Multi-channel LSTM-CNN model for Vietnamese sentiment analysis

Conference Paper · October 2017

DOI: 10.1109/KSE.2017.8119429

Huy-Tien Nguyen
Ho Chi Minh City University of Science
38 PUBLICATIONS **398** CITATIONS

SEE PROFILE

Bac Le
Ho Chi Minh City University of Science
146 PUBLICATIONS **2,517** CITATIONS

SEE PROFILE

2017 9th International Conference on Knowledge and Systems Engineering(KSE)

Multi-channel LSTM-CNN model for Vietnamese sentiment analysis

Quan-Hoang Vo*, Huy-Tien Nguyen†, Bac Le* and Minh-Le Nguyen†
*Department of Computer Science,
University of Science, VNU-HCM, Vietnam
†School of Information Science,
Japan Advanced Institute of Science and Technology (JAIST)

vo.hoang.quan@outlook.com, ntienhuy@jaist.ac.jp,
lhbac@fit.hcmus.edu.vn, nguyenml@jaist.ac.jp

***Abstract*—Convolutional neural network (CNN) and Long Short Term Memory (LSTM) have shown the state of the art results for sentiment analysis in English corpus. However, there are not many studies of this approach for Vietnamese corpus. In our work, CNN and LSTM are employed to generate information channels for Vietnamese sentiment analysis. Because each deep learning model (e.g. CNN, LSTM) has a particular advantage, this scenario provides a novel and efficient way for integrating the advantages of CNN and LSTM. In addition, we introduced a Vietnamese corpus, which collected comments/reviews from Vietnamese commercial web pages and was annotated by three human annotators. We evaluated our approach on our corpus and VLSP corpus. According to the experimental results, the proposed model outperforms SVM, LSTM, and CNN on the two datasets.**

I. Introduction

In the community of natural language processing, sentiment analysis is a fundamental task and has attracted a huge amount of research in recent years[1][2]. Millions of comments on social networks such as Facebook, Twitter and so on are shared by users. An important information to be analyzed from those comments is opinions/sentiments, which express subjective opinions of particular users. Sentiment analysis includes subjectivity classification which labels a given text as either subjective or objective and sentiment classification which classifies a subjective text as positive, negative or neutral[1].

Recently, deep learning models are applied successfully for NLP tasks, especially sentiment analysis such as CNN [27], [28], [29], LSTM [30], [34]. The CNN model employs convolutional filters to capture local relationships between neighbor words in a sentence but fails for long-distance dependencies. In the other hand, LSTM can handle CNN's limitation by using inner cells for memorizing information for a long period of time. For this reason, we proposed an approach taking the advantages of these methods. In our work, CNN and LSTM are employed to construct two information channels. These channels are expected to enhance the classification performance of the softmax layer (details in Figure1). To evaluate our model for Vietnamese corpus, we compared our model with LSTM, CNN, SVM on two datasets VS and VLSP. In VS dataset, we collected 17,500 reviews/comments from Vietnamese e-commercial sites (i.e. TinhTe.vn, Tiki.vn, etc.) and labeled for positive/negative/neutral by three annotators. According to the experimental results, our method outperforms the other methods on both of datasets. The main contributions of this work are as follows:

- We proposed a multi-channel LSTM-CNN model for Vietnamese sentiment analysis. This approach integrates the advantages of CNN and LSTM into one model. Our model outperforms the individual models: CNN, LSTM on Vietnamese datasets.
- We built a Vietnamese sentiment (VS) corpus containing 17,500 reviews from Vietnamese e-commercial sites, which are labeled manually for positive/negative/neutral by three annotators.

II. Related work

Sentiment analysis is the field of study which analyzes people's opinions, attitudes, and emotions toward to entities. In practice, opinion mining is a challenging task. Taboada assigned sentiment labels to text by extracting sentiment-bearing words[3]. Bing Liu formulated the sentiment analysis task as a classification task and applied supervised machine learning techniques for this problem[2]. In this approach, dominant research concentrated on designing effective features such as word ngram[4], emoticon[5], sentiment words[6]. However, it takes a great effort to design handcraft features. Recently, the study of deep learning models has provided an efficient way to learn continuous representation vectors for sentiment classification. Bengio and Mikolov proposed a presentation of learning techniques for semantic word representation[7][8]. The authors generated word embedding vectors carrying semantic meanings by using a neural network in the context of a word prediction task. Embedding vectors of words are close to each other if they share similar meanings. In different contexts, the semantic information possibly determines opposite opinions. Consequently, there have been many studies of learning sentiment specific word representation by employing sentiment text[10][17][22]. For sentence and document level, composition approach attracted many studies. Yessenalina and

