Fine-Tuning Word Embeddings for Aspect-Based Sentiment Analysis

Duc-Hong Pham[1,2], Thi-Thanh-Tan Nguyen[2], and Anh-Cuong Le[3(✉)]

[1] Faculty of Information Technology, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
[2] Faculty of Information Technology, Electric Power University, Hanoi, Vietnam
{hongpd,tanntt}@epu.edu.vn
[3] Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh, Vietnam
leanhcuong@tdt.edu.vn

Abstract. Nowadays word embeddings, also known as word vectors, play an important role for many natural language processing (NLP) tasks. In general, these word embeddings are learned from unsupervised learning models (e.g. Word2Vec, GloVe) with a large unannotated corpus and they are independent with the task of their application. In this paper we aim to enrich word embeddings by adding more information from a specific task that is the aspect based sentiment analysis. We propose a model using a convolutional neural network that takes a labeled data set, the learned word embeddings from an unsupervised learning model (e.g. Word2Vec) as input and fine-tunes word embeddings to capture aspect category and sentiment information. We conduct experiments on restaurant review data (http://spidr-ursa.rutgers.edu/datasets/). Experimental results show that fine-tuned word embeddings outperform unsupervisedly learned word embeddings.

1 Introduction

Word representations have now become a critical component of many natural language processing systems. Alternatively, words can be represented as vectors in a semantic space. This approach is also called as distributed representations of words, or word embeddings. The simplest way for this approach is to represent each word in the vocabulary as a one-hot vector, which contains only a one in a position and zeros in all other positions. The dimensionality of these vectors equals to the vocabulary size. Using this technique can overcome the "hand-crafted" drawback of semantic networks (e.g. WordNet), but its vectors still cannot provide useful evidence to evaluate similarity between words. To overcome this limitation, some studies such as [3,8,9] learn word representations based on contexts of words and the achieved results are words with similar grammatical usages and semantic meanings. As the result, each word is represented as a real-valued vector with a low-dimensional and continuous, this known as a word embedding. By this way each word will carry more semantic and grammatical

K. Ekštein and V. Matoušek (Eds.): TSD 2017, LNAI 10415, pp. 500–508, 2017.
DOI: 10.1007/978-3-319-64206-2_56

information, expressing the relationship between the words through the measurement between the vectors, for example, two words "excellent" and "good" are mapped into neighboring vectors in the embedding space. Since word embeddings capture rich linguistic information, they have been used as inputs or a representational basis (i.e. features) for many NLP tasks, such as text classification, document classification, information retrieval, question answering, name entity recognition, sentiment analysis, and so on.

Aspect-based sentiment analysis is a special type of sentiment analysis, it aims to identify the different aspects of entities in textual reviews and the corresponding sentiment toward them. There are some related studies such as aspect term extraction [1,10], aspect category detection and aspect sentiment classification [1,4,5]. Although the context-based word embeddings have been proven useful in many NLP tasks (Collobert et al. [2]), but they can ignore the aspect category and sentiment information when applying to the aspect-based sentiment analysis tasks. The problem here is how to utilize these kinds of information and integrate them into the general word embeddings. We will consider this problem as the task of tuning the general word embeddings towarding the objective of aspect category detection and sentiment analysis classification.

In this paper we consider convolutional neural networks which have achieved remarkably strong results in many studies, such as sentence classification [6], aspect extraction for opinion mining [10]. Inspiring from that, we will propose a model using a convolutional neural network that takes a labeled data set and the learned word embeddings from an unsupervised learning model (e.g. Word2Vec) as input, and fine-tunes the word embeddings to capture aspect category and sentiment information in labeled sentences. In experiment, we evaluate the effectiveness of word embeddings by applying them to two tasks of aspect-based sentiment analysis including aspect category detection and aspect sentiment classification.

2 Related Work

In this section we review existing works closely related to our work. It includes word embedding techniques, word embeddings in sentiment analysis, and aspect-based sentiment analysis.

Word embedding techniques began development in 2003 with the neural network language model (NNLM) in (Bengio et al. [3]). In 2013, Mikolov et al. [8] proposed two models, namely skip-gram and continuous bag-of-words (CBOW). They improve training efficiency and learn high-quality word representation by simplifying the internal structure of the NNLM. In 2014, Pennington et al. [9] proposed the GloVe model using a global log-bilinear regression model that outperforms the original models of Skip-gram and CBOW.

