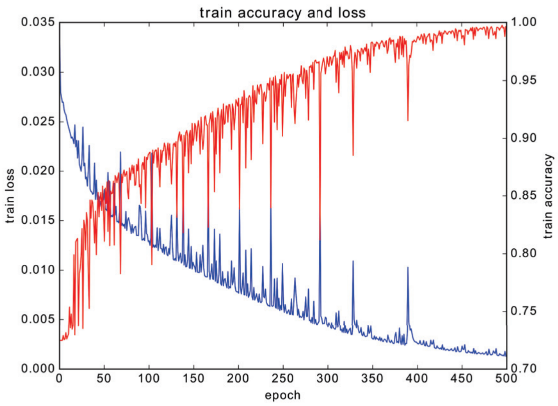

Figure 2: *Red:* training accuracy. *Blue:* training loss

Method	Restaurant	Laptop
Feature+SVM	80.9	72.1
LSTM	74.3	66.5
TDLSTM	75.6	68.1
ATAE-LSTM	77.2	68.7
MemNet	81.0	72.4
BBLSTM	79.6	73.0
BBLSTM-SL	**81.3**	**74.9**

Table 2: Comparison of different methods on SE-ABSA14

Most research teams use SE-ABSA14 as the benchmark dataset for ABSA because SE-ABSA15 and SE-ABSA16 datasets have conflict sentiment for each aspect based on its categories. Our model brings significant improvements in the SE-ABSA14 dataset, outperforms TDLSTM, ATAE-LSTM in laptop domain. Experimental results of baseline models and our model are given in Table 2.

We achieved the highest accuracy score in the SE-ABSA15, 81.2% as being shown in Table 3. The best accuracy score in this contest, 78.69%, was achieved by Sentiue (Saias 2015) with a MaxEnt classifier along with features based on n-grams, POS tagging, lemmatization, negation words and publicly available sentiment lexicon (MPQA, Bing Liu's lexicon, AFINN). The system of ECNU (Zhang and Lan 2015) (78.10%) used features based on n-grams, PMI scores, POS tags, parse trees, negation words and scores based on seven sentiment lexicon. The lsislif team (Hamdan, Bellot, and Bechet 2015) (75.50%) relied on a logistic regression model (Liblinear) with various features: syntactic (e.g., unigrams, negation), semantic (Brown dictionary), sentiment (e.g., MPQA, SentiWordnet).

In the SE-ABSA16 contest, our model achieved the best result among all models based on LSTM as being shown in Table 4, better 0.5% than HP-LSTM (Ruder, Ghaffari, and Breslin 2016), which is the state-of-the-art method using LSTM in SE-ABSA16. Once again, the result of BBLSTM-SL shows that sentiment terms are useful features for words.

Each different word embeddings of a particular corpus obtain competitive results as given in Table 5. But using word embeddings, which is trained on a specific domain corpus by word2vec, gives a slight improvement than others. We do not use G.L and W.L pre-trained word embeddings in SE-ABSA15 and SE-ABSA16 because they do not contain embedding vectors for aspect categories such as FOOD#QUALITY, SERVICE#QUALITY, DRINK#OPTIONS.

Method	Accuracy
Sentiue	78.7
ECNU	78.1
Lsislif	75.5
SVM + BOW	63.6
BBLSTM	79.8
BBLSTM-SL	**81.2**

Table 3: Comparison of different methods on SE-ABSA15

Method	Accuracy
SVM	76.5
LSTM	81.4
HP-LSTM	85.3
BBLSTM	84.4
BBLSTM-SL	**85.8**

Table 4: Comparison of different methods on SE-ABSA16

SE dataset	W.S	W.L	G.S	G.L
ABSA14-Lap	**74.9**	**74.9**	74.6	**74.9**
ABSA14-Res	**81.3**	80.0	80.3	80.7
ABSA15-Res	81.2	-	**81.3**	-
ABSA16-Res	**85.8**	-	84.1	-

Table 5: Classification accuracy of BBLSTM-SL with different word embeddings

sure coming back **restaurant#general**(label 1,predict 1)
waiter(label 2,predict 2) horrible rude disinterested
isn't cheapest **food#prices**(label 0,predict 1) **food#quality**(label 1,predict 1) but worth every time
a great **assortment wraps**(label 1,predict 1) if not mood for **traditional mediterranean fare**(label 0,predict 1)
good go for **drinks**(label 0,predict 0) if want get drunk because lucky if can get one **drink**(label 0,predict 0) hour **service**(label 2,predict 2) bad

