Fine-Tuning BERT on Sentiment Analysis for Vietnamese

Nguyen Quoc Thai, Nguyen Thoai Linh
Department of Computer Science
University of Information Technology
VNU-HCM
Ho Chi Minh City,Viet Nam
{*16521089,16521716*}*@gm.uti.edu.vn*

Quoc Hung Ngo, Hoang Ngoc Luong
University of Information Technology
Vietnam National University - HCM City
Ho Chi Minh City, Viet Nam
{*hungnq,hoangln*}*@uit.edu.vn*

***Abstract*—Sentiment analysis is an important problem in the field of Nature Language Processing (NLP). It serves to analyze and evaluates human feedback on a specific issue. Many deep learning models have been proposed to tackle this problem, including BERT. In this paper, we will introduce two BERT fine-tuning methods for the sentiment analysis problem for Vietnamese comments, a method proposed by the BERT authors using only the [CLS] token as the inputs for an attached feed-forward neural network, a method we have proposed, in which all output vectors are used as inputs for other classification models. Experimental results on two datasets show that models using BERT is outperforming than other models. In particular, in both results, our method always produces a model with better performance than the BERT-base method.**

Keywords – sentiment analysis, language model pre-train, BERT, deep learning.

1. Introduction

Nowadays, millions of comments on social networks, e-commerce websites are shared by users. This kind of data is valuable reference information for both customers and providers, however, it is difficult for humans to process such large amounts of information. Therefore, we need to build an automated evaluation system to show how customer feel. The system can predict emotions of content on a scale of 1 to 5 or can indicate what content is complimenting or criticizing. However, this article only make ”positive” and ”negative” prediction in the comments. It is the sentiment analysis task, which is a natural language processing (NLP) fundamental task. In 2018, Jacob Devlin et al [5] introduce a new language representation model, namely BERT (Bidirectional Encoder Representations from Transformers). This model has successfully improved recent works in finding representations of words in a digital space through its context.

In this paper, we will present 2 methods of using BERT for sentiment analysis Vietnamese comments. Firstly, we implement the classification method in [5]. Secondly, we propose a new method for classification that combine BERT into three classification model. Our goal is prove the effectiveness of BERT when using sentiment analysis compared to other models. Then we want to test another Fine-tuning method for BERT with sentiment analysis, along with selecting the right model to combine with BERT. Therefore, that results are the best.

This paper is organized as follow. Section 1 introduces of this paper. Section 2 present related work for sentiment analysis. Section 3 introduces Word Embedding, Language model and BERT. Section 4 introduces two method for fine-tuning BERT on sentiment analysis. Section 5 show experimental results are presented in Section 4. Finally, Section 6 concludes the paper.

2. Related Work

Sentiment analysis is one of text classification tasks, in which the input text is classified as sentiment labels, such as two label *“positive”* or *“negative”*. Therefore, previous research took advantage of machine learning methods to solve this problem. In 2002, the first research to classify the reviews into group, positive and negative [3]. In this research, the author used supervised learning models such as, Support Vector Machine (SVM) [6] [7], Naïve Bayes [14] [1] for the classification task. Or another approach, they use a dictionary of emotional vocabulary, which indicates whether a word is positive or negative along with it level [9]. Deep learning allows for better learn word and context representations. Yoon Kim used Convolution Neural Network (CNN) [16] for the document classification task, the author process with each character-based. Mikolov et al [19] proposed an unsupervised learning algorithm neural network called “Paragraph vector”. This method is similar [21]. However, they not using bag-of-words of context words as input, the author uses an additional matrix of text with each column of matrix is a *“Paragraph vector”*.

In Vietnamese text, Duyen et al Use Naïve Bayes, Max Entropy Model and SVM to classify reviews on agoda sites [11], which allow users to book hotel rooms on travel occasions, the results show that the SVM model achieved the best result. In deep learning approach, Quan et al [20] proposed a model that combine between Long Short-Term Memory

(LSTM) and CNN. It named multi-channel LSTM-CNN for Vietnamese sentiment analysis, which achieved a better performance than CNN, LSTM. This method similar [10], a deep learning model for managing negative comments on social networks, word vector is passed over the CNN network then the output is used as input for LSTM network to perform classification.

