Learning Semantic Representations for Rating Vietnamese Comments

Duc-Hong Pham*, Anh-Cuong Le† and Thi-Kim-Chung Le‡
*University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
†Faculty of Information Technology, Ton Duc Thang University, Vietnam
‡Electric Power University, Hanoi, Vietnam
hongpd@epu.edu.vn, leanhcuong@tdt.edu.vn, chungltk@epu.edu.vn

***Abstract*—Opinion mining and sentiment analysis has recently become a hot topic in the field of natural language processing and text mining. This paper addresses the problem of overall rating for comments in Vietnamese language. The traditional approach of using bag-of-words for feature representation would cause a very high dimensional feature space and doesn't reflect relationship between words. To capture more linguistic information, this paper provides a new neural network model containing three layers: (1) word embedding; (2) comment representation (i.e. comment feature vector); and (3) comment rating prediction. In which, the word embedding layer is designed to learn word embeddings which can capture semantic and syntactic relations between words, the second layer uses a semantic composition model for comment representation, and the third layer is designed as a perceptron and it stands for predicting overall rating of a comment. In experiment, we use a Vietnamese data set which contains comments on the domain of mobile phone products. Experimental results show that our proposed model outperforms traditional neural network models with comment representations based on bag of word model or word vector averaging.**

***Keywords*—Sentiment analysis, Deep learning, Semantic composition model, Neural network, Comment rating prediction**

I. Introduction

The development of World Wide Web technologies has helped people to share their opinions about products and/or services to others. The opinions are mainly expressed by textual reviews/comments on various WWW sources such as social networks, forums, news, etc. These reviews contain useful information for enterprises to improve their products/services and customers to make decision of purchasing. However, the amount of reviews is often very large and grows at a rapid rate. Hence, it becomes very difficult for finding enough necessary information as well as understanding the overall view of opinions. Many studies have been done with the objective of opinion mining and sentiment analysis, such as extracting opinions and classifying sentiment [5], [6], mining comparative sentences from textual reviews [8], [11], extracting and summarizating information from textual reviews [16], [25].

Some other studies focused on predicting review/comment ratings such as [15], [17], [20], [22]. However these studies are to rating reviews/comments based on a bag of words, which have a limitation as it cannot capture semantic interactions between different words in a sentence/comment. This may cause inaccuracies of predicting review/comment ratings.

Identifying features and extracting features are very important in Natural Language Processing (NLP) tasks. Recently, deep learning models can automatically learn feature representations (i.e learning semantic representations) such as learning word vector representations [3], [12], learning sentence/paragraph or document vector representations [14]. However these models are unsupervised deep learning models, they only consider the semantics of words/texts. When applying them to the problem of rating comments, they can ignore several information such as user's ratings on textual comments. This paper provides a new neural network model containing three layers: (1) word embedding; (2) comment representation; and (3) comment rating prediction. Word embeddings at the word embedding layer are learned to capture semantic and syntactic relations between words. Initially they are initialized by the learned word embedding vectors from Word2Vec model [12], they then are recomputed (i.e. updated again) to fit our model, this means that we improve unsupervised word embeddings to supervised word embeddings. The comment feature vector layer uses the semantic composition model [13] with input are word embeddings to learn the representations of comments.

We evaluate the proposed model on the data collected from two websites[1] including 1445 comments on mobile phone products. Experimental results show that our proposed model outperforms the traditional neural network model with comment representations based on bag of word model or word vector averaging.

II. Relate Works

Pang et al., [17] considered the task of review rating prediction as a regression/classification problem. They predicted review ratings by a supervised meta algorithm using a metric labeling formulation of the problem. Siersdorfer et al., [22] studied the influence of sentiment expressed in comments on the ratings of comments by using a SentiWordNet thesaurus and a lexical WordNet-based resource which contains sentiment annotations. Then, in order to compute ratings for textual comments not yet rated, they used support vector machine classification for predicting comment ratings. Qu et al. [21] introduced and used the bag-of-opinion feature to represent

Corresponding author Anh-Cuong Le

[1]http://tinhte.vn and http://sohoa.vnexpress.net/

978-1-4673-8929-7/16/$31.00 ©2016 IEEE

review features. D.H.Pham et al. [19] proposed a neural network model to identify the important degree of aspects from reviews. The result of rating predictions perform in [17], [19], [21], [22] are very much dependent on the selected features from the training data.