Table 6: Case study. {0: neutral, 1: positive, 2: negative}

Case study

Table 6 are some examples showing the upsides and downsides of our model. Although our model does not handle negation terms (e.g., isn't, don't, but) very well, it performs well in the review which contains sentiment words. We tried to use negation lexicon and embedded it to the word embeddings like the way we have done with sentiment lexicon, but that does not work. On the other hand, it is hard for recurrent neural networks to handle the information of a long and complicated review which contains more aspects.

Conclusion

In this paper, we show two main things. Firstly, we build a bitmask bidirectional LSTM network for solving aspect-based sentiment analysis. It brings significant improvements in SE-ABSA14, SE-ABSA15, and SE-ABSA16 datasets. Secondly, using word embedding of a particular domain and sentiment lexicon as a word feature increases the classification accuracy up to 1.9%. Our BBLSTM-SL model achieves the state-of-the-art results. In the future, we would like to improve our model by using more features, combining with other techniques, and handling the negation terms.

References

Bengio, Y.; Simard, P.; and Frasconi, P. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* 5(2):157–166.

Deriu, J.; Lucchi, A.; De Luca, V.; Severyn, A.; Müller, S.; Cieliebak, M.; Hofmann, T.; and Jaggi, M. 2017. Leveraging large amounts of weakly supervised data for multi-language sentiment classification. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, 1045–1052. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee.

Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W., and Titterington, M., eds., *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, 249–256. Chia Laguna Resort, Sardinia, Italy: PMLR.

Hamdan, H.; Bellot, P.; and Bechet, F. 2015. Lsislif: Crf and logistic regression for opinion target extraction and sentiment polarity analysis. In *SemEval@ NAACL-HLT*, 753–758.

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

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168–177. ACM.

Jain, L. C., and Medsker, L. R. 1999. *Recurrent Neural Networks: Design and Applications*. Boca Raton, FL, USA: CRC Press, Inc., 1st edition.

Kiritchenko, S.; Zhu, X.; Cherry, C.; and Mohammad, S. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *SemEval@ COLING*, 437–442.

Lu, B.; Ott, M.; Cardie, C.; and Tsou, B. K. 2011. Multi-aspect sentiment analysis with topic models. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, 81–88. IEEE.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; and Manandhar, S. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 27–35. Dublin, Ireland: Association for Computational Linguistics and Dublin City University.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Manandhar, S.; and Androutsopoulos, I. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 486–495. Denver, Colorado: Association for Computational Linguistics.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; AL-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *ProWorkshop on Semantic Evaluation (SemEval-2016)*, 19–30. Association for Computational Linguistics.

Ruder, S.; Ghaffari, P.; and Breslin, J. G. 2016. A hierarchical model of reviews for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 999–1005.

Saias, J. 2015. Sentiue: Target and aspect based sentiment analysis in semeval-2015 task 12. Association for Computational Linguistics.

Tang, D.; Wei, F.; Yang, N.; Zhou, M.; Liu, T.; and Qin, B. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1555–1565. Baltimore, Maryland: Association for Computational Linguistics.

Tang, D.; Qin, B.; Feng, X.; and Liu, T. 2015. Target-dependent sentiment classification with long short term memory. *CoRR, abs/1512.01100*.

Tang, D.; Wei, F.; Qin, B.; Yang, N.; Liu, T.; and Zhou, M. 2016. Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering* 28(2):496–509.

Tang, D.; Qin, B.; and Liu, T. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 214–224.

Varghese, R., and Jayasree, M. 2013. Aspect based sentiment analysis using support vector machine classifier. In *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, 1581–1586. IEEE.

Wang, Y.; Huang, M.; Zhu, X.; and Zhao, L. 2016. Attention-based lstm for aspect-level sentiment classification. In *EMNLP*, 606–615.

Zhang, Z., and Lan, M. 2015. Ecnu: Extracting effective features from multiple sequential sentences for target-dependent sentiment analysis in reviews. In *SemEval@ NAACL-HLT*, 736–741.