Learning Word Sentiment with Neural Bag-Of-Words Model Combined with Ngram

Chunzhen Jing, Jian Li, Xiuyu Duan

Learning Word Sentiment with Neural Bag-Of-Words Model Combined with Ngram

Chunzhen Jing, Jian Li, and Xiuyu Duan

Beijing University of Posts and Telecommunications
No.10 Xitucheng Road, Haidian District Beijing, China
{jcz,lijian,duanxy}@bupt.edu.cn

Abstract. To better analyze the sentiment, attitude, emotions of users from written language, it is necessary to identify the sentiment polarity of each word not only the overall sentiment (positive/neutral/negative) of a given text. In this paper we propose a novel approach by using a method based on Neural Bag-Of-Words (NBOW) model combined with Ngram, aiming at achieving a good classification score on short text which contain less than 200 words along with sentiment polarity of each word. In order to verify the proposed methodology, we evaluated the classification accuracy and visualize the sentiment polarity of each word extracted from the model, the data set of our experiment only have the sentiment label for each sentence, and there is no information about the sentiment of each word. Experimental result shows that the proposed model can not only correctly classify the sentence polarity but also the sentiment of each word can be successfully captured.

Keywords: word sentiment, short text, neural bag-of-words, Ngram

1 Introduction

Automatic sentiment analysis is a fundamental problem and one of the most active research areas in natural language processing (NLP) which has been widely used in data mining and text mining [18, 20]. Detecting sentiment on short text such as reviews on certain product or exchanging information and opinions via short 200 words messages is becoming ubiquitous. There has been a large amount of research in this area of sentiment classification. Sentiment classification mainly focus on categorizing these texts in either two (binary sentiment analysis) or three (ternary sentiment analysis) categories, and this is an explicitly unordinal classification problem.

Neural network and deep learning have shown great promise in natural language processing (NLP) over the past few years. Examples are in semantic analysis [9], machine translation [1,4]. However many techniques of deep learning in sentiment classification suffer from over-abstraction problem [19], traditionally most of it has focused on classifying the text into several different categories, the only information obtained from these techniques is the polarity of the texts, and it's difficult to extract the sentiment knowledge more in depth, such as the sentiment of each word, i.e., positive intensity and negative intensity of a certain word.

In this paper, we propose a sentiment classification model based on Neural Bag-Of-Words (NBOW) [8] combined with Ngram, named NBOWN. The main advantage of the proposed model is its ability to extract the sentiment of each word in a text without explicit word-level polarity information. It identifies the words only by sentence-level polarity that is more abstracted but easier to availability.

In our model, each word is represented as a continuous-valued vector [3] and each sentence is represented as a matrix whose rows correspond to the word vector used in the sentence. Then, the model is trained using these sentence matrices as inputs and the sentiment labels as the output. Both the sentence-level polarity and words-level polarity for all words in the text can be extracted while the training, which helps us better understand the result of sentence-level sentiment classification.

The rest of the paper is organized as follows. First in Section 2 we discuss about the related works.In section 3 we briefly introduce the NBOW model and present our proposed model, named as Neural Bag-Of-Words-Ngram (NBOWN) model, in Section 4. In section 5, we give details about the data and the experiment setup. Section 6 gives experiment results and visualization of word sentiment performed by our model. Finally, we give our conclusions in Section 7.

2 Related Work

A variety of neural network architectures have been proposed for different language processing tasks. In sentiment classification, fully-connected feed forward neural networks [6], convolutional neural networks (CNN) [10, 26] and also recurrent/recursive neural networks (RNN) [7] have been used. The CNN models are characterized by a set of convolution filters acting as a sliding window over the input sequence, which act as powerful n-gram feature extractors, typically followed by a pooling operation (such as max-pooling) [29] to generate a fixed-vector representation of the input sentence.

Recently, recurrent neural network architectures (RNNs) [17], such as long short term memory networks (LSTMs) [16] and Gated Recurrent Unit (GRU) [5], have received significant attention for various NLP tasks. However, the long term relationships captured well by LSTMs/GRU are of minor importance to the sentiment analysis of short texts. Even though the attention mechanism based on recurrent neural networks [27] can learn the task specific word importance, it doesn't explicitly model the sentiment polarity of each word in the text. Additionally, RNNs are much more computationally expensive, and both CNNs and RNNs require careful hyper-parameter selections and regularizations [28].