Recently, deep learning models have been demonstrated that they can automatically learn feature representations for some sentiment analysis problems such as (Socher et al., [23]; Xu et al., [24]). Sentiment in a sentence/review predicted based on learning representations of texts with different levels (e.g. word, phrase, sentence and document) from using deep learning models. For unsupervised deep learning models, Mikolov et al. [12] learning word feature vector by a context-prediction method. Pennington et al. [18] proposed a weighted least squares model using global word-word co-occurrence counts and thus makes efficient use of statistics, then produces word vectors. Glorot et al. [7] use stacked denoising autoencoder; Le et al. [14] assumed that the paragraph vector is shared across all contexts generated from the same paragraph and they proposed paragraph vector model to learn representation for sentence/paragraph or document. Some supervised deep learning models such as Socher et al. [23] proposed a recursive deep neural networks (RNN), (Kalchbrenner et al. [9]; Kim et al. [10]) proposed deep convolutional networks.

For Vietnamese, several recent studies have been done on sentiment analysis such as Bach et al. [1] proposed a weakly supervised method for sentiment classification and they experimental conducted on two datasets of Vietnamese and Japanese, Bach et al. [2] discovering Vietnamese comparative sentences, they considered identifying comparative sentences as a classification problem, and recognition of relations as a sequence learning problem . To the best of our knowledge, none of the previous studies use machine learning methods for learning representations of Vietnamese sentences/comments or reviews. In this paper, we apply the composition model Hermann et al. [13] into hidden layer of neural network to learn representation of comments.

III. PROPOSED METHOD

In this section, we first introduce word embedding and semantic composition model which will be applied in our model and we then present the problem definition, we then present our proposed model.

1) Word Embedding: Word embedding is an important technical in NLP, it can captures semantic and syntactic relationships between words in documents. Word embeddings (Mikolov et al. [12]) are encoded by column vectors in an embedding matrix $\mathrm{W} \in \mathrm{R}^{\mathrm{kx}|\mathrm{V}|}$, $|V|$ is the size of the word vocabulary V. A word $w \in V$ is transformed into its word embedding x by a linear projection

$$x = \mathrm{W}.e \tag{1}$$

where e is a |V| - dimensional one-hot vector of the word w

2) Semantic Composition Model: The semantic composition model takes word vectors of words in a sentence/document as input, and generate the higher-level representations for a sentence/document. Hermann et al. [13] proposed two models namely ADD and BI, the ADD model computes a sentence representation by the sum of word vectors of all words in its text. For the BI model, let x be a sentence/review including n word vectors/sentence vectors $x_1, x_2, ..., x_n$, the composition function computes the higher-level representation over bi-gram pairs in its words as follows:

$$v(x) = \sum_{i=1}^{n-1} f([x_i + x_{i+1}]) \tag{2}$$

where $f([a + b])$ is element-wise weighted addition of two vectors a and b with a non-linear function (i.e. $f(y) = \tanh(y) = \frac{e^y - e^{-y}}{e^y + e^{-y}}$), $v(x)$ is the vector of the sentence x.

For example, given a sentence *"Bphone tầm giá này khó cạnh tranh (Bphone is difficult to competitive by such price)"*, the semantic representation of this sentence computes as follows: v(Bphone tầm giá này khó cạnh tranh) = $f(v(\text{Bphone}) + v(\text{tầm})) + f(v(\text{tầm}) + v(\text{giá\_này})) + f(v(\text{giá\_này}) + v(\text{khó})) + f(v(\text{khó}) + v(\text{cạnh\_tranh}))$. Figure 1 shows the computation model of this representation.

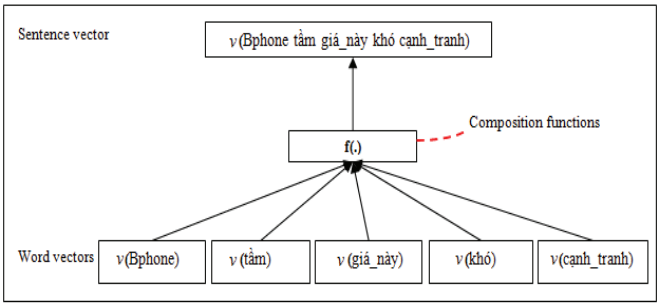


Fig. 1: An example computes semantic representation for the sentence *"Bphone tầm giá này khó cạnh tranh"* is derived from its five word vectors $v(Bphone)$, v(tầm), v(giá_này), v(khó) and v(cạnh_tranh).