978-1-5386-3576-6/17/$31.00 ©2017 IEEE

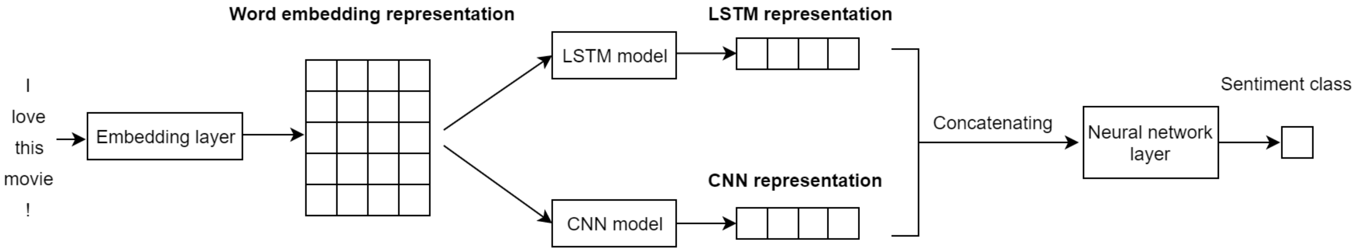

Figure 1. Multi-channel LSTM-CNN model

Cardie modeled each word as a matrix and used iterated matrix multiplication to present a phrase [15]. Deep recursive neural networks (DRNN) over tree structures were employed to learn sentence representation for sentiment classification such as DRNN with binary parse trees[16], Recursive tensor neural network with sentiment treebank[17]. Convolutional neural network (CNN) has recently been applied efficiently for semantic composition[18][19][20]. By using convolutional filters to capture local dependencies in term of context windows and applies a pooling layer to extract global features. Le and Mikolov applied paragraph information into the word embedding technique to learn semantic document representation[21]. Tang used CNN or LSTM to learn sentence representation and encoded these semantic vectors in document representation by Gated recurrent neural network[22]. Zhang[31] proposed Dependency Sensitive CNN to build hierarchically textual representations by processing pretrained word embeddings. Wang[32] used a regional CNN-LSTM to predict the valence arousal ratings of texts.

In Vietnamese text, Kieu and Pham[13] proposed a rule-based system for Vietnamese sentiment classification using the Gate framework and describe experiments on a corpus of computer product reviews. Unlike the work of Kieu and Pham, Duyen [14] took machine learning approaches to examine the task. The author also investigated the impact of the overall score of a review on classifying the sentiment of a sentence. Machine learning has been shown to have several advantages over a rule-based approach. Kieu and Pham, 2010[2] proposed opinion analysis system for "computer" product in Vietnamese reviews using rule-based method for constructing automatic evaluation of users' opinion at sentence level. However, this system could not detect implicit features which occur in sentences without feature words as same as considered for features in only one sentence. Phan and Cao[33] used Skip-gram word estimation model with SVM-based classification for opinion mining Vietnamese food places text reviews.

As we mentioned above, there is not much research focusing on deep learning models for Vietnamese sentiment analysis. One of the reasons is that deep learning models require a large training data. In our work, a Vietnamese corpus with 17500 reviews is annotated. By examining various deep learning models, we proposed a multi-channel LSTM-CNN model for Vietnamese sentiment analysis, which gives a better performance than CNN, LSTM.

III. Background

A. Long Short Term Memory model

In LSTM architecture, Hochreiter[23] designs a memory cell which preserves its state over a long period of time and non-linear gating units regulating information flow into and out of the cell. By employing this memory cell, LSTM has the ability to capture efficiently long distance dependencies of sequential data without suffering the exploding or vanishing gradient problem of Recurrent neural network[25].