TABLE 1. APPROACHES FOR SENTIMENT ANALYSIS

Vietnamese	English
Naïve Bayes, MEM, SVM [11]	SVM [6] [7], Naïve Bayes [14] [1]
Multi-channel LSTM-CNN [20], CNN-LSTM [10]	TextCNN [16]
-	Unsupervised learning [19] [21]
-	Lexicon based [9]

In summary, deep learning is the most popular method in the field for sentiment analysis. However, the models are limited by Vietnamese resources for Natural Language Processing and Word2Vec, GloVE is fixed, not contextually flexible. So, we use BERT to solve above problems.

3. Background

3.1. Word Embedding

Word Embedding is a method of mapping each word into a multi-dimensional real space, however it is much smaller than the dictionary size, and then there was a lot of research to solve his problem. Tomas Mikolos proposed Word2Vec [21], which is a statistical method for efficiently learning a standalone word embedding from a text corpus. The global Vectors for Word Representation (GloVe) algorithm is an other method for efficiently learning word vectors, developed by Pennington et al. at Stanford [8]. Bidirectional Encoder Representation from Transformer (BERT) [5] use a method that learn contextually vector.

3.2. Language Model

The language model is a probability distribution over text sets. The language model can show how much probability a sentence (or phrase) belongs to a language. Language models analyze bodies of text data to provide a basis for their word predictions.

$$P(x_1 \ldots . x_n)$$

In which, x_1,x_2,$x_n \ldots$ is the sequence of words that make up a sentence, n is length of the sentence (n $>$ 1).

Currently, pre-train language models are widely used in NLP tasks. These models have been trained on a very large dataset, integrating many languages and knowledge. Therefore, users can use them in many different NLP tasks [5].

3.3. BERT

BERT is a multi-layered structure of Bidirectional Transformer encoder layers, based on the architecture of transformer [2]. BERT uses Bidirectional Transformer encoders replace Encoders combining Decoders. BERT fine-tuning tasks do not require Decoder blocks. Therefore, they replace Decoder blocks with similar Encoder blocks.

The most advantage of BERT is apply two-dimensional training techniques of Transformers from a very famous Attention model to a Language Model. This is different from previous NLP studies, which looking at a text string from left to right, BERT combines how to look at a text string from 2 dimensions (from left to right and right to left). This method can greatly improve the retention of Representations of words in sentences. The published results also show that the trained language model has a more profound contextual meaning than previous models.

The authors have proposed two sizes for the BERT model:

- $BERT_{BASE}$: 12 encoder block, 12 head attention, 110 million parameter.
- $BERT_{LARGE}$: 24 encoder block, 16 head attention, 340 million parameter.

4. Methodology

In the sentiment analysis for comment, BERT can be used with two methods:

- **Feature extraction:** Using BERT as a feature extraction model. The architecture of the BERT model is preserved by authors, and its outputs are featured input vectors for the subsequent classification models to solve the given problem.
- **Fine-tuning:** In this method, we will have to change a bit of the architecture of the model by adding some layers at the end of BERT model, these layers will solve the problem, and then we will retrain the model. This process is called fine-tuning. In [5], this method is also the main method used to evaluate benchmarks on different tasks, showing BERT superiority over previous models.

The BERT-specific feature extraction method is used by the authors in the entity identification task (CoNLL-2003 NER) [4] according to the results given that F1 measurement when performing the feature extraction is smaller than when performing fine-tune BERT.

TABLE 2. COMPARISON BETWEEN FINE-TUNING AND FEATURE-BASED BERT [5]

SYSTEM	Dev F1	Test F1
ELMO	95.7	92.2
CVT	-	92.6
CSE	-	**93.1**
Fine-tuning approach		
$BERT_{LARGE}$	96.6	92.8
$BERT_{BASE}$	96.4	92.4
Feature-based approach ($BERT_{BASE}$)		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

TABLE 2 shows when using the fine-tune methods it get better performance than using the specific extraction method, even though the combination of hidden vectors has been performed, the result is still not as good as fine-tune. The combination of hidden vectors also takes a lot of time and resources. Therefore, this is an unreliable method using fine-tuning. For the above reasons, we decided to use the fine-tune BERT method to solve the sentiment analysis problem.

To Fine-tuning BERT, we need to use BERT pre-train, it is required when using pre-train that it has been trained with Vietnamese dataset. This article will use the pre-train BERT-Base Multilingual Cased[1] provided by Google, the reason for using this pre-train is because it supports multiple languages, including Vietnamese along with other languages (104 languages). Cased means that it will be case-sensitive, the tone of words will also be kept suitable for use in Vietnamese. This paper will introduce two methods of Fine-tuning BERT for sentiment analysis.