A Bag-Of-Words BOW represents text as a vector of word features such as word occurrence frequency and variants of term frequent-inverse document frequency known as tf-idf. BOW methods can be also applied in many areas [2, 24]. With the development of neural network and deep learning based language processing, the syntactic and semantic characteristics of words and their surrounding context can be captured by using a more powerful continuous vector representation of words [3, 12], such as word2vec [15], GloVe [21] and they outperform the count based word representation. The Neural Bag-Of-Words (NBOW) [8] model performs classification with an average

of the input word vectors and achieves an impressive performance. We focus our model based on Neural Bag-Of-Words (NBOW) model.

3 Neural Bag-Of-Words (NBOW) model

The NBOW model is a fully connected network, the input is an average of the d dimensional word vectors, for the words w in text X, corresponding vector v_w is looked up, and a hidden vector representation is obtained as follows:

$$s = \frac{1}{|X|} \sum_{w \in X} v_w \tag{1}$$

The average vector s is fed to a fully connected layer to estimate the probabilities for the output label as:

$$\hat{y} = softmax(Ws + b) \tag{2}$$

where $W \in R^{d \times K}$, K is class number, b is a bias vector and softmax is like follows:

$$softmax(q) = \frac{exp(q)}{\sum_{j=1}^{K} exp(q_j)} \tag{3}$$

For sentiment classification tasks the NBOW is trained to minimise the cross entropy loss using a gradient descent algorithm.

4 Proposed model: Neural Bag-Of-Words-Ngram (NBOWN)

While the NBOW learns word vectors specialised for the sentiment classification task, and the overall sentiment of the sentence can be captured, it lacks to identify high-contributing words to classification results, and it cannot tell the sentiment of a certain word. This paper presents a novel approach for sentiment classification on short text. Both the importance and the contribution to each sentiment polarity of each word can be captured.

It is easy to realize that the NBOW model is essentially a fully connected feed forward network with a BOW input vector, and it is a unigram model which only the unigram pattern of the text is considered. Inspired by the powerful n-gram extractors in CNNs, We thus propose the Neural Bag-Of-Words-Ngram (NBOWN) model, with the motivation to enable the NBOW model to combine with the unigram, bigram and trigram knowledge of the text.

To get the impact of each words in a text on each sentiment polarity, we first map each word vector to a 3-dimensional vector, each dimension shows the sentiment of this word, which can be positive, neutral and negative. The method proposed in NBOW2 [23] model was used to let the model learn the word importance weights which are task specific, as [23] shows the word weights learned by the model achieve accuracy closer to tf-idf variants.

The unigram pattern score is a weighted average of the 3-dimensional vectors mapped from the word vectors.

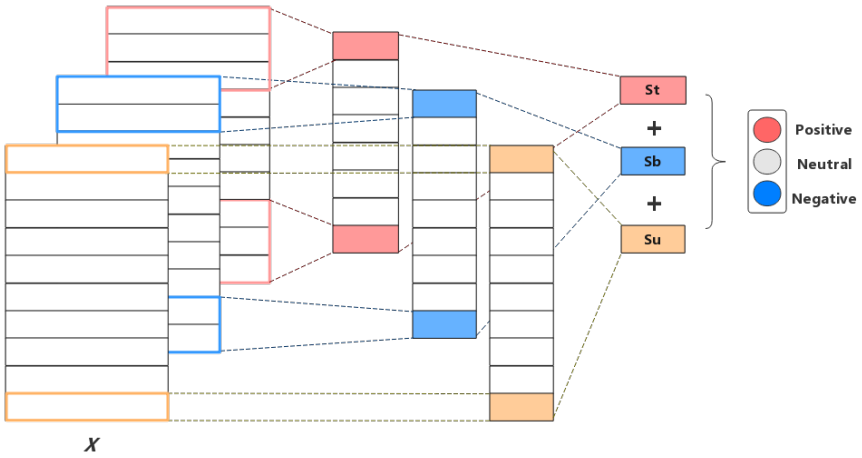

Fig. 1. Framework of proposed method

$$s_u = \frac{1}{|X|} \sum_{u \in X} W_u v_u \alpha_u \tag{4}$$

where v_u is a unigram pattern of word vector v_w, and in unigam pattern they are equal, $W_u \in R^{d \times K}$ maps the d-dimensional vector v_u to a K dimensional vector, K is number of class, in our cases $K = 3$, the α_u are the scalar word importance weights for unigram pattern $v_u \in X$, α_u are obtained by introducing a vector a_u in the model, and are calculated as follows:

$$\alpha_u = f(v_u \cdot a_u) \tag{5}$$

where $v_u \cdot a_u$ represents a dot product between input vector v_u and vector a_u, and f scales the importance weights to range $[0, 1]$. In our model, the sigmoid fuction $f(t) = (1 + e^{-t})^{-1}$ is used.

