Journal of Computer Science and Cybernetics, V.36, N.4 (2020), 305–323
DOI 10.15625/1813-9663/36/4/14829

A TWO-CHANNEL MODEL FOR REPRESENTATION LEARNING IN VIETNAMESE SENTIMENT CLASSIFICATION PROBLEM

QUAN NGUYEN HOANG, LY VU, QUANG UY NGUYEN*

Faculty of Information Technology, Le Quy Don Technical University

Abstract. Sentiment classification (SC) aims to determine whether a document conveys a positive or negative opinion. Due to the rapid development of the digital world, SC has become an important research topic that affects to many aspects of our life. In SC based on machine learning, the representation of the document strongly influences on its accuracy. Word embedding (WE)-based techniques, i.e., Word2vec techniques, are proved to be beneficial techniques to the SC problem. However, Word2vec is often not enough to represent the semantic of Vietnamese documents due to the complexity of semantics and syntactic structure. In this paper, we propose a new representation learning model called a two-channel vector to learn a higher-level feature of a document for SC. Our model uses two neural networks to learn both the semantic feature and the syntactic feature. The semantic feature is learnt using Word2vec and the syntactic feature is learnt through Parts of Speech tag (POS). Two features are then combined and input to a Softmax function to make the final classification. We carry out intensive experiments on 4 recent Vietnamese sentiment datasets to evaluate the performance of the proposed architecture. The experimental results demonstrate that the proposed model can enhance the accuracy of SC problems compared to two single models and three state-of-the-art ensemble methods.

Keywords. Sentiment analysis; Deep learning; Word to vector (Word2vec); Parts of speech (POS); Representation learning.

1. INTRODUCTION

Sentiment classification (SC) is a task of determining the psychological, emotional and opinion tendencies of users through comments and reviews in a document. Due to the great explosion of data from the Internet, SC has become an emerging task in many online applications. People's opinions have a certain influence on the choice of a product, the improvement of services, the decision to support individuals and organizations, or agreement with a policy. The emotional polarity of positive and negative reviews helps a user decide on whether or not to buy a product. Thus, SC of user reviews has become an important research topic in text mining and information retrieval of data from the Internet.

The main goal of SC is to classify user reviews in a document into opinion poles, such as positive, negative, and possibly neutral sentiments. There are two popular approaches for SC: The lexicon-based approach and the machine learning-based approach. The lexicon-based approach is usually based on a dictionary of negative and positive sentiment values assigned to words. This method thus depends on human effort to define a list of sentiment words and sometimes it suffers from low coverage. Recently, machine learning methods

*Corresponding author.
E-mail addresses: nghoangquan@gmail.com (Q.N.Hoang); vuthily.tin2 (L.Vu); quanguyhn@gmail.com (Q.U.Nguyen).

have been widely applied to SC and they often achieve higher accuracy than lexicon-based approaches in some recent researches [6, 14, 19]. These techniques often use Bag-of-Words (BOW) or Term Frequency-Inverse Document Frequency (TF-IDF) features to describe the characteristics of documents. However, these features can not represent the semantics of documents and sometimes they are ineffective for SC.

In recent years, deep learning has played an important role in natural language processing (NLP) [2, 25, 32, 33, 35, 36]. The advantage of deep neural networks is that they allow automatic extraction of features from documents. Mikolov [11] proposed a Word Embedding (WE) model, namely Word2vec, using a neural network with one hidden layer to learn the word representation. Word2vec can represent the semantic relation of words that are placed closely in a sentence. This representation is then widely used in SC [3, 5, 9]. However, the vector calculated by Word2vec does not consider the context of the document [20]. Another shortcoming is that Word2vec could present the opposite meaning words closely together in the feature space [26] and resulting in the difficulty for machine learning algorithms in SC.

To handle the limitation of Word2vec, Rezaeinia et. al. [20] proposed an Improved Word Vector (IWV) that combines the vectors of Word2vec, parts of speech (POS) and sentiment words for English documents. The IWV is then inputted to a Convolutional Neural Network (CNN) to learn the higher level of features. The results show that IWV can increase the accuracy of the SC problem compared to using only Word2vec. However, the combination method in [20] has some limitations when applying to Vietnamese language.