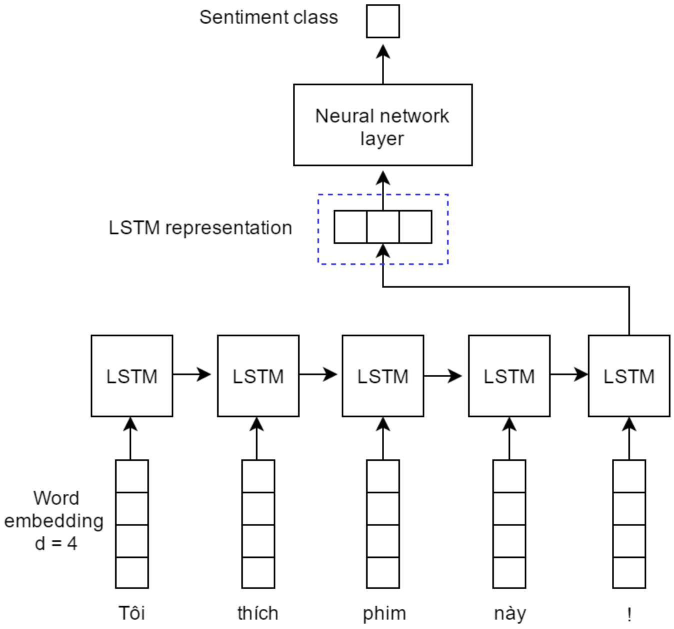

Figure 2. Illustration of our LSTM model for sentiment classification. Each word is transfered to a 4 dimension vector and then fed to LSTM model.

Sentences of variable length are transformed to fix-length vectors by recursively applying a LSTM unit to each input word x_t of sentences and the previous step h_{t-1}. At each time step t, the LSTM unit with l-memory dimension defines 6 vectors in $\mathbb{R}^l$: input gate i_t, forget gate f_t, output gate o_t, tanh layer u_t, memory cell c_t and hidden state h_t as follows