In this paper, due to the fact that textual comments are too short, we treat them as sentences and apply the composition model to learn interesting interactions between words in a comment, we use the composition function as follows:

$$v(x) = \sum_{i=1}^{n-1} f(\mathrm{U} \odot [x_i + x_{i+1}] + [\mathrm{u}_0]) \tag{3}$$

where $x_i \in \mathrm{R}^{\mathrm{k}}$, $\mathrm{U} \in \mathrm{R}^{\mathrm{kxk}}$ is the composition matrix, $\mathrm{u}_0 \in \mathrm{R}^{\mathrm{k}}$ is a bias vector and $v(x) \in \mathrm{R}^{\mathrm{k}}$ is the vector of the sentence x. $\odot$ denotes element-wise multiplication

3) Problem Definition: Given a collection of textual comments $D = \{\mathrm{d}_1, \mathrm{d}_2, ..., \mathrm{d}_{|D|}\}$, each textual comment is assigned a numerical rating. Let $V = \{w_1, w_2, ..., w_{|V|}\}$ be a word vocabulary, $\mathrm{W} \in \mathrm{R}^{\mathrm{kx}|\mathrm{V}|}$ be an embedding matrix of words in V and $\mathrm{U} \in \mathrm{R}^{\mathrm{kxk}}$ be a composition matrix which used to map words in a comment into comment representation.

We assume that a comment d contains n words $\{w_{d1}, w_{d2}, ..., w_{dn}\}$, each word w_{di} is represented by a |V|

- dimensional one-hot vector e_{di} and a k - dimensional word embedding vector x_{di}. We denote the comment feature for comment d be a vector $v(d)$ which has the dimension equal to the dimension of word embedding vector, $\beta = (\beta_1,\beta_2,...,\beta_k)$ as a weight vector of comment features. Our goal is to learn for identifying W, U and β.

4) Learning Semantic Representations for Rating Vietnamese Comments (LSRRVCs): This section presents a new neural network model using composition model to learn semantic representations for rating Vietnamese comments.

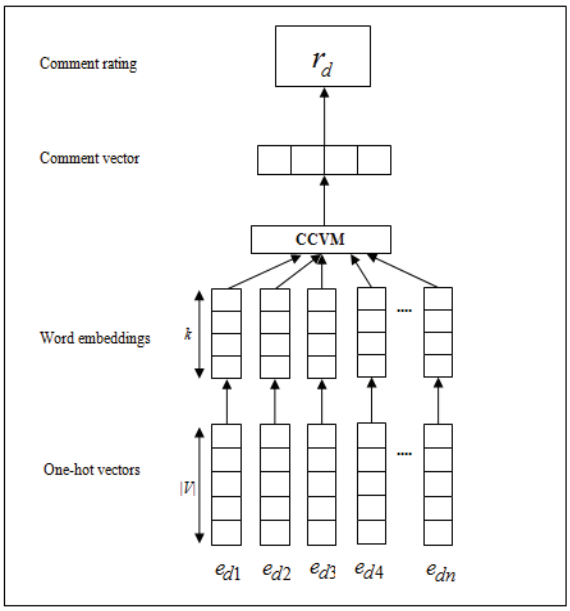


Fig. 2: An illustration of the LSRRCs model, the input are the one-hot word vectors of words in comment d, at the word embedding vector layer, we compute word embedding vectors by mapping each one-hot word vector into k - dimensional vector, then at the comment feature vector layer we compute the comment vector representation by CCVM, finally we compute the rating for comment d.

Unlike a traditional neural network model that takes the identified comment feature vectors from bag of word model as input and then attempts to learn the parameters to predict comment rating. We integrate both learning comment feature representations and learning comment rating predictions into a new neural network model, we call this model as Learning Semantic Representations for Rating Vietnamese Comments Model (LSRRCs).