First, the resource of the sentiment words in Vietnamese may not be enough to generate an effective sentiment word vector for documents [1]. Second, IWV is formed by concatenating the Word2vec vector and the one-hot POS vector thus this vector can not be updated during the training process [2].

In this paper, we propose a deep learning-based model for learning representation in SC called Two-Channel Vector (2CV). In 2CV, one neural network is used for learning the representation based on Word2vec and another network is used for learning the representation from POS. The outputs of two neural networks are combined to form 2CV and this vector is input to a Softmax layer to make the final classification. 2CV has the ability to represent the semantic relationship of words by the Word2vec feature and the syntactic relationship using the POS feature. The combination of the semantic and syntactic features helps 2CV improve the performance of SC. The contributions of this paper are as follows:

- We propose a novel deep learning model for learning the representation in SC for Vietnamese language in which two networks are used to learn Word2vec and POS features, respectively. These features are then concatenated to form the final feature, i.e., 2CV, and 2CV is inputted to a Softmax function to produce the final classification.

- We apply this model to four datasets of Vietnamese SC. The experiment results show that our model has superior performance compared to two methods using a single feature and three recently proposed models that also used a combination of multiple features.

[1] In fact, we could only find one resource of sentiment words in Vietnamese [31] compared to six resources [20] in English.

[2] It is often not relevant to retrain a vector in one-hot representation.

The rest of the paper is organized as follows: Section 2 highlights recent research on the SC problem. In section 3 we briefly describe the fundamental of CNN and Long Short-Term Memory (LSTM). The proposed model is then presented in Section 4. This is followed by Section 5 and Section 6 presenting experimental results, the analyses and discussion of our proposed technique. Finally, in Section 7 we present some conclusions and suggest future work.

2. RELATED WORK

SC at the document level aims to determine whether the document conveys a positive, negative or neutral opinion [35]. When using machine learning for SC, the representation of the document is crucial that affects the accuracy of classification models. Traditionally, words in a document are represented using BOW or WE techniques. The BOW-based models represent a document as a fixed length numeric vector where each element of the vector presents the word occurrence or word frequency (TF-IDF score) [35]. The dimension of the feature vector is the length of the word vocabulary. Thus, the BOW feature vector is usually a sparse vector particularly for documents containing a small number of words.

Moraes et al. [13] compared two machine learning methods including Support Vector Machine (SVM) and Artificial Neural Network (ANN) for SC at the document level. Their experiment results showed that ANN usually achieved better results than SVM especially on the benchmark dataset of movie reviews. Glorot et al. [4] studied the transfer learning approach for SC. They proposed a method based on deep learning techniques, i.e., Stacked Denoising AutoEncoder, to learn a higher-level of the BOW feature in documents. The experiments showed that the proposed representation is highly beneficial SC. Zhai et al. [34] proposed a semi-supervised AutoEncoder to learn the features of documents. Johnson et al. [8] introduced a method that utilizes BOW in the convolutional layer of CNN. To preserve the sequential information of words, they also proposed the sequential CNN model for SC.

Overall, BOW is very popular for representing documents for SC. However, BOW also has some limitations. First, it ignores the word order, thereby two different documents could have the same representation if they have the same set of words. Second, the feature vector of a document is often very sparse and high dimensional. Third, BOW only encodes the presence and the frequency of words. It does not capture the semantics of words in the document.

To overcome the shortcomings of BOW, WE-based techniques, i.e., Word2vec, are used in SC. K. Yoon et al. [9] used word's vectors of Google pre-trained models to extract features of sentiment sentences. The features are used as the input to a CNN to classify the sentiment of sentences. Kai et al. [23] proposed a Tree-Structured LSTM to learn semantic representations for SC. Tang et al. [25] introduced a method based on a neural network to learn document representation where the sentence relationship is considered. The proposed method has two steps: First, the representation is learned by CNN or LSTM. Second, the semantics of sentences and their relationship in the document are encoded by Gated Recurrent Unit (GRU). The model is then used for classifying user's movie reviews [26]. Xu et al. [32] proposed an LSTM-based model to capture the semantic information in a long text. The main idea is to adapt the forgetting gates of LSTM to capture global and local semantic features. Zhou et al. [36] introduced an attention-based LSTM for cross-lingual SC. The

proposed model included two LSTMs to adapt the sentiment information from a rich resource language (English) to a poor resource language (Chinese).