To apply the ngram pattern to NBOW model, the bigram and trigram pattern of the text is used, like follows,

$$v_b = mean(v_{wi}, v_{wi+1}) \tag{6}$$

$$v_t = mean(v_{wi}, v_{wi+1}, v_{wi+2}) \tag{7}$$

$$mean(v_{wi}, v_{wi+1}, ...v_{wn}) = \frac{1}{n+1}(v_{wi} + v_{wi+1} + .. + v_{wn}) \tag{8}$$

the v_b is an mean value of the word vectors v_{wi} and v_{wi+1}, v_{wi} represents ith word vector in text X. The bigram pattern is an average of v_{wi} and its adjacent vector v_{wi+1}, the trigram pattern follows the same way. To address the sparse problem when introducing the ngram to the NBOW model, we use the same method of Equation 5. The bigram/trigram pattern score is a weighted average of K dimensional vectors mapped from bigram/trigram patterns.

$$s_b = \frac{1}{|X| - 1} \sum_{b \in X} W_b v_b \alpha_b \tag{9}$$

$$s_t = \frac{1}{|X| - 2} \sum_{t \in X} W_t v_t \alpha_t \tag{10}$$

The α_b and α_t is a scalar scales the importance weights of a ngram pattern. And is calculated as follows:

$$\alpha_b = f(v_b \cdot a_b) \tag{11}$$

$$\alpha_t = f(v_t \cdot a_t) \tag{12}$$

For final result, like the Figure 1 shows, the softmax function will get the probablity estimates of scores get in ngram model.

$$\hat{y} = softmax(s_u + s_b + s_t) \tag{13}$$

And the sentiment distribution of ngram pattern in text X can be calculated as:

$$d = softmax(W_{ngram} v_{ngram}) \tag{14}$$

d is the sentiment distribution of certain ngram pattern in text X, and it is calcuted as a softmax estimates of product W_{ngram} which can be either W_u, W_b, W_t and its corresponding ngram pattern v_{ngram}.

5 Experiment

To analyse and verify the proposed NBOWN model, we used publicly available Amazon Unlocked Mobile [1] and Twitter Airline review dataset,[2] both of the reviews written in English, Amazon Unlocked Mobile consists of review sentences and ratings from 1 to 5, 1 for very negative, 5 for very positive, Twitter Airline consists of review sentences and sentiment labels contains positive, neutral and negative. Both of the reviews in the datasets are short, and contain less than 200 words. We also make available the source code used in our experiments[3].

[1] https://www.kaggle.com/PromptCloudHQ/amazon-reviews-unlocked-mobile-phones/data

[2] https://www.kaggle.com/c/twitter-airlines-sentiment-analysis/data

[3] https://github.com/JingChunzhen/sentiment_analysis/tree/master/nbow

5.1 Data

Rating	Number	Sentiment
1	72337	Negative
2	24728	
3	31765	Neutral
4	61392	Positive
5	223605	

Table 1. Rating Number of Amazon Unlocked Mobile.

Sentiment	Number
Negative	9082
Neutral	3069
Positive	2334

Table 2. Sentiment of Twitter Airline.

In Amazon Unlocked Mobile, as shown in Table 1, the reviews with ratings smaller than or equal to 2 was used as negative examples, greater or equal to 4 as positive examples, 3 as neutral examples.

For training the NBOWN model, we randomly extract 15% of the original training set as the validation set and use remaining 85% as the final training set.

5.2 Word Embedding and Performance Measure

Each sentence was split into tokens using space, all tokens were used to learn the word embedding vectors. We fixed the embedding size to 100, and initialized the embedding layer using pre-trained GloVe, and because the embeddings learned in unsupervised phase contain very little information about sentiment of word [13, 14], since the context for a positive word tends to be very similar to the context of a negative word, to add polarity information to the embeddings, we jointly trained the embeddings and the parameters of the model. Training was performed with the Adam gradient descent algorithm [11]. Additionally, early stopping [22] was used when the validation error starts to increase.

6 Result

6.1 Classification Performance

We used several methodologies to comparision with NBOWN, CNN, bidirectional LSTM, bidrectonal LSTM with Attention, NBOW and NBOW2. Three different window sizes 2, 3, 5 (how many words are considered in one receptive field) was used in CNN, while the number of filters was fixed to 128. For RNN models, the hidden size in LSTM unit was fixed to 128, with attention size to 50, dropout was added for both CNN and RNN models, and dropout rate [25] was set to 0.15.