We show the LSRRCs model as in Figure 2, where corresponds to the comment feature vector layer of a comment we apply a comment composition vector model (CCVM) to compute its representation. Specifically, for a comment d, the embedding word vector of word $w_{di} \in d$ is computed as follows:

$$x_{di} = \mathrm{W}.e_{di} \tag{4}$$

A comment feature representation d is computed by the CCVM model as follows:

$$v(d) = \sum_{i=1}^{n-1} f(\mathrm{U} \odot [x_{di} + x_{d(i+1)}] + [\mathrm{u}_0]) \tag{5}$$

where $\mathrm{u}_0 \in \mathrm{R}^{\mathrm{k}}$ is a bias vector. The rating of comment d is computed as follows:

$$\hat{r}_d = \mathrm{sigm}(v(d).\beta + \beta_0) \tag{6}$$

where $\mathrm{sigm}(y) = 1/(1+e^{-y})$, β_0 is a bias term.
Let r_d be the desired target value of the rating of comment d, the cross entropy cost function for the comment d is as follows:

$$C_d = -r_d \log \hat{r}_d - (1 - r_d) \log(1 - \hat{r}_d) \tag{7}$$

The cross entropy error function (CEE) for the data set D is,

$$E(\theta) = -\sum_{d \in \mathrm{D}} (r_d \log \hat{r}_d + (1 - r_d) \log(1 - \hat{r}_d)) + \frac{1}{2}\lambda_\theta \|\theta\|^2 \tag{8}$$

where $\theta = [\mathrm{W}, \mathrm{U}, \mathrm{u}_0, \beta, \beta_0]$, λ_θ is the regularization parameter 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. Each element of the weights in the parameters θ is updated at time *t + 1* according to the formula:

$$\theta(t+1) = \theta(t) - \eta \frac{\partial E(\theta)}{\partial \theta} \tag{9}$$

where η is the learning rate. Note that we will present the detailed gradient equations in the section Appendix.

The Algorithm 1 presents the steps for learning $\theta = [\mathrm{W}, \mathrm{U}, \mathrm{u}_0, \beta, \beta_0]$.

Algorithm 1 : The algorithm LSRRCs

Input: A collection of textual comments $D = \{\mathrm{d}_1, \mathrm{d}_2, ..., \mathrm{d}_{|D|}\}$, each textual comment $d \in \mathrm{D}$ is given a rating r_d, the learning rate η, the error threshold ε, the iterative threshold *I* and the regularization parameter λ_θ

Step 0: *t=0*; initialize $\mathrm{W}, \mathrm{U}, \mathrm{u}_0, \beta, \beta_0$

Step 1: for *iter=0* to *I* do

for each textual comment $d \in \mathrm{D}$ do

Compute word embeddings at time *t* according to Eq. (4);

Compute comment vector at time *t* according to Eq. (5);

Compute comment rating at time *t* according to Eq. (6);

Update weights in θ at time *t+1* according to Eq. (9);

Step 2: For offline learning, the step 1 may be repeated until the iteration error $\frac{1}{|D|} \sum_{d \in \mathrm{D}} \left| r_d - \hat{r}_d(t) \right|$ is less than the error threshold or the number of iterations have been completed.

Output: $\mathrm{W}, \mathrm{U}, \mathrm{u}_0, \beta, \beta_0$

After obtaining $\theta = [\mathrm{W}, \mathrm{U}, \mathrm{u}_0, \beta, \beta_0]$, given a new collection of textual comments $D_{\mathrm{test}} = \{d_1, d_2, ..., d_{|D_{\mathrm{test}}|}\}$, where each $d \in D_{\mathrm{test}}$ is not assigned a numerical rating . We can

compute the rating of comment d as follows: (a) get each the embedding vector according to equation (4); (b) compute comment representation according to equation (5); (c) compute the rating according to equation (6).

IV. EXPERIMENTS

A. Experimental data

We use the data including 1445 comments on mobile phone products collected from the two websites[2] in which each comment is labeled in the range from 1 star to 5 stars by at least three annotators, then get the the average.

We perform the pre-processing on this data set: we use a standard stop word list as in[3] to remove stop words and use the UETsegmenter[4] to make word segmentation for this data. We then split comments into sentences and obtain 2341 sentences in total and around 1.620 sentences for each comment on average.