Recently, multi-channel models have also been proposed to solve the SC problem. Vo et al. [29] proposed a parallel model of CNN and LSTM channels using the Word2vec feature. The objective is to use LSTM and CNN networks to exploit both local and global features. The output vectors are then concatenated and inputted to a Softmax function to predict the sentiment class of the input document. Shin et al. [21] proposed a model of two parallel CNN channels. One channel uses the Word2vec as the input and the other channel uses the sentiment word vector. The sentiment word vector is formed using 6 sentiment word resources in English.

In general, WE-based techniques have been proven to be an effective technique in SC. These approaches often produce higher accuracy compared to techniques based on BOW. In this paper, we further develop the WE-based method, i.e., Word2vec, to learn the representation of documents for SC. Specifically, we combine the Word2vec and POS features to create a new representation of documents. The new representation of the documents (2CV) thus can represent more useful information about the documents, thereby increasing the performance of SC.

3. BACKGROUND

This section briefly presents two deep learning networks (CNN and LTSM) used in SC and the technique to learn word representation in natural language processing. CNN is the most popular deep neural network and it is very effective for image analysis. In sentiment classification, each document can be represented as a matrix (similar to an image) in which the row is the number of words in the document and the column is the size of the word2vec vector. LSTM is a special form of RNN with the ability to remember long dependencies. Thank to this design, the LSTM network is the most popular structure applied to language processing problems including the sentiment classification problem.

3.1. Convolution neural network

Convolution neural network [10] is a class of deep neural networks that are often used in visual analysis. Recently, CNN is also widely used for the SC problem [9, 25]. To apply CNN for the SC problem, a document is represented as a matrix of size $s \times N$ where s is the number of words in the document and N is the dimension of each word vector x_i.

A convolution operation involves a filter $m \in \mathbb{R}^{kN}$ where k is the number of words used to produce a new feature. For example, a feature c_i is generated from a window of words $x_{i:i+k-1}$ as described in the following equation

$$c_i = f\left(m \times x_{i:i+k-1} + b\right), \tag{1}$$

where $b \in \mathbb{R}$ is a bias term and f is a non-linear activation function, such as the Sigmoid or Hyperbolic tangent (Tanh). The filter m is applied to each possible window of words in the document $\{x_{1:k}, x_{2:k+1}, \ldots, x_{s-k+1:s}\}$ to produce a new feature map c. At the last layer, these features are passed to a fully connected Softmax layer to predict the sentiment class of the input document.

3.2. Long short-term memory

Long short-term memory networks are a type of recurrent neural network capable of learning long-term dependence in sequence prediction problems like SC [1, 18, 24]. The key element in LSTMs is cells. An LSTM cell at step t comprises a cell state C_t and a hidden state h_t in Figure 1.

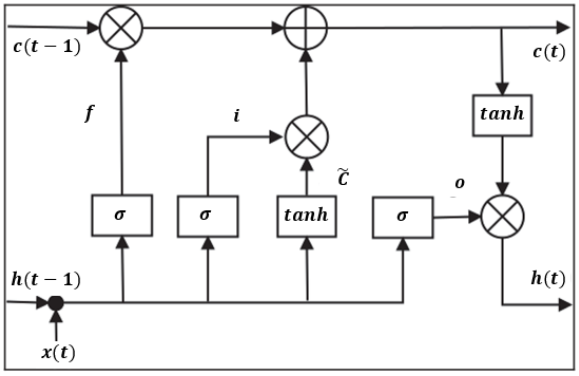


Figure 1. Architecture of an LSTM

To predict the sentiment class for a document d of N words, the words in d will be input to the cell in sequence order. At each step, the inputs to the cell are the current word x_i and the output of the previous word h_{t-1}. Another input is the state value of the previous step C_{t-1}. This value, i.e., C_{t-1} is to decide which information is forgotten and which information is forwarded to the next step. At the final step (the last word), the output h_f is inputted to a Softmax function to predict the sentiment class of the input document. More detailed description can be found in [7].