Many studies learned sentiment-specific word embeddings for sentiment analysis, they attempt to capture more information into word embeddings, such as Maas et al. [7] extended the unsupervised probabilistic model to incorporate sentiment information. Tang et al. [12] proposed three neural network models

to learn word vectors from tweets containing positive and negative emoticons. Ren et al. [11] improved the models of Tang et al. [12] by incorporating topic information and learning topic-enriched multiple prototype embeddings for each word. Zhou et al. [15] proposed a semi-supervised word embedding based on the skip-gram model of Word2Vec algorithm, the word embeddings in their model can capture some information as semantic, the relations between sentiment words and aspects.

Some tasks of aspect-based sentiment analysis have been done by using lexical information and classification methods, such as aspect sentiment classifier (Wagner et al. [14]; Ganu et al. [4]), aspect category detection (Ganu et al. [4]; Kiritchenko et al. [5]). They have a limitation that they cannot capture semantic interactions between different words in sentences. Recently, some studies have been used word embeddings as input, such as Alghunaim et al. [1] using CRFsuite or SVM-HMM with word embeddings as features. Poria et al. [10] used word embeddings as input, they then use a deep convolutional neural network to produce local features around each word in a sentence and combine these features into a global feature vector. However, most these studies have been used the learned word embeddings from Word2Vec model (Mikolov et al. [8]) which only capture word semantic relations and ignore supervised information such as aspect category and aspect sentiment, that is our objective in this paper.

3 Proposed Method

Given a set of labeled sentences $D = \{\mathrm{d}_1,\mathrm{d}_2,...,\mathrm{d}_{|D|}\}$ extracting from a collection of reviews in a particular domain (e.g. restaurant), each sentence $d \in D$ is assigned two labels, i.e. aspect sentiment and aspect category. Let k be the number of aspect category labels and m be the number of aspect sentiment labels in D. We denote $a_d \in \mathrm{R}^{\mathrm{k}}$ to be a binary label vector of the aspect categories in the sentence d. Each value in a_d indicates that the sentence d is discussing an aspect category or not. Denoted $o_d \in \mathrm{R}^{\mathrm{m}}$ to be a binary label vector of the aspect sentiments in the sentence d. Each value in o_d indicates that the sentence d is discussing an aspect sentiment or not. Let $V = \{\omega_1, \omega_2, ..., \omega_{|V|}\}$ be a vocabulary and $W \in \mathrm{R}^{\mathrm{nx|V|}}$ be a word embedding matrix, where the i-th column of W is the n-dimensional embedded vector for word i-th, w_i in V. We assume that the matrix W is learned from an unsupervised learning model (e.g. Word2Vec) with a large number of unlabeled sentences. We need to fine-tune word embeddings in matrix W.

In the following, we will present a model using a convolutional neural network to fine-tune word embeddings. The word embeddings are initialized by the learned word embeddings from an unsupervised learning model (e.g. Word2Vec). They then are re-computed to capture aspect category and sentiment information. We named this model as Word Embedding Fine-Tuning (WEFT) model, its architecture is shown in Fig. 1. Our model is similar to the CNN-non-static model of (Kim et al. [6]), but different from it, our model has two output vectors corresponding to aspect category and aspect sentiment information.

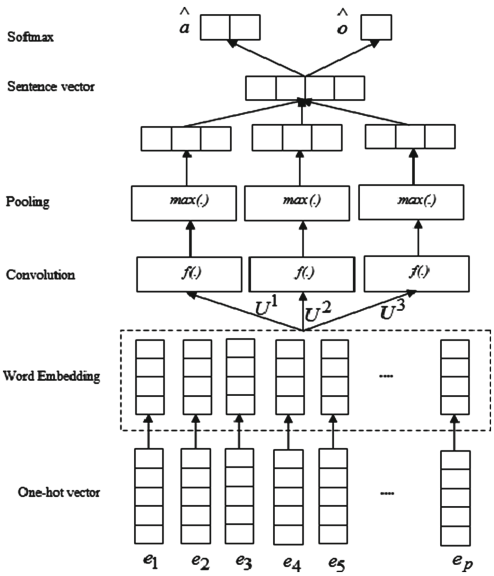

Fig. 1. An illustration of the WEFT model

For a sentence $d \in D$ containing p words, we present each layer with necessary formulations and notations as follows

Word Embedding Layer. The word embedding of word i-th is computed as:

$$x_i = W.e_i \tag{1}$$

where e_i is a one-hot vector of the word i-th in the sentence d.