Table I: Evaluation Data Statistics

Number of comments	1445
Number of sentences	2341
Average number of sentences in a comment	1.620

B. Parameter Settings and Experimental Results

For word embedding initialization (i.e. matrix W), we additional use 2813 comments unlabeled and these comments also are collected from two websites (i.e. *tinhte.vn* and *sohoa.vnexpress.net*). We then pre-process comments and use the CBOW model (Mikolov et al., 2013) in the tool Word2Vec[5] with the window size of context is 4, the word frequency threshold is 5 (note that ignore all words with total frequency lower than this) and the size of word vector is 300 to learn word embedding vector for each word. After obtaining word embedding vectors, we use them to initialize word embeddings in our LSRRCs model.

For other parameters, we initiate as follows: the learning rate $\eta = 0.015$, the error threshold $\varepsilon = 10^{-4}$, the iterative threshold *I=3000*, the regularizations $\lambda_{\mathrm{W}} = 10^{-4}$, $\lambda_{\mathrm{U}} = \lambda_{\mathrm{u}_0} = 10^{-3}$ and $\lambda_\beta = \lambda_{\beta_0} = 10^{-5}$, all weights in matrix U are randomly initialized in the range of $[-1, 1]$, weights in the vector β are also randomly initialized in the range of $[-1, 1]$, the parameters u_0 and β_0 are initialized to be zero.

We perform the algorithm LSRRCs with parameters are initialized as above. In Table II, we show the comment rating predictions for three comments which we randomly select from our achieved results. Note that the ground-truth comment ratings is in parenthesis. We can see that the result of rating prediction is very close to the value rating in the ground-truth comments.

[2]http://tinhte.vn and http://sohoa.vnexpress.net/

[3]http://seo4b.com/thuat-ngu-SEO/stop-words-la-gi.html

[4]https://github.com/phongnt570/UETsegmenter

[5]https://github.com/piskvorky/gensim/

C. Evaluation

We evaluate the LSRRCs model with three variants as follows:

- **LSRRCs-rand**: This model uses all word embeddings which are randomly initialized and then updated during the training phase.
- **LSRRCs-static**: This model doesn't use one-hot vectors but it uses the learned word embeddings from the continuous bag-of-words model.
- **LSRRCs-non-static**: This model uses the learned word embeddings from Word2Vec as word embedding initialization and then word embeddings are updated during the training phase.

Several other models are compared to the LSRRCs model as follows:

- **Neural network (1 layer)+Bag of words**: In this model, the feature vectors of comments are represented by bag of words model which contains 1524 words. These feature vectors are the inputs for the neural network (1 layer) and we denote this model as "NN+Bag of words".
- **Neural network (1 layer)+Word vector averaging**: In this model, the feature vectors of comments are represented by averaging word embedding vectors (note that the word embeddings learned from Word2Vec). These feature vectors are input for neural network (1 layer) and we denote this model as "NN+Word vector averaging".

We choose the three metrics which are Mean Absolute Error (MAE), Root mean square error (RMSE) and Normalized discounted cumulative gain (nDCG) to measure the differences of comment ratings.

1. MAE is computed as: $MSE = \frac{1}{|D_{test}|} \sum_{d=1}^{|D_{test}|} |r_{d^*} - r_d|$
2. RMSE is computed as: $RMSE = \sqrt{\frac{1}{|D_{test}|} \sum_{d=1}^{|D_{test}|} (r_{d^*} - r_d)^2}$
3. nDCG is computed as: $n\mathrm{DCG} = \frac{DCG}{IDCG}$ where $DCG = \sum_{d=1}^{|D_{test}|} \frac{2^{r_d}-1}{\log(d+1)}$ is that highly relevant comments appearing lower in a search result list should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result. The discounted cumulative gain accumulated at a particular rank position d, $IDCG$ is computed the same as the DCG but it uses comment ratings in the ground-truth.

Where r_{d^*} denotes the ground-truth rating for comment d, r_d denotes the prediction rating for comment d. The smaller MSE or the smaller $RMSE$ value means a better performance, the lager nDCG means a better performance. All models use the same data set (i.e. 1445 comments labeled) and each model is performed in 10 times for training and testing, we then get the average value of metrics to report. In each

Table II: Comment rating prediction for three example comments

Comments	Comment rating
M8 ra rồi vẫn còn khuyết điểm mà HTC không chịu fix , lại đi sản xuất tiếp bản 2 SIM .	2.356 (2.0)
Màu đen nhìn sang vãi các bác nhỉ . anh samsung chắc sẽ lại lăn tăn vụ kim loại này đây Chỉ một từ " Tuyệt "	4.731 (5.0)
Nhà anh samsung chắc đang chạy loạn lên mua nhôm đồng sắt phế rồi Con này mà vỏ bằng kim loại thì Iphone cũng phải chào thua !	3.810 (4.0)

time, we randomly select 75% of given comments to train, the remaining 25% of given comments to test.