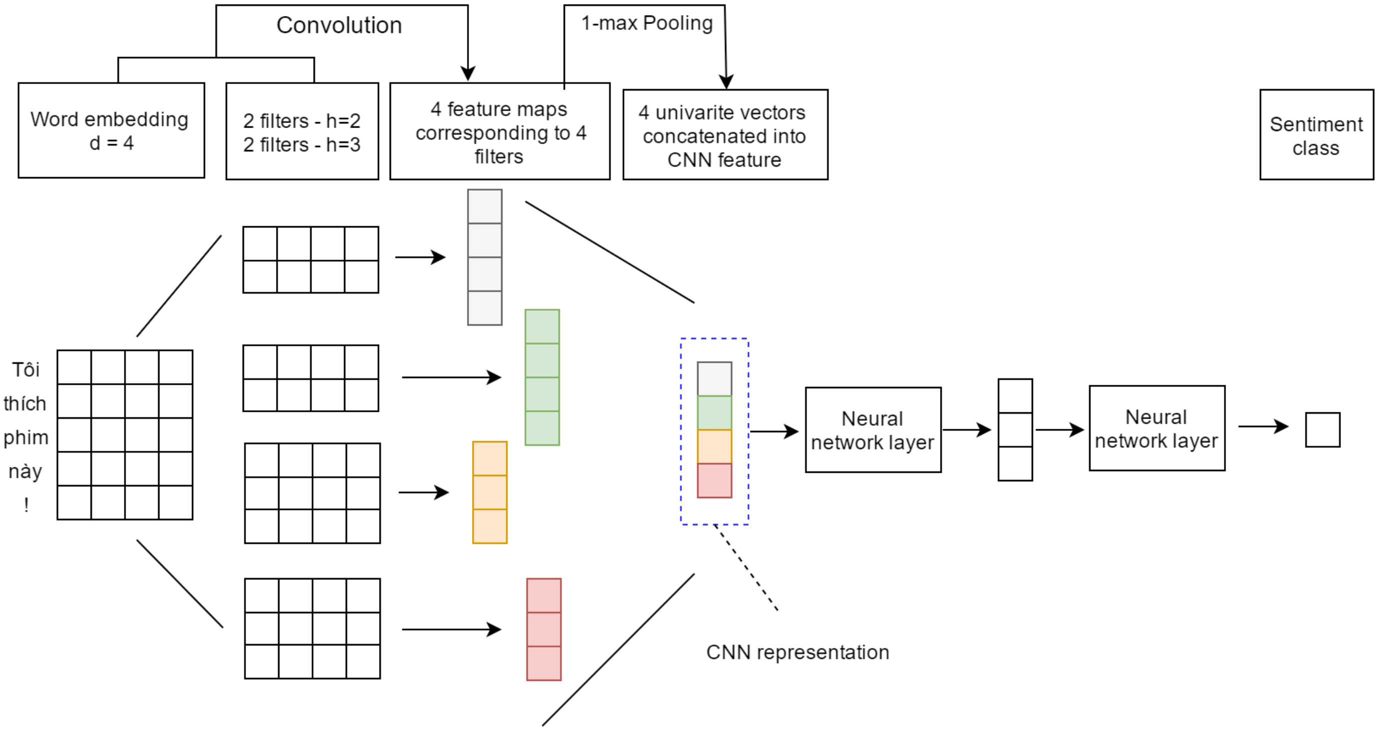

Figure 3. Illustration of our CNN model for sentiment analysis. Given a sequence of d-dimension word embeddings ($d = 4$), the model applies 4 filters: 2 filters for region size $h = 2$ and 2 filters for region size $h = 3$ to generate 4 feature maps. Afterward, 1-max pooling operator is applied to extract the largest value in each feature map. These values are concatenated into a CNN feature and passed to a neural network layer to synthesize a high level feature. Finally, this high level feature is used as input to the last neural network layer for sentiment classification.

(from Tai[24]):

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{1}$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$
$$u_t = \tanh(W_u x_t + U_u h_{t-1} + b_u) \tag{4}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot u_t \tag{5}$$
$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

where $\sigma, \odot$ respectively denote a logistic sigmoid function and element-wise multiplication; W_i, U_i, b_i are respectively two weights matrices and a bias vector for input gate i. The denotation is similar to forget gate f, output gate o, tanh layer u, memory cell c and hidden state h. Intuitively, the forget gate makes a decision of which previous information in the memory cell should be forgotten, while the input gate controls what new information should be stored in the memory cell. Finally, the output gate decides the amount of information from the internal memory cell should be exposed. These gate units help a LSTM model remember significant information over multiple time steps. Figure 2 explains how to employ the LSTM architecture for memorizing sentiment information over sequential data.

B. Convolutional neural network

We present a sentence of length s as a matrix $d \times s$, where each row is a d-dimension word embedding vector of each word. Given a sentence matrix **S**, CNN performs convolution on this input via linear filters. A filter is denoted as a weight matrix W of length d and region size h. W will have $d \times h$ parameters to be estimated. For an input matrix $S \in \mathbb{R}^{d \times s}$, a feature map vector $O = [o_o, o_1, ..., o_{s-h}] \in \mathbb{R}^{s-h+1}$ of the convolution operator with a filter W is obtained by applying repeatedly W to sub-matrices of S:

$$o_i = W \cdot S_{i:i+h-1} \tag{7}$$

where $i = 0, 1, 2, ..., s - h$, $(\cdot)$ is dot product operation and $S_{i:j}$ is the sub-matrix of S from row i to j.

Each feature map O is fed to a pooling layer to generate potential features. The common strategy is 1-max pooling [26]. The idea of 1-max pooling is to capture the most important feature v corresponding to the particular feature map by selecting the highest value of that feature map:

$$v = \max_{0 \leq i \leq s-h} \{o_i\} \tag{8}$$

We have described in detail the process of one filter. Figure 3 shows an illustration of applying multiple filters with variant region sizes to obtain multiple 1-max pooling values. After pooling, these 1-max pooling values from feature maps are concatenated into a CNN feature. Intuitively, the CNN feature is a collection of maximum values from the feature maps. To make a connection to these values, we provide a NN layer to synthesize a high level feature from the CNN feature.

Finally, this high level feature is passed to a NN layer with sigmoid activation to generate the probability distribution over sentiment labels.