Convolution Layer. This layer receives $x_1, x_2, ..., x_p$ as input and we use three convolutional filters with widths as 1, 2 and 3 to encode the semantics of unigrams, bigrams and trigrams in the sentence d. For the filter t-th, $1 \leq t \leq 3$, we use the convolution operation to obtain a new vector sequence $y_1^{1t}, y_2^{1t}, ..., y_p^{1t}$ according to the following equation:

$$y_i^{1t} = f(U^t.x_{i:i+h^t-1} + u^t) \tag{2}$$

where $x_{i:i+h^t-1}$ denotes the concatenation of words $x_i, x_{i+1}, ..., x_{i+h^t-1}$, h^t is a filter window size to combine embedding vectors and $f(.)$ is a non-linear function (i.e. $f(y) = \tanh(y) = \frac{e^y - e^{-y}}{e^y + e^{-y}}$). $U^t \in \mathrm{R}^{\mathrm{C} \times \mathrm{h^t.n}}$ and $u^t \in \mathrm{R}^{\mathrm{C}}$ are model parameters, which are learned during training, C is the output dimension.

Pooling Layer. We apply the max pooling operation [2] to mix the varying number of features from the convolution layer into one vector with fixed dimension

$$y^{2t} = [max(y_{i1}^{1t}), max(y_{i2}^{1t}), ..., max(y_{iC}^{1t})] \tag{3}$$

where y_{ij}^{1t} denotes the j-th dimension of y_i^1, $y^2 \in \mathrm{R}^{\mathrm{C}}$ is the output of the filter t-th.

Sentence Vector Layer. In order to obtain the sentence representation vector, this layer concatenates the output vectors of three convolutional filters into one vector as follows $v(d) = [y^{21}, y^{22}, y^{23}]$, $v(d) \in \mathrm{R}^{3\mathrm{C}}$.

Softmax Layer. The aspect category vector $\hat{a}_d$ and the sentiment vector $\hat{o}_d$ are computed as follows: $\hat{a}_d = \mathrm{g}(V^1.v(d) + b^1)$ and $\hat{o}_d = \mathrm{g}(V^2.v(d) + b^2)$, where $V^1 \in \mathrm{R}^{\mathrm{k} \times 3\mathrm{C}}$ and $V^2 \in \mathrm{R}^{\mathrm{m} \times 3\mathrm{C}}$ are weight matrices from the sentence vector layer to the softmax layer, $b^1 \in \mathrm{R}^{\mathrm{k}}$ and $b^2 \in \mathrm{R}^{\mathrm{m}}$ are bias vectors, g is the softmax function.

Model Learning. The cross entropy cost function for the data set D is,

$$E(\theta) = -\sum_{d \in D}(\sum_{i=1}^{k} a_{di} \log \hat{a}_{di} + \sum_{i=1}^{m} o_{di} \log \hat{o}_{di}) + \frac{1}{2}\lambda_\theta \|\theta\|^2 \quad (4)$$

where $\theta = [U^1, U^2, U^3, W, V^1, V^2, u^1, u^2, u^3, b^1, b^2]$ and $\|\theta\|^2 = \sum_i {\theta_i}^2$ is a norm regularization term. In order to compute the parameters θ, we apply back-propagation algorithm with stochastic gradient descent to minimize this cost function.

4 Experiments

4.1 Experimental Data

We use two data sets on the domain of restaurant products: the first data set contains 3,111,239 unlabeled sentences which is extracted from 229,907 reviews[1]. This data set will be used to learn word embeddings. The second data set contains 190,655 labeled sentences, is extracted from 52,574 reviews[2], this data set used in previous work (Ganu et al. [4]; Wang et al. [13]). It contains six aspect category labels as *Price, Food, Service, Ambience, Anecdotes*, and *Miscellaneous*. The four aspect sentiment labels include *Positive, Negative, Neutral* and *Conflict*. Each sentence is labeled for both aspect category and sentiment labels. We randomly get 75% of the given sentences to fine-tune word embeddings, the remaining 25% of given sentences to evaluate the quality of the WEFT model. Some statistics of the second data set are shown in Table 1.