Table III: Comparing with other methods

Method	MAE	RMSE	nDCG
Bag of words + NN	0.851	1.003	0.793
Word vector averaging + NN	0.822	0.996	0.802
Our LSRRCs-rand	0.814	0.976	0.811
Our LSRRCs-static	0.772	0.948	0.831
Our LSRRCs-non-static	**0.762**	**0.940**	**0.845**

In Table III, we show the mean achieved value of three metrics for each method, we can see that our LSRRCs-non-static model performs better than other models on all three metrics. Our LSRRCs-static model performs better than NN+Bag of words model and the NN+Word vector averaging model on MAE and RMSE. In additional, the NN+Word vector averaging model also performs better than NN+Bag of words model on MAE and RMSE, these indicate that the feature vectors of comments are represented based on word vectors better than based on bag of words model. They also indicate that the feature vectors of comments are learned based on composition model using the learned word embedding vectors from word2vec better than Word vector averaging, word embedding vectors randomly.

In order to evaluate in greater detail the influence of the dimensional word vector on comment rating predictions. We chose the size of dimensional word vector is from 200 to 400 and perform to predict comment rating in each these. Figure 3 and Figure 4 show metrics achieved. The left Figure 3 shows performance comparison with Mean Absolute Error and the right Figure 3 shows performance comparison with Root mean square error, we see that our LSRRCs-non-static performs slightly better than other models in all cases. However all three models: LSRRCs-non-static, LSRRCs-static and LSRRCs-rand perform best with the dimensional word vector =300, they do not improve by increasing/decreasing the dimensionality (i.e. increasing from 350 to 400; decreasing from 250 to 200). This means that increasing/decreasing the dimensionality of the word vectors does not necessarily lead to better results.

V. CONCLUSION

In this paper, we have proposed a new neural network model using compositional model to learn semantic representations for rating Vietnamese comments. Through experimental results, we have demonstrated that our proposed LSRRCs (i.e. three variants: LSRRCs-rand, LSRRCs-static and and LSRRCs-non-static) performs slightly better than NN+Bag of words model and NN+Word vector averaging model on

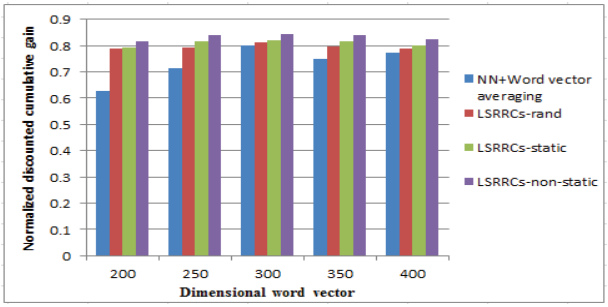


Fig. 4: Normalized discounted cumulative gain

all three metrics, they indicate that the feature vectors of comments are learned based on composition model better than word vector averaging or bag of words. In addition, the experimental results have showed the important role of pre-training of word embedding vectors in learning comment feature representations.

In the future, we aim at studying the comment rating predictions based on aspects which assume that have some aspects mentioned in Vietnamese comments.

ACKNOWLEDGMENTS

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

REFERENCES

[1] N.X. Bach, T.M. Phuong. Leveraging User Ratings for Resource Poor Sentiment Classification. In *Proceedings of the 19th International Conference on Knowledge-Based and Intelligent Information Engineering Systems (KES)*, pp. 322-331, 2015.

[2] N.X. Bach, P.D. Van, N.D. Tai, T.M. Phuong. Mining Vietnamese Comparative Sentences for Sentiment Analysis. *In Proceedings of KSE*, pp. 162-167, 2015.

[3] Y. Bengio, R. Ducharme and P. Vincent. Neural Probabilitistic Language Model. In *Journal of Machine Learning Research 3*, pp. 1137-1155, 2003.