IV. Multi-channel LSTM-CNN model

We could separate a deep learning network into two parts: (i) **Building feature representation** - this part encodes target information into representation vectors; (ii) **Classifying layer** - Given the representation vectors constructed by the first part, this part tries to learn a layer (or a boundary) for classifying them into target labels. Given an input, each deep learning model has its own way to capture target information into feature vectors. In particular, CNN model uses convolutional filters to capture local dependencies between neighbor words. However, the limitation of filter lengths makes CNN model hard to learn overall dependencies of a whole sentence/document. We could consider a representation vector constructed by CNN as concatenating local relationship values. In LSTM model, a memory cell is introduced to preserve information over a long period of time. As a result, a feature vector constructed by LSTM carry overall dependencies of a whole sentence/document. We expect those two vectors (one carrying local relationship and one carrying overall relationship) to support well each other for enhancing the classification performance.

Given an input, our proposed model generates a word embedding representation by an embedding layer. Then this embedding representation is fed to LSTM model for generating a LSTM feature vector and to CNN model for generating a CNN feature vector. These vectors as two information channels are concatenated and fed into a neural network for classifying. Figure 1 visualizes our model. The proposed architecture improves the performance by capturing both local and global dependencies of a sentence/document. In the training phase, the algorithm for first-order gradient-based optimization AdaMax is used to learn model parameters. Details of AdaMax method can be found in [35].

Table I
Statistic summary of datasets. $|V|$ is vocabulary size.

Dataset	*positive*	*neutral*	*negative*	*total*	$\|V\|$
VS	5,988	5,573	5,939	17,500	21,540
VLSP	1,700	1,700	1,700	5,100	13,931

V. Experiment

A. Dataset

We evaluated the proposed model on two Vietnamese datasets **VS** and **VLSP**[1]. For splitting a dataset into a train set and a test set, we applied 5-fold cross-validation on both of datasets. Table I shows the statistic summary of the datasets.

- In **VS** dataset, we collected about 17,500 comments/reviews for various products (i.e. books, laptops, foods, phones, etc.) from Vietnamese e-commercial sites (i.e. TinhTe.vn, Lazada.vb, Tiki.vn, etc.). These reviews were labeled for overall sentiment polarity (positive, neutral, negative) by three annotators. The inter-rater agreement Cohen's Kappa score for each pair of annotators is more than 0.74.
- **VLSP** dataset was provided by VLSP 2016 campaign for sentiment analysis. The dataset contains only real data collected from social media.

Table II
Experimental results on VS and VLSP datasets. Accuracy metric is used for evaluation. "*With token*" denotes using Word segmentation for text preprocessing

Method	VS		VLSP	
	With token	Without token	With token	Without token
SVM	77.87	87.21	56.27	54.57
LSTM	78.98	85.83	56.63	50.9
CNN	81.49	86.66	59.02	55.84
Our proposed	81.86	**87.72**	**59.61**	56.01

B. Experimental setup

To tune hyper-parameters of our model, we do a grid search on 30% of each dataset. As a result, we obtain these hyper-parameters as follows:

- For CNN, we used 3 region sizes of $3, 5, 7$; the number of each region size is 150, the dimension of penultimate NN layer is 100 and the size of embedding layer is 200.
- For LSTM, the LSTM layer has $d = 128$ and the size of embedding layer is 200.

Table III
The proposed model's performance on each class.

Dataset	*Class*	*Precision*	*Recall*	*F1*
VS without token	Positive	0.92	0.9	0.91
	Neutral	0.81	0.89	0.85
	Negative	0.9	0.828	0.864
VLSP with token	Positive	0.622	0.742	0.676
	Neutral	0.534	0.476	0.5
	Negative	0.632	0.57	0.598

C. Results & discussion

For evaluation, we compared the proposed model against SVM with Bag of Word (BOW) feature, LSTM and CNN.

[1] $http://vlsp.org.vn/evaluation\_campaign\_OM$

Table IV
SOME TYPICAL EXAMPLES OF THE WRONG PREDICTION.