The maximum number of words in Amazon Unlocked Mobile was set to 164, and in Twitter Airline was 34. Zero paddings were added if the length of the review was shorter than this number, whereas the last words were trimmed if reviews were longer than this number.

Test	Amazon	Twitter
CNN	0.9374	0.7884
bi-LSTM	0.9364	0.7742
bi-LSTM-Attention	0.9450	0.7783
NBOW	0.8853	0.7971
NBOW2	0.8867	0.7856
NBOWN	**0.9147**	**0.8013**

Table 3. The test accuracy between methodology

Table 3 shows the classification accuracies for several models. All the word vectors in the model was initialized by GloVe, and updated during the training. The NBOWN models achieved 91.47% in Amazon Unlocked Mobile, and achieved 80.13% in Twitter Airline. Higher than NBOW and NBOW2 methods. It is worth noting that the CNN and RNN based approaches operate on rich word seqeuence information and have been shown to perform better than NBOW approaches in Amazon Unlocked Mobile dataset. Because the length of reviews in Twitter is much shorter than reviews in Amazon, the RNN based approach didn't achieve very impressive result. And in Twitter Airlines, NBOWN was not far from CNN and LSTM methods.

6.2 Visualization of Words Sentiment

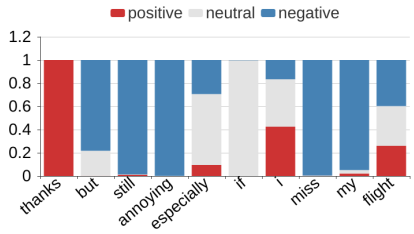

Fig. 2. Sentiment Distribution of Words in Twitter Airline

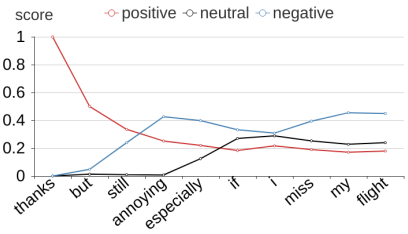

Fig. 3. Sentiment Trend of a Negative Review in Twitter Airline

As Figure 2, Figure 4, Figure 6 show, the positive word thanks, great, awesome, faster and the negative words annoying, miss, awful, blurry can be well captured, Figure 3, Figure 5, Figure 7 show the sentiment trend with the increase of comments as each word comes in. Every point of the curve in Figure 3, Figure 5, Figure 7 is a sentiment score of positive/neutral/negative of the sentence of the current length, and is calculated as follows:

$$S_i = \frac{(i-1) * S_{i-1}}{i} + \frac{\alpha_i * p_i}{i} \tag{15}$$

where S_i represents the positive/neutral/negative score of review of current length i, α_i represents the importance of current ith word, and is calculated by Equation 5, p_i represents the positive/neutral/negative score of the ith word.

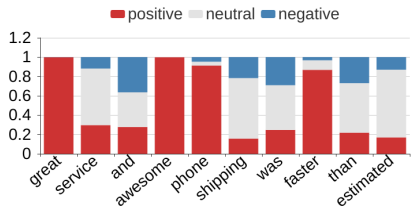

Fig. 4. Sentiment Distribution of Words in Amazon Unlocked Mobile

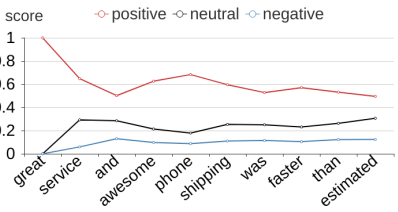

Fig. 5. Sentiment Trend of a Positive Review in Amazon Unlocked Mobile

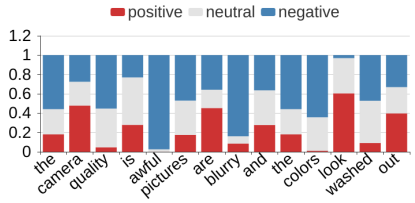

Fig. 6. Sentiment Distribution of Words in Amazon Unlocked Mobile

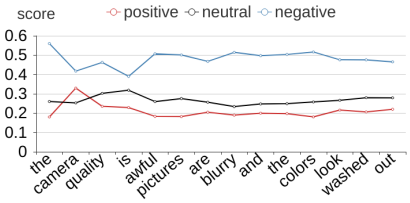

Fig. 7. Sentiment Trend of a Negative Review in Amazon Unlocked Mobile

As Figure 3 shows, at the begining, because of the first word thanks, the sentence level sentiment is positive, the sentiment changes to negative when it encounters the negative word annoying, the negative curve goes a step higher when it encounters another negative word miss. Finally, the sentence level sentiment of this review is negative.