- **Fine-tuning BERT using token [CLS]:** In [5], the authors performed fine-tuning of BERT at the sequence level. They will add a special token to perform the classification tasks. The token named [CLS] is added to the beginning of the sentence and will represent the entire sentence. The output vector of this token will be sent through the feed forward neutral network to perform input sentence classification. We named this model was *BERT-base*. Architectural model will be described through Figure 1.

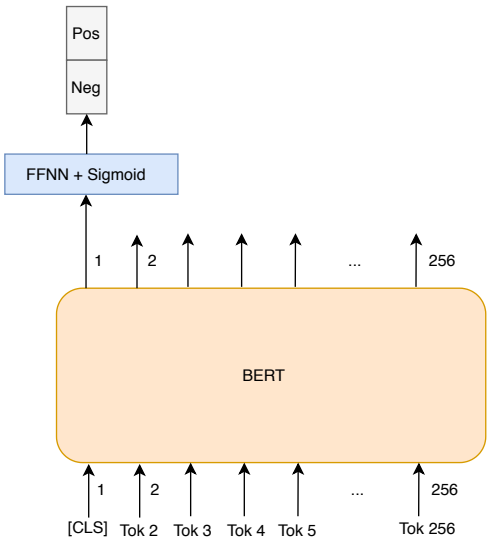

Figure 1. Architecture BERT-base

- **Fine-tuning BERT using all token:** We using the entire output of BERT including the token [CLS], these outputs will form a matrix of dimensions equal to SEQ_LEN x h. With SEQ_LEN is the maximum length of the input sequence, h is length of hidden vector. From there, we can use this output matrix as an input to other classification models. In this method, we will use three models to combine with BERT. A Recurrent model is LSTM, a convolution is TextCNN and a model that combines recurrent and convolution models is RCNN. Architectural model of this method will be described through Figure 2.

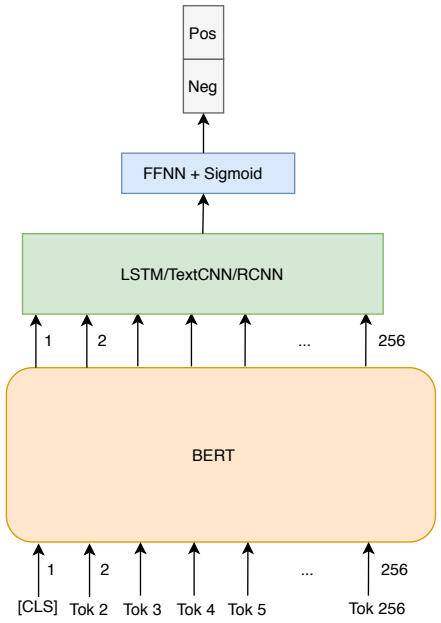

Figure 2. Architecture BERT using all token

Long Short-Term Memory (LSTM): The issue of distance dependence is the main reason why this research do not use the RNN model. Therefore, we will use another architecture that uses the idea of RNN as the Long Short-Term Memory (LSTM) model. LSTM is the most popular and most used recurrent model, we use a LSTM layer to extract the features received from BERT.

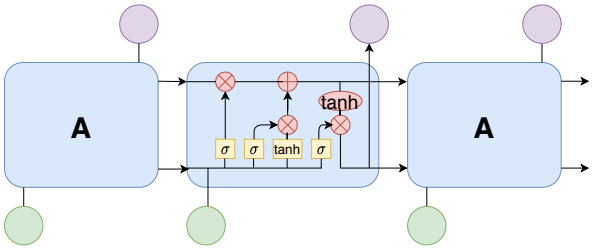

Figure 3. Architecture LSTM

Text Convolution Neural Network (TextCNN): The convolution model in this research is TextCNN [18]. This is the CNN model widely used in nature language processing problems, especially the elimination problems thanks to the quick training time and good results. The TextCNN model we use is a little different from the TextCNN model

1. https://github.com/google-research/bert

suggested in [18].Specifically, we will use 4 regions , 4 region as [2,3,4,5] and only use 1 filter for each region.