[4] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuglu and P. Kuksa. Natural Language Processing (Almost) from Scratch. In *Journal of Machine Learning Research*, pp. 2493-2537, 2011.

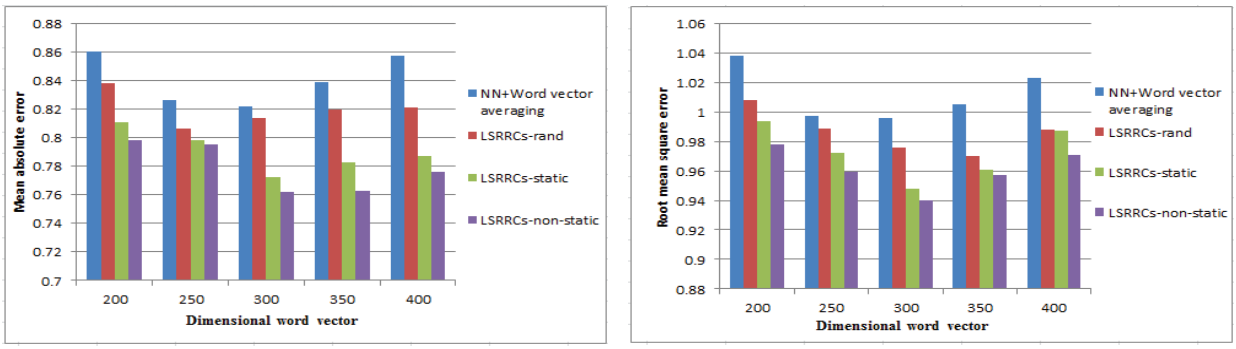
[5] K. Dave, S. Lawrence and D.M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of WWW*, pp. 519-528, 2003.

[6] A. Devitt and K. Ahmad. Sentiment polarity identification in financial news: A cohesion-based approach. In *Proceedings of ACL*, pp. 984-991, 2007.

[7] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification:a deep learning approach. In *Proceedings of ICML*, pp. 513-520, 2011.

[8] N. Jindal and B. Liu. Identifying comparative sentences in text documents. In *Proceedings of SIGIR*, pp. 244-251, 2006.

[9] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A sentence model based on convolutional neural networks. In *Proceedings of ACL*, 2014.



(a) Mean absolute error

(b) Root mean square error

Fig. 3: Performance comparison with Mean absolute error and Root mean square error

[10] Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, pp. 1746-1751, 2014.

[11] H. Kim and C. Zhai. Generating Comparative Summaries of Contradictory Opinions in Text. In *Proceedings of CIKM*, pages 385-394, 2009.

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS*, pp. 1-9, 2013.

[13] K. Moritz Hermann and P. Blunsom. Multilingual models for compositional distributed semantics. In *Proceedings of ACL*, pp. 58-68, 2014.

[14] Q.V. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of ICML*, pp. 1188-1196, 2014.

[15] F. Li, N. Liu, H. Jin, K. Zhao, Q. Yang, and X. Zhu. Incorporating Reviewer and Product Information for Review Rating Prediction. In *Proceedings of IJCAI*, pp.1820-1825, 2011.

[16] B. Liu, M. Hu and J. Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of WWW*, pp. 342-351, 2005.

[17] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pp. 115-124, 2005.

[18] J. Pennington, R. Socher, and C.D. Manning. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pp. 1532-1543, 2014.

[19] D.H. Pham and A.C. Le and T.K.C Le. A Least Square based Model for Rating Aspects and Identifying Important Aspects on Review Text Data. In *Proceedings of NICS*, pp. 16-18, 2015.

[20] D.H. Pham, A.C. Le. A Neural Network based Model for Determining Overall Aspect Weights in Opinion Mining and Sentiment Analysis. In *Indian Journal of Science and Technology*, Volume 9, Issue 18, pp. 1-6, 2016.

[21] L. Qu, G. Ifrim, and G. Weikum. The bag-of-opinions method for review rating prediction from sparse text patterns. In *Proceedings of COLING*, pp. 913-921, 2010.

[22] S. Siersdorfer, S. Chelaru, W. Nejdl, J. San Pedro. How useful are your comments?: analyzing and predicting youtube comments and comment ratings. In *Proceedings of WWW*, pp. 891-900, 2010.

[23] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, and C.Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pp. 1631-1642, 2013.