7 Conclusion and Future Work

In this paper, we propose NBOWN, a classification model based on Neural Bag-Of-Words combined with Ngram. Among the BOW methods, we achieved the best results, overall, we have some unique advantages: compared with CNN and RNN models, our model is much less computationally expensive, while the attention mechanism based on RNN model can only identifys the importance of each words in a text, we can successfully get the sentiment polarity of each words.

Although the experimental results were favorable, the current study still has some limitations, which lead us to future research directions. First our proposed method used

a simple ngram pattern which is the mean value of the word vectors and its adjacent word vectors, the order of words was not considered. Second, we simply used a simple space-based token for training word vectors, the classification performance might be improved if more sophisticated preprocessing techiques are performed.

References

1. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *Computer Science*, 2014.
2. Yalong Bai, Wei Yu, Tianjun Xiao, Chang Xu, Kuiyuan Yang, Wei Ying Ma, and Tiejun Zhao. Bag-of-words based deep neural network for image retrieval. In *ACM International Conference on Multimedia*, pages 229–232, 2014.
3. Yoshua Bengio, Rjean Ducharme, Pascal Vincent, Christian Jauvin, Jaz Kandola, Thomas Hofmann, Tomaso Poggio, and John Shawetaylor. *A neural probabilistic language model.* Springer Berlin Heidelberg, 2006.
4. Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Computer Science*, 2014.
5. Junyoung Chung, Caglar Gulcehre, Kyung Hyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *Eprint Arxiv*, 2014.
6. Ronan Collobert and Jason Weston. A unified architecture for natural language processing:deep neural networks with multitask learning. In *International Conference on Machine Learning*, pages 160–167, 2008.
7. Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Meeting of the Association for Computational Linguistics*, pages 49–54, 2014.
8. Yoav Goldberg. A primer on neural network models for natural language processing. *Computer Science*, 2015.
9. Thomas Hofmann. Probabilistic latent semantic analysis. *Proc Uncertainty in Artificial Intelligence*, 41(6):289–296, 2013.
10. Yoon Kim. Convolutional neural networks for sentence classification. *Eprint Arxiv*, 2014.
11. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Computer Science*, 2014.
12. Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. 4:II–1188, 2014.
13. Quanzhi Li, Sameena Shah, Rui Fang, Armineh Nourbakhsh, and Xiaomo Liu. Tweet sentiment analysis by incorporating sentiment-specific word embedding and weighted text features. In *Ieee/wic/acm International Conference on Web Intelligence*, pages 568–571, 2017.
14. Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, 2011.
15. T. Mikolov, W. T. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. *In HLT-NAACL*, 2013.
16. Tomas Mikolov, Armand Joulin, Sumit Chopra, Michael Mathieu, and Marc'Aurelio Ranzato. Learning longer memory in recurrent neural networks. *Computer Science*, 2014.
17. Tomas Mikolov, Martin Karafit, Lukas Burget, Jan Cernock, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September*, pages 1045–1048, 2010.

18. Muhammad Nadeem. *Survey on Opinion Mining and Sentiment Analysis*. Springer US, 2015.
19. Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis:capturing favorability using natural language processing. In *International Conference on Knowledge Capture*, pages 70–77, 2003.
20. Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations Trends in Information Retrieval*, 2(12):1–135, 2008.
21. Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.
22. Lutz Prechelt. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks the Official Journal of the International Neural Network Society*, 11(4):761, 1998.
23. Imran Sheikh, Irina Illina, Dominique Fohr, and Georges Linars. Learning word importance with the neural bag-of-words model. In *The Workshop on Representation Learning for Nlp*, pages 222–229, 2016.
24. Yizhong Wang Junyang Lin Shuming Ma, Xu Sun. Bag-of-words as target for neural machine translation. *ACL*, 2018.
25. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
26. Peng Wang, Jiaming Xu, Bo Xu, Chenglin Liu, Heng Zhang, Fangyuan Wang, and Hongwei Hao. Semantic clustering and convolutional neural network for short text categorization. 2015.
27. Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2017.
28. Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *Eprint Arxiv*, 2014.
29. Matthew D. Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *Eprint Arxiv*, 2013.