4.2 Implementation of the WEFT Model

We develop a back-propagation algorithm with stochastic gradient descent to minimize the cost function in Eq. (4). We then use it to fine-tune word embeddings with the parameters: the filter window sizes $h^1 = 1$, $h^2 = 2$ and $h^3 = 3$;

[1] https://www.yelp.com/datasetchallenge/.
[2] http://spidr-ursa.rutgers.edu/datasets/.

Table 1. Statistics of the second data set

Aspect	Number of sentences used for	
	Fine-tuning word embeddings	Evaluating word embeddings
Price	4,386	1,462
Food	44,912	14,970
Service	22,470	7,489
Ambience	17,729	5,909
Anecdotes	18,396	6,132
Miscellaneous	35,100	11,700
Total	142,993	47,662

the output dimension C of each filter is 100; the mini-batch size is 60; the regularizations $\lambda_{\mathrm{W}} = \lambda_{\mathrm{U}^1} = \lambda_{\mathrm{U}^2} = \lambda_{\mathrm{U}^3} = 10^{-4}$, $\lambda_{\mathrm{u}^1} = \lambda_{\mathrm{u}^2} = \lambda_{\mathrm{u}^3} = 10^{-5}$, $\lambda_{\mathrm{V}^1} = \lambda_{\mathrm{b}^1} = \lambda_{\mathrm{V}^2} = \lambda_{\mathrm{b}^2} = 10^{-3}$; the weight matrices U^1, U^2, U^3, V^1, V^2 are randomly initialized in the range of $[-1, 1]$; the bias vectors u^1, u^2, u^3, b^1, b^2 are initialized to be zero; the learning rate η is 0.025; the iterative threshold $I = 50$. Matrix W is initialized by the learned word embeddings from an unsupervised learning model (e.g. Word2Vec).

4.3 Evaluation

We evaluate our WEFT model through fine-tuning the learned word embeddings from the models: CBOW, skip-gram of Word2Vec (Mikolov et al. [8]) and GloVe (Pennington et al. [9]). We denote variants of the WEFT model as follows: WEFT-rand: Using randomly initialized word embeddings and then modified during training. WEFT-SG, WEFT-CB and WEFT-GV are three models that fine-tune the learned word embeddings from skip-gram, CBOW and GloVe model. Note that in our work, we use the tools of Word2Vec[3] and GloVe[4] to learn word embeddings, the size of dimensional word embedding is 300 and the window size of context is 4.

In fact, we can not evaluate the word embeddings directly because we can not obtain the ground-truth word embeddings from the given dataset. As a result, we choose to evaluate the word embeddings indirectly through using them as input of a prediction model in two tasks of aspect-based sentiment analysis, i.e. aspect category detection and aspect sentiment classification. Specifically, we use the CNN model (Kim et al. [6]) as the prediction model for these tasks. The lower prediction results in any experiment mean that the word embeddings used in that case are low. In each experiment case, the CNN model performs 4-fold cross validation with 47,662 sentences. The metrics are used to measure the prediction results as F1 score and Accuracy.

[3] https://github.com/piskvorky/gensim/.
[4] https://nlp.stanford.edu/projects/glove/.

Table 2. Results for ACD

Method	F1 score
SG	77.87
CB	78.54
GV	79.19
WEFT-rand	81.43
WEFT-SG	81.50
WEFT-CB	81.76
WEFT-GV	**82.09**

Table 3. Results for ASP

Method	Pos-F1	Neg-F1	Neu-F1	Con-F1	Accuracy
SG	87.05	52.03	65.74	55.46	78.77
CB	86.93	52.25	66.60	55.93	79.22
GV	87.10	51.07	71.02	57.85	80.35
WEFT-rand	88.65	64.18	74.13	56.40	82.15
WEFT-SG	90.87	64.63	73.82	60.23	83.82
WEFT-CB	93.12	64.70	77.03	61.17	84.05
WEFT-GV	93.61	64.77	77.11	61.43	**84.23**

Aspect Category Detection (ACD). In Table 2, we show the achieved F1-score of aspect category detection task for each method. In a general observation, using the fine-tuned word embeddings from the WEFT model gives the better results. This indicates that the fine-tuned word embeddings from the WEFT model is better than other models. Additionally, although the WEFT-rand only captures aspect category and sentiment information in each labeled sentence, but it outperforms CBOW, skip-gram and GloVe model. This shows that the aspect category and sentiment information play an important role in word embeddings.