3.3. Word2vec

Word2vec is a method of representing words proposed by Mikolov [11]. It includes two architectures: Continuous Bag of Words (CBOW) and Skip-gram. CBOW predicts a target word based on context words while Skip-gram predicts context words from a target word. Since CBOW usually works better than the Skip-gram for the syntactic task [11], we will apply the CBOW architecture to extract the word vector in this paper.

Figure 2 presents the CBOW architecture to build a word vector using a fully connected neural network. In this figure, the goal is to project a spare input vector to a dense vector in the hidden layer h. For each input word, x_i, the context words or target words are t words before and t words after the word x_i in the document. Let V be the size of the vocabulary of words in the corpus [16], the input word, x_i, is represented by V-dimension one-hot vector. This vector has all values as 0 except the index of the x_i in the vocabulary where the value is 1. $W_{V\times N}$ is the weight matrix of the neural network from the input layer to the hidden layer h and $W'_{V\times N}$ is the weight matrix of the neural network from the hidden layer h to the output layer, where N is the size of the hidden layer. The output layer then inputs to the Softmax function to get the output label $\hat{y}_j$. The optimization process is used to reduce the difference between y_j and the expected output $\hat{y}_j$ by minimizing the Cross-Entropy loss

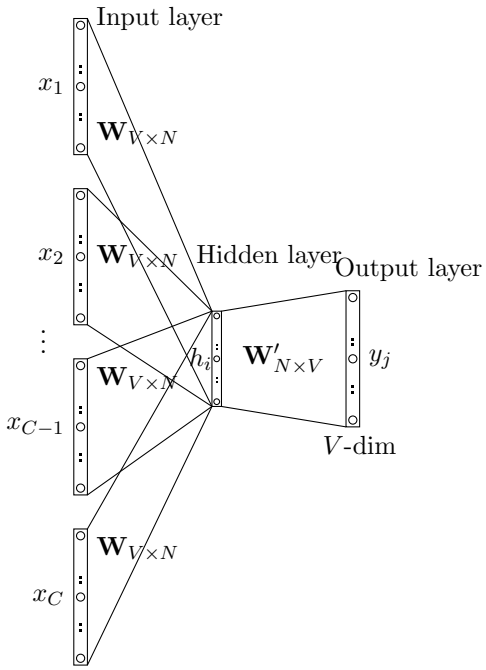


Figure 2. Architecture of Continue Bag of Words (CBOW)

function. After training, the representation of word vectors is the matrix W, the vector of the j^{th} word in the dictionary is the value in the j^{th} row of the matrix W.

4. PROPOSED METHOD

This section presents our proposed model for leveraging the accuracy of SC. First, we present the techniques to pre-process the documents. Second, the method to extract POS from sentences is described. Last, we present the proposed neural network model.

4.1. Pre-processing

The first step is to pre-process the input documents. Since Word2vec and POS features are based on the word level, it is necessary to pre-process raw documents to remove unexpected characters in the documents. The pre-processing process includes several tasks including removing special characters, replacing symbols by words that have corresponding descriptions, and tokenizing words. The removed characters consist of !"\#$%&'()*+,−./ : ;<=>?@[] ∧ ‘{|} ∼, except for _ to connect syllables in Vietnamese. The symbols replaced by words are presented in Table 1.

Moreover, since the POS feature is extracted at the sentence level, it is necessary to separate a document into sentences. As a result, we obtain two documents from the original document. The first document includes a set of words that are used to learn the Word2vec feature. The second document includes the POS of the words that are used to learn the POS feature. Finally, we build two vocabularies corresponding to these documents. The vocabu-

Table 1. Symbols and abbreviations are replaced by words

Icon	Text replace	Abbreviation	Text replace
👍	thích	ko	không
😍	yêu	thjk	thích
😀	vui_vẻ	mún	muốn
😖	giận_dữ	thank	cám_ơn
😒	buồn	iu	yêu
👎	ghét	dc	được

laries are used to define words in the document which are then inputted to the Word2vec network and the POS network in our model.