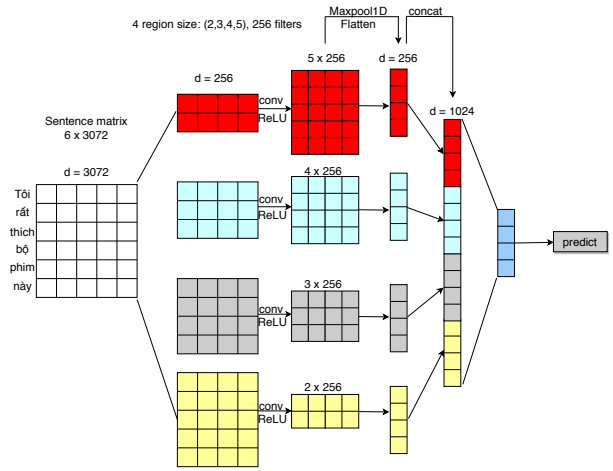

Figure 4. Architecture TextCNN [17]

Recurrent Convolution Network (RCNN): The model is a combination of a recurrent and a recurrent, we choose RCNN [13]. This model using 2 LSTM architectures combined, one layer LSTM will learn the context of words from left to right, the other will learn contextually from right to left, the output of both networks will be passed through a conv1d class to continue extracting. Architectural model will be described through Figure 5

Figure 3, Figure 4, Figure 5 describe in detail the architectural models that will be associated with BERT, and to perform classification of *Positive* and *Negative* labels, we will use logistics function that is $Sigmoid$.

Based on the results of comparison between Fine-Tune and Feature-based methods in TABLE 2, it is easy to see that Fine-Tuning method give better performance. However looking closely, we see that concat last four Hidden output gives the best results with Feature-based method. From here, We decided to contact last four hidden outputs when performing Fine-Tuning. Each hidden output will have $dim = h$. Therefore, when concat four hidden output dimensions of the output vector will be $dim = 4 \times h$.

5. Experiments

5.1. Dataset

Vietnamese sentiment datasets often have small numbers, most serve aspect-based sentiment analysis. Therefore, in this paper we will build two sets of data to serve because training process and evaluation of models.

We will use 2 sets of review commenting data on e-commerce sites. First, the *ntc-sv*[2] datase includes comments on restaurants and foodies on Foody. This dataset consists of 50,000 samples. The labels explained are based on the average score (avg_score), above 8.5 is labeled *positive*, less than 5 is labeled *negative*.

For the second dataset, we will use the training dataset of a competition on *AIVIVN*[3] on sentiment analysis, including comments on product reviews on the e-commerce site. In addition, we have collected some comments about the restaurants on *Foody*[4], then implemented the data label through the average score (avg_score) similar to the ntc-sv dataset. However, it has little change, if above 7.5 will be labeled *positive*, below 5 will be labeled *negative*. We will call this dataset is *vreview* for easy comparison.

The reason we use two datasets because we want to evaluate the overall models in both cases, along with assessing the advantages and disadvantages of using BERT in different data cases. We have made some data statistics for the evaluation process that show in TABLE 3 and TABLE 4. This is the statistics after data has been processed.

TABLE 3. DATA DESCRIPTION AFTER PRE-PROCESS

Dataset	Train		Test		Totally
	Positive	Negative	Positive	Negative	
ntc-sv	20,493	20,267	5,000	5,000	50,760
vreview	22,979	19,537	8,301	6,795	57,612

TABLE 4. STATISTICS OF WORDS INCLUDED IN THE COMMENTS

	vreview	ntc-sv
Mean	55.45	86.57
Stdn	63.75	77.41
Min	1	1
25%	14	37
50%	32	65
75%	76	111
Max	435	1,501

5.2. Comparison of method

In this section, we will compare several models, which can be used for sentiment analysis for Vietnamese:

- **SVM / Boosting:** There are two basic machine learning algorithms, used much before deep learning algorithms prevailed, with SVM we use features from n-grams, n in the range [1,5]. Our Boosting algorithm will use XGBoosting [15] with a deep of 15.
- **FastText + LSTM / TextCNN / RCNN:** We choose three algorithms to combine with word embedding model FastText is similar to 3 models associated with BERT, this will help us more accurately evaluate the results that the model of we archive. The pretrained FastText [5] used has been trained on Vietnamese dataset.