Aspect Sentiment Prediction (ASP). In Table 3, we show the achieved results of each method. In most cases, using the fine-tuned word embeddings from the WEFT model give the better results. This indicates that the fine-tuned word embeddings from our model helped improve the aspect sentiment predicted results.

Table 4. Each target word is given with its four most similar words using cosine similarity of the vectors determined by each model

	good	bad	food	price
GloVe model	excellent	poor	props	prices
	decent	awful	postings	pricing
	fantastic	horrible	vary	penny
	costco	terrible	gosh	albeit
WEFT-rand model	delight	horrible	snacks	prices
	goodness	alien	restaurant	pricey
	millionaires	calling	hospitality	pricing
	paycheck	poor	fashion	350
WEFT-GV model	excellent	terrible	snacks	prices
	great	lousy	foods	pricing
	wonderful	worse	meal	bills
	bron	poor	variation	pricey

Querying Words. We also evaluate the quality of the word embeddings through querying words. Given some sentiment words or aspect words, we can find out the most similar words with them. In Table 4, we show the interesting results of the most similar words to four given words "good", "bad", "food", and "price". Two models GloVe and WEFT-GV capture broad semantic similarities. The WEFT-rand model captures the true semantic of two aspect words "food" and "price", but it does not capture the good meaning of the two sentiment words "good" and "bad", this is because the WEFT-rand model does not use the semantic of input text. The WEFT-GV model captures the three information kinds, i.e. semantic, aspect category and aspect sentiment, thus it seems to be better than GloVe. This comparison again indicate that adding aspect category and sentiment information into word embeddings is very important.

5 Conclusion

In this paper, we have proposed a model using a convolutional neural network to fine-tune word embeddings for aspect-based sentiment analysis. Word embeddings in our model are initialized by the learned word embeddings from unsupervised learning models and then recomputed to capture aspect category and sentiment information. From experimental results, we have demonstrated that the fine-tuned word embeddings from our WEFT model outperform other word embeddings, which are learned from CBOW, skip-gram and GloVe model.

Acknowledgement. This paper is supported by The Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2014.22.

References

1. Alghunaim, A., Mohtarami, M., Cyphers, S., Glass, J.: A vector space approach for aspect based sentiment analysis. In: Proceedings of NAACL-HLT, pp. 116–122 (2015)
2. Collobert, R., Weston, J.: A unified architecture for natural language processing. In: Proceedings of the ICML, pp. 160–167 (2008)
3. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
4. Ganu, G., Elhadad, N., Marian, A.: Beyond the stars: improving rating predictions using review text content. In: Proceedings of WebDB, pp. 1–6 (2009)
5. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.M.: NRC-Canada-2014: detecting aspects and sentiment in customer reviews. In: Proceedings of SemEval, pp. 437–442 (2014)
6. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of EMNLP, pp. 1746–1751 (2014)
7. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Nguyen, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of ACL, pp. 142–150 (2011)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of Workshop at ICLR (2013)

9. Pennington, J., Socher, R., Manning, C.D.: GloVe global vectors for word representation. In: Proceedings of EMNLP, pp. 1532–1543 (2014)
10. Poria, S., Cambria, E., Gelbukh, A.: Aspect extraction for opinion mining with a deep convolutional neural network. Knowl. Based Syst. **108**, 42–49 (2016)
11. Ren, Y., Zhang, Y., Zhang, M., Ji, D.: Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings. In: AAAI, pp. 3038–3044 (2016)
12. Tang, D., Qin, B., Liu, T.: Learning sentiment-specific word embedding for Twitter sentiment classification. In: Proceedings of ACL, pp. 1555–1565 (2014)
13. Wang, L., Liu, K., Cao, Z., Zhao, J., de Melo, G.: Sentiment-aspect extraction based on restricted boltzmann machines. In: Proceedings of ACL, pp. 616–625 (2015)
14. Wagner, J., Arora, P., Cortes, S., Barman, U., Bogdanova, D., Foster, J., Tounsi, L.: Aspect based polarity classification for semeval task 4. In: Proceedings of SemEval, pp. 223–229 (2014)
15. Zhou, X., Wan, X., Xiao, J.: Representation learning for aspect category detection in online reviews. In: Proceedings of AAAI, pp. 417–423 (2015)