No.#	Review	True label	Predicted label
1	*ti vi dep qua*	Positive	Negative
2	*Về hình thức màu sắc khá đẹp lượng nước chứa vừa đủ với nhà có diện tích nhỏ ... mua về vợ rất thích tuy nhiên cây lau hơi yếu đến giờ thì mình đã phải mua cây khác thay thế vì bị gãy. Ngoài nhược điểm đó ra thì mình thấy khá ổn ở mức giá của tiki*	Positive	Neutral
3	*Sản phẩm này giá khá cao nên mình tranh thủ canh giảm giá mới dám mua. Ấn tượng khi nhận hàng cũng không thích lắm vì vỏ son màu vàng nhợt mình cũng không tìm thấy hạn sử dụng cũng như không thấy ghi chi tiết trên vỏ son. Không biết loại này có chì hay không nữa.*	Neutral	Positive
4	*máy dùng tương đối tốt ổn định. chán cái là phụ kiên không đầy đủ như nhà sản xuất. hộp đựng không phải của máy. ba nhãn nhà nhập khẩu thì ba thông tin khác nhau*	Neutral	Positive
5	*Toi moi mua 1cai de dung nhung moi dc may hom bep da bi lứt vòm lửa cháy bập bùng rất sợ nếu dc chọn lại tôi sẽ k mua bếp này.mua hàng qua mang hên sui...*	Negative	Neutral
6	*Thiết kế đẹp nam tính cảm biến vân tay tiện lợi cảm ứng nhạy. Máy nóng pin trung bình hiệu năng chưa được tối ưu tốt camera không thật sự như quảng cáo với những ưu và khuyết điểm như trên thì giá bán ra không hợp lý (quá mắc).*	Negative	Neutral

Word segmentation[36] was also applied for text preprocessing. Table II shows the results of our experiments. According to the results, our models outperform the other methods on the two datasets. In our observation, the text of VS dataset is quite informal and contains many grammar mistakes. These factors affect seriously to the performance of segmenting word as well as analyzing sentiment in VS dataset. In the empirical results, our model gives a better performance than LSTM and CNN. That proves the efficiency of our approach which takes the advantages of LSTM and CNN.

To evaluate the performance of our model on each class, we measured precision, recall and F1 scores for each class on two datasets. Table III shows the results of this experiment. According to these scores, we observed that the performance on positive class and negative class is better than neutral class. Intuitively, it is difficult to rate a neutral comment because the opinions are inclined to be negative or positive. In addition, a review can contain both positive and negative opinions; however, the overall sentiment is neutral. These reasons make the sentiment analysis on neutral class be more difficult than the other classes.

D. Error analysis

To evaluate the limitation of the proposed model, we manually inspect some cases of the wrong prediction, which are showed in table IV. These reviews are typical examples of the proposed model's weakness.

The first reason is that a word missing diacritical marks could be ambiguous. In example #1, the word "dep" was written without diacritical marks. This word could be understood as "dẹp"(sometimes carrying negative sentiment) or "đẹp" (mostly carrying positive sentiment). This fact gives a challenge to our model. To improve our model, we believe that some text preprocessing techniques and syntactic information should be applied to avoid ambiguous cases.

The second source of the wrong prediction comes from reviews containing both negative sentiment and positive sentiment. According to our observation, the model has a tendency to assign neutral labels to those reviews. This fact is even difficult for humans to decide sentiment polarity. To address this problem, the model needs to evaluate the intensity of each sentiment polarity. From these intensity scores, an overall sentiment is generated.

VI. CONCLUSION

In this work, we introduced a novel model which integrates the advantages of CNN and LSTM. This approach captures both local and global dependencies in a sentence. Our model gives the better results than CNN, LSTM and SVM on Vietnamese datasets. In addition, we also collected 17,500 reviews from Vietnamese social media websites and annotated these reviews as positive/neutral/negative opinions. We expect this dataset to facilitate research on deep learning models for Vietnamese sentiment analysis.

As discussed in the error analysis section, we are going to improve the performance on the unambiguous cases as well as the cases containing both positive sentiment and negative sentiment. In addition, we are also collecting data for training a Vietnamese word vector model. It is expected to give a better representation for applying deep learning models.

REFERENCES

[1] Bo Pang and Lillian Lee, "Opinion mining and sentiment analysis. Foundations and trends in information retrieval", 2(1-2):1-135, 2008.

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

[3] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede, "Lexiconbased methods for sentiment analysis, Computational linguistics", 37(2):267–307, 2011.

[4] Sida Wang and Christopher D Manning, "Baselines and bigrams: Simple, good sentiment and topic classification", In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, pages 90–94. Association for Computational Linguistics, 2012.

[5] Jichang Zhao, Li Dong, Junjie Wu, and Ke Xu. "Moodlens: an emoticon-based sentiment analysis system for chinese tweets". In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1528–1531. ACM, 2012.