2. https://streetcodevn.com/blog/sav

3. https://www.aivivn.com/contests/1

4. https://forum.machinelearningcoban.com/t/du-lieu-review-cua-foody/203

5. https://fasttext.cc/docs/en/crawl-vectors.html

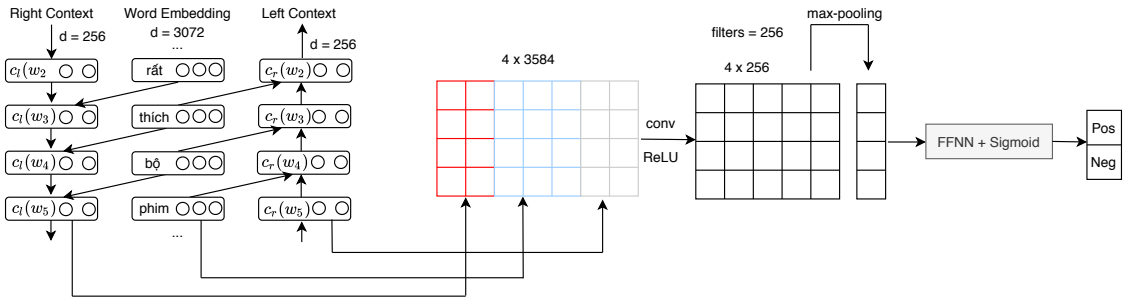

Figure 5. Architecture RCNN

- **GloVE + LSTM / TextCNN / RCNN:** We will use glove-python[6] library to train word embedding. The reason for this is because we don't have GloVE pre-trains with the same dimensions as the FastText pre-train. Therefore, it will ensure the fairness for all. Models associated with GloVE will be similar to models combine with FastText.

5.3. Results and discussion

The results of comparing the models with the two datasets are shown in the table below. Measure is F1-score.

TABLE 5. RESULT OF OUR MODEL ON NTC-SV DATASET COMPARED TO OTHER MODELS

Model	Precision(%)	Recall(%)	F1(%)
SVM	89.23	92.52	90.84
XGBoost	88.76	90.58	89.63
FastText + TextCNN	67.9	89.1	77.1
FastText + LSTM	88.5	89.7	89.1
FastText + RCNN	89.2	91.7	90.4
Glove + TextCNN	69.7	87.7	77.7
Glove + LSTM	88.7	91.8	89.8
Glove + RCNN	85.8	85.8	90.7
BERT-base	88.13	**94.02**	90.9
BERT-LSTM	**89.78**	92.08	90.91
BERT-TextCNN	88.85	93.14	90.94
BERT-RCNN	88.76	93.68	**91.15**

TABLE 6. RESULT OF OUR MODEL ON VREVIEW DATASET COMPARED TO OTHER MODELS

Model	Precision(%)	Recall(%)	F1(%)
SVM	86.26	86.9	86.5
XGBoost	87.69	88.45	88.07
FastText + TextCNN	61.8	**94**	74.6
FastText + LSTM	88.5	86.4	87.5
FastText + RCNN	84.5	89.8	87.1
Glove + TextCNN	62.6	93	74.8
Glove + LSTM	85.8	85.8	85.8
Glove + RCNN	84.0	88.6	86.2
BERT-base	86.08	88.44	87.2
BERT-LSTM	85.25	89.9	87.5
BERT-TextCNN	**90.9**	85.2	87.98
BERT-RCNN	87.08	89.38	**88.22**

6. https://github.com/maciejkula/glove-python

Based on the results obtained from TABLE 5, TABLE 6 we have some conclusions:

- When comparing the models on the ntc-sv dataset, we see that the results are not much different, even when comparing our models and svm, it can be said this is a comparison between giants. and tiny people. Therefore, deep learning models is not feasible in this case. Deep learning models train much longer. However, it give no much better performance than SVM.
- Models using Word Embedding FastText and Glove produced similar results. First, TextCNN models produce very poor results, much lower than other models. Secondly, the model using architecture of LSTM and RCNN, especially models using RCNN architecture produce very good results, because it combines the features that are received from many sources, from there is more information to do in classifying comments, in return that the training time will be significantly longer.
- BERT-LSTM and BERT-TextCNN have quite good and similar results, like BERT-base in both datasets, showing ineffectiveness in combining with LSTM and TextCNN architectures.
- BERT-RCNN is the model that gives the best results in the methods we use, showing the suitability of the RCNN model to not only BERT but also in other Embedding methods. Therefore, This is the most suitable model to be combined with BERT to implement Fine-tuning in the second way with the very good results that it brings.

Initially, we had high expectations for BERT. However, the empirical results did not see a significant difference between BERT and other models, although we used a different model that made the most of the information. Information that BERT stores. This result of models can explain by the following reasons:

- The data has not really made it difficult for the models, the length of the comments is relatively short, the preprocess eliminates most of the possible interference, making it easier to categorize the comments.