4.2. Extracting part-of-speech

In the Vietnamese language, words can be considered as the smallest elements that have a distinctive meaning. Based on their usage, words are categorized into several types of POS such as verbs, nouns, adjectives, adverbs. The POS feature helps to distinguish poly-semantic words in a sentence. Moreover, it also has a distinctive structure in each language. The combination of the POS feature and a uni-gram word is able to keep the meaning of the original word.

To extract POS from documents, we first tokenize each document into sentences. After that, we get POS tagging in each sentence by using the VnCoreNLP tool from Vu et al [30]. The POS of each word in the document is represented as a one-hot vector with the size of d [3]. A matrix with the size of $s \times d$ (s is the number of words in the document) is the POS representation of the document.

4.3. Architecture of the proposed model

Our proposed model (2CV) includes two channels where each channel is a neural network. The neural networks are used to learn the higher-level features of documents. The first channel learns a higher-level feature from the Word2vec feature and the second channel learns a higher-level feature from the POS feature. As a result, the proposed model can learn the higher-level representation of a document that captures both semantic property and the syntactic structure of documents.

Figure 3 describes 2CV in detail. Two types of features, i.e., Word2vec and POS, are extracted from the input documents. Each feature is then passed to a neural network channel. In this paper, we use two popular deep network models including LSTM or CNN (Figure 5) to learn features from Word2vec and POS due to their effectiveness for SC [25, 32, 35]. The outputs from two channels are concatenated to form the presentation of the documents. This representation is then inputted to the Softmax function to make the final classification.

Figure 4 presents the structure of 2CV that uses LSTM-based architecture and Figure 5 is the structure of 2CV that uses CNN architecture. In Figure 4, P, V, and N are shorted for Pronoun, Verb, and Noun, respectively. They are the POS of the words in the input

[3] d equals the number of POS in the Vietnamese language.