[24] L. Xu, K. Liu, and J. Zhao. Joint opinion relation detection using one-class deep neural network. In *Proceedings of COLING*, pp. 677-687, 2014.

[25] L. Zhuang, F. Jing, and X. Zhu. Movie review mining and summarization. In *Proceedings of CIKM*, 2006, pp. 43-50.

APPENDIX

We present the detailed gradient equations as follows:

The gradient of $E(\theta)$ according to β_i is,

$$\frac{\partial E(\theta)}{\partial \beta_i} = \sum_{d=1}^{|D|} \frac{\partial E(\theta)}{\partial r_d} \cdot \frac{\partial r_d}{\partial \beta_i} = -\sum_{d=1}^{|D|} (r_d - \hat{r}_d).v(d)_{di} + \lambda_\beta.\beta_i; 1 \le i \le k \quad (10)$$

The gradient of $E(\theta)$ according to β_0 is,

$$\frac{\partial E(\theta)}{\partial \beta_0} = \sum_{d=1}^{|D|} \frac{\partial E(\theta)}{\partial r_d} \cdot \frac{\partial r_d}{\partial \beta_0} = -\sum_{d=1}^{|D|} (r_d - \hat{r}_d) + \lambda_{\beta_0}.\beta_0 \quad (11)$$

The gradient of $E(\theta)$ according to $\mathrm{u_{ij}}$ is,

$$\frac{\partial E(\theta)}{\partial \mathrm{u_{ij}}} = \sum_{d=1}^{|D|} \frac{\partial E(\theta)}{\partial r_d} \cdot \frac{\partial r_d}{\partial v(d)_{di}} \cdot \frac{\partial v(d)_{di}}{\partial \mathrm{u_{ij}}}$$

$$= -\sum_{d=1}^{|D|} (r_d - \hat{r}_d).\beta_i. \sum_{l=1}^{n-1} [(x_{dlj} + x_{d(l+1)j}).(1 - p_l^2)] + \lambda_{\mathrm{U}}.u_{ij} \quad (12)$$

The gradient of $E(\theta)$ according to $\mathrm{u_{0i}}$ is,

$$\frac{\partial E(\theta)}{\partial \mathrm{u_{0i}}} = \sum_{d=1}^{|D|} \frac{\partial E(\theta)}{\partial r_d} \cdot \frac{\partial r_d}{\partial v(d)_{di}} \cdot \frac{\partial v(d)_{di}}{\partial \mathrm{u_{0i}}} = -\sum_{d=1}^{|D|} (r_d - \hat{r}_d).\beta_i. \sum_{l=1}^{n-1} (1 - p_l^2) + \lambda_{\mathrm{u_0}}.u_{0i} \quad (13)$$

Where $1 \le i \le k$ and $1 \le j \le k$ are indices of rows and columns in matrix U, $p_l = tanh(\sum_{j=1}^{k} u_{ij}.(x_{dlj} + x_{d(l+1)j}) + \mathrm{u_{0i}})$.

The gradient of $E(\theta)$ according to the j-th element $\mathrm{w_{dlj}}$ of the word embedding vector $\mathrm{x_{dl}}$ in matrix W is,

$$\frac{\partial E(\theta)}{\mathrm{w_{dlj}}} = \sum_{d=1}^{|D|} \frac{\partial E(\theta)}{\partial r_d} \cdot \frac{\partial r_d}{\partial v(d)_{di}} \cdot \frac{\partial v(d)_{di}}{\partial \mathrm{x_{dlj}}} \cdot \frac{\partial \mathrm{x_{dlj}}}{\partial \mathrm{w_{dlj}}} = -\sum_{d=1}^{|D|} (r_d - \hat{r}_d).\beta_i.(2 - p_1^2 - p_2^2).\mathrm{u_{ij}} + \lambda_{\mathrm{W}}.\mathrm{w_{dlj}} \quad (14)$$

where $p_1 = tanh(\sum_{j=1}^{k} u_{ij}.(x_{d(l-1)j} + x_{dlj}) + \mathrm{u_{0i}})$, $p_2 = tanh(\sum_{j=1}^{k} u_{ij}.(x_{dlj} + x_{d(l+1)j}) + \mathrm{u_{0i}})$; $2 \le l \le n$, $1 \le i \le k$ and $1 \le j \le k$