- Fine-tuning BERT, both BERT-base and when combining BERT with other algorithms, uses a lot of specific resources, namely RAM and GPU. The input chain of BERT after sub-splitting is 512, because our resources are not enough to be able to train BERT with the input string length of 512, if we have enough resources to train training with maximum chain length can achieve significantly better performance.
- Pre-train BERT-Base Multilingual Cased is not really optimal. The sub-word separator in BERT uses the sub-word separation technique for English. when applied to Vietnamese leads to inaccurate separation, the BERT pre-train training for Vietnamese has a contextual deviation in words.

6. Conclusion

We have fine-tuned BERT using the pre-trained multilingual BERT with two approaches and both exhibit superior performance compared to other models considered in this research. With the results , we found that the combination of BERT, LSTM and TextCNN yielded no more improvement than Fine-Tuning BERT. However, this combination also made the training time significantly longer. Therefore, we do not recommend the use of LSTM or TextCNN in combination with BERT. You should combine BERT with models using RCNN or other models that Combination of Recurrent and Convolution models. And, when performing fine-tuned BERT, you can use the entire output of BERT, which helps the rear classification model to archive more information.

The experimental results show that the improvement of the accuracy of the BERT-RCNN model using the proposed group method, the improvement is not too significant. However, it has been showed the potential of this method.

References

[1] Lopamudra Dey Sanjay Chakraborty et al. , "Sentiment Analysis of Review Datasets Using Naïve Bayes and K-NN Classifier", in International Journal of Information Engineering and Electronic Business , 2016.

[2] Ashish Vaswani et al. "Attention is all you need", 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

[3] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan, "Sentiment classification using machine learning techniques", in Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2002), 2002.

[4] Erik F Tjong Kim Sang and Fien De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition". In CoNLL (2003).

[5] Jacob Devlin, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". 2018.

[6] Bhumika M. Jadav and Vimalkumar B. Vaghela, "Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis", in International Journal of Computer Applications 2016.

[7] Nurulhuda Zainuddin and Ali Selamat, "Sentiment analysis using Support Vector Machine", 2014 International Conference on Computer, Communications, and Control Technology (I4CT).

[8] Jeffrey Pennington, Richard Socher, Christopher Manning, "Glove: Global Vectors for Word Representation", in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

[9] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede, "Lexicon-Based Methods for Sentiment Analysis", Computational Linguistics, vol. 37, no. 2, pp. 267-307, 2011.

[10] Khuong Vo, Tri Nguyen, Dang Pham, Mao Nguyen, Minh Truong, Dinh Nguyen, Tho Quan, "Handling negative mentions on social media channels using deep learning", Journal of Information and Telecommunication, vol. 3, no. 3, 2019.

[11] Nguyen Thi Duyen, Ngo Xuan Bach, Tu Minh Phuong, "An Empirical Study on Sentiment Analysis for Vietnamese". The 2014 International Conference on Advanced Technologies for Communications (ATC'14).

[12] Peter D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews", in Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL- 2002), 2002.

[13] Siuwei Lai, Liheng Xu, Kang Liu, Jun Zhao, "Recurrent Convolutional Neural Networks for Text Classification", Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (2015).

[14] Kavya Suppala, Narasinga Rao, "Sentiment Analysis Using Naïve Bayes Classifier", in International Journal of Innovative Technology and Exploring Engineering (IJITEE) 2019

[15] Tianqi Chen, Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System", The 22nd ACM SIGKDD International Conference (2016).

[16] Xiang Zhang, Junbo Zhao, Yann LeCun, "Character-level Convolutional Networks for Text Classification", arXiv:1509.01626, 2015.

[17] Ye Zhang, Byron C. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification", IJCNLP 2017.

[18] Yoon Kim "Convolution Neutral Network for Sententce Classification", Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[19] Quoc Le, Tomas Mikolov, "Distributed Representations of Sentences and Documents", in Proceedings of the International Conference on Machine Learning (ICML 2014), 2014.

[20] Quan-Hoang Vo, Huy-Tien Nguyen, Bac Le, Minh-Le Nguyen, "Multi-channel LSTM-CNN model for Vietnamese sentiment analysis", 2017 9th International Conference on Knowledge and Systems Engineering (KSE).

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean , "Distributed Representations of Words and Phrases and their Compositionality", in Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013), 2013.