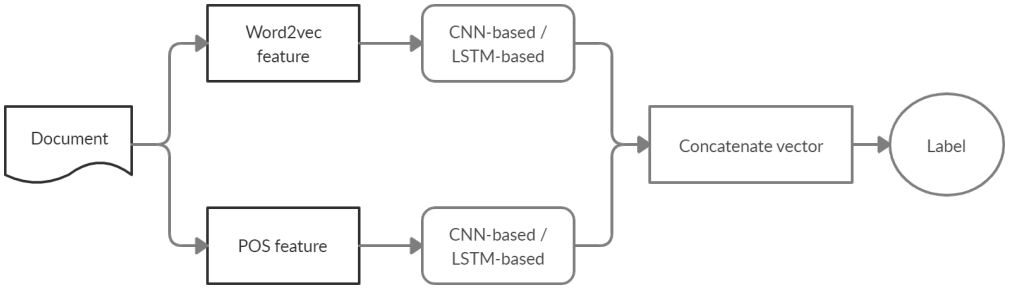


Figure 3. Model using two channels

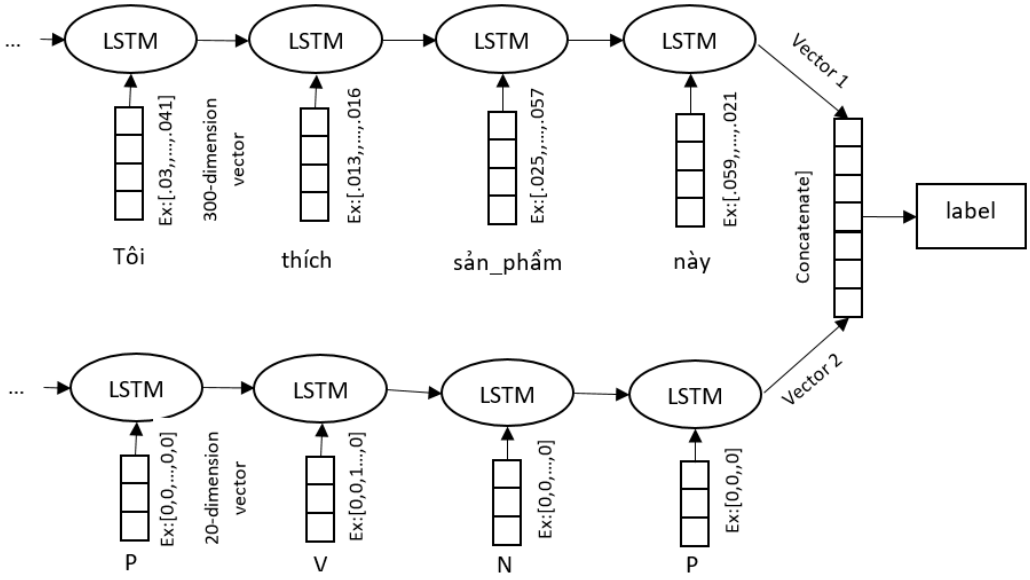


Figure 4. Model using two LSTM channels

sentence. In the LSTM-based model, each word is represented by a 300 dimension vector and its POS is presented by a 20 dimension vector. In Figure 5, each word is also represented by a 300 dimension vector and the document is padded to the length of 100 words before inputting to the network.

Our model differs from the model in [20] by using two channels to learn two sets of features separately. In the first channel, Word2vec is pre-trained from the Vietnamese corpus of documents using the CBOW model and then fine-tuned by LSTM or CNN. In the second channel, the POS feature represented as a one-hot encoding vector is learned by another LSTM or CNN. More precisely, two neural networks are used to learn two features separately and then the outputs of two networks are concatenated to form the new feature. Conversely, Rezae el at. [20] combined all features before inputting them to deep neural networks for training.

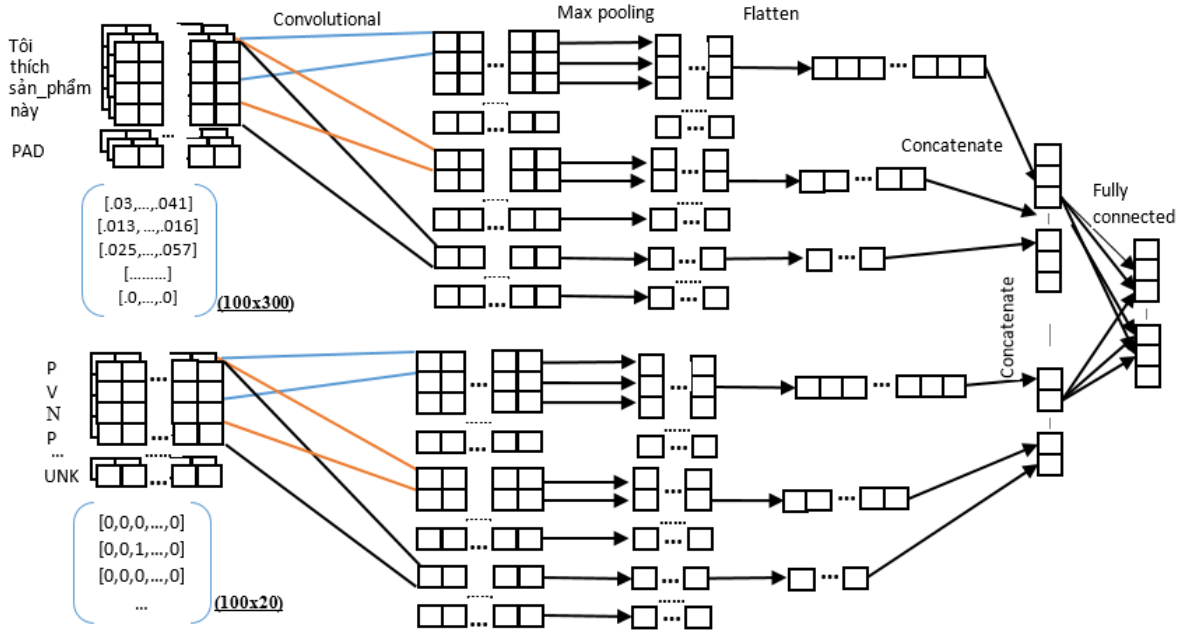


Figure 5. Model using two CNN channels

5. EXPERIMENTAL SETTTINGS

This section describes the datasets, the parameter's settings and the performance metrics used in the paper.

5.1. Datasets

To evaluate the accuracy of the proposed model, we tested it on four Vietnamese sentiment datasets. The number of total samples and the samples in each class are shown in Table 2. In this table, Pos, Neg, and Neu are the numbers of positive, negative and neutral samples.

- VLSP dataset: This is a Vietnamese sentiment dataset of electronic product reviews provided by Vietnamese Language and Speech Processing (VLSP) [17].
- AiVN dataset: This is the Vietnamese sentiment dataset used in the opinion classification contest organized by AI4VN [4].
- Foody dataset: Vietnamese sentiment dataset for comments on food and service[5].
- VSFC dataset: Vietnamese sentiment dataset about student feedbacks [28].

[4]https://www.aivivn.com/contests/1.
[5]https://streetcodevn.com/blog/dataset.

Table 2. Description of Vietnamese sentiment datasets

	VLSP			**VSFC**			**Foody**		**AiVN**	
Class	Pos	Neu	Neg	Pos	Neu	Neg	Pos	Neg	Pos	Neg
Train	1700	1700	1700	5643	458	5325	15000	15000	6489	4771
Test	350	350	350	1590	167	1409	5000	5000	2791	2036
Total	2050	2050	2050	7233	625	6734	20000	20000	9280	6807

5.2. Parameter's setting

In the experiments, we use two neural networks to learn the features form Word2vec and POS. The first network is LSTM and the second network is CNN. The dimension of the Word2vec feature is 300. The POS feature vectors have a dimension of 20 corresponding to 20 POS taggers in Table 3. In the LSTM-based model, the length of the input document is

Table 3. List of POS taggers in Vietnamese language

Acronym	POS tagger	Acronym	POS tagger
A	Adjective	C	Coordinating Conjunction
CH	Punctuation Mark	E	Preposition
FW	Foreign Word	I	Interjection
L	Determine	M	Numeral
N	Noun	Nb	Borrow Noun
Nc	Category Noun	Nu	Noun Unit
Ny	Acronym Noun	P	Pronoun
R	Adverb	T	Particle
V	Verb	X	Not Categorized
Z	Word constituent elements	UNK	Unknow

normalized to the average of document length in the training dataset. This model uses one LSTM layer with 64 hidden units and the *Tanh* function.

In the CNN-based model, we perform 1-dimension convolution with 50 output filters. The filter sizes are set at $2, 3, 4$ corresponding to *n-grams* features for text data [9]. In the pooling step, we use the max method to extract important features as in [9]. The output of the Max-Pooling layer is flattened into a vector. This vector is then inputted to a fully connected layer of 64 hidden units and the *Tanh* activation function.

5.3. Evaluation metrics

We use three metrics, namely, accuracy (ACC), F-score (F1), and Area Under the Curve (AUC) score [22] to compare the tested methods. These metrics are calculated based on the four following definitions.

- True Positive (TP): A TP is an outcome where the model correctly predicts the positive class.
- True Negative (TN): A TN is an outcome where the model correctly predicts the negative class.
- False Positive (FP): An FP is an outcome where the model incorrectly predicts the positive class.

- False Negative (FN): An FN is an outcome where the model incorrectly predicts the negative class.

ACC is the most common criterion to compare classification algorithms. Formally, the ACC of a classifier method applied on a dataset is calculated as in Equation (2)

$$\text{ACC} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}}. \tag{2}$$

F1 score is the harmonic mean of Precision calculated by $\frac{\text{TP}}{\text{TP+FP}}$ and Recall calculated by $\frac{\text{TP}}{\text{TP+FN}}$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}. \tag{3}$$

AUC is the area under the Receiver Operator Characteristic (ROC) curve. The ROC curve is plotted with the True Positive Rate (TPR) against the False Positive Rate FPR where TPR is on the y-axis and FPR is on the x-axis. Here, TPR known as sensitivity measures the proportion of positive cases in the data that are correctly identified (Equation 4). FPR known as (specificity) is the proportion of negative cases incorrectly identified as positive cases in the data (Equation 5)

$$\text{TPR} = \frac{\text{TP}}{\text{TP + FN}}, \tag{4}$$

$$\text{FPR} = \frac{\