[6] Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. "Sentiment analysis of short informal texts". Journal of Artificial Intelligence Research, 50:723–762, 2014.

[7] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. "A neural probabilistic language model". journal of machine learning research, 3(Feb):1137–1155, 2003.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space". arXiv preprint arXiv:1301.3781, 2013.

[9] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. "Learning word vectors for sentiment analysis". In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA, June. Association for Computational Linguistics, 2011.

[10] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. "Learning word vectors for sentiment analysis". In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 142–150. Association for Computational Linguistics, 2011.

[11] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. "Semi-supervised recursive autoencoders for predicting sentiment distributions". In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 151–161. Association for Computational Linguistics, 2011.

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

[13] B.T. Kieu and S.B. Pham. "Sentiment Analysis for Vietnamese". In Proceedings of Second International Conference on Knowledge and Systems Engineering (KSE), pp. 152–157, 2010.

[14] Duyen Nguyen Thi, Ngo Xuan Bach, and Tu Minh Phuong. "An empirical study on sentiment analysis for Vietnamese". Advanced Technologies for Communications (ATC), International Conference on. IEEE, 2014.

[15] Ainur Yessenalina and Claire Cardie. "Compositional matrix-space models for sentiment analysis". In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 172–182. Association for Computational Linguistics, 2011.

[16] Ozan Irsoy and Claire Cardie. "Deep recursive neural networks for compositionality in language". In Advances in Neural Information Processing Systems, pages 2096–2104, 2014.

[17] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. "Recursive deep models for semantic compositionality over a sentiment treebank". In Proceedings of the conference on empirical methods in natural language processing (EMNLP), volume 1631, page 1642. Citeseer, 2013.

[18] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. "A convolutional neural network for modelling sentences". arXiv preprint arXiv:1404.2188, 2014.

[19] Yoon Kim. "Convolutional neural networks for sentence classification". arXiv preprint arXiv:1408.5882, 2014.

[20] Ye Zhang and Byron Wallace. "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification". arXiv preprint arXiv:1510.03820, 2015.

[21] Quoc V Le and Tomas Mikolov. "Distributed representations of sentences and documents". In ICML, volume 14, pages 1188–1196, 2014.

[22] Duyu Tang, Bing Qin, and Ting Liu. "Document modeling with gated recurrent neural network for sentiment classification". In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1422–1432, 2015.

[23] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8: 1735-1780, 1997.

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

[25] Goller, Christoph, and Andreas Kuchler. "Learning task-dependent distributed representations by backpropagation through structure." Neural Networks, 1996., IEEE International Conference on. Vol. 1. IEEE, 1996.

[26] Boureau, Y-Lan, Jean Ponce, and Yann LeCun. "A theoretical analysis of feature pooling in visual recognition." Proceedings of the 27th international conference on machine learning (ICML-10). 2010.

[27] Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. "A convolutional neural network for modelling sentences." arXiv preprint arXiv:1404.2188, 2014.

[28] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882, 2014.

[29] Zhang, Ye, and Byron Wallace. "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification." arXiv preprint arXiv:1510.03820, 2015.

[30] Wang, Xin, et al. "Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory." ACL (1). 2015.

[31] Zhang, Rui, Honglak Lee, and Dragomir Radev. "Dependency sensitive convolutional neural networks for modeling sentences and documents." arXiv preprint arXiv:1611.02361, 2016.

[32] Wang, Jin, et al. "Dimensional sentiment analysis using a regional cnn-lstm model." The 54th Annual Meeting of the Association for Computational Linguistics. Vol. 225, 2016.

[33] Phan, Dang-Hung, and Tuan-Dung Cao. "Applying skip-gram word estimation and SVM-based classification for opinion mining Vietnamese food places text reviews." Proceedings of the Fifth Symposium on Information and Communication Technology. ACM, 2014.

[34] Liu, Pengfei, Shafiq R. Joty, and Helen M. Meng. "Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings." EMNLP. 2015.

[35] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980, 2014.

[36] Huyen, Nguyen Thi Minh, Azim Roussanaly, and Hô Tuong Vinh. "A hybrid approach to word segmentation of Vietnamese texts." International Conference on Language and Automata Theory and Applications. Springer Berlin Heidelberg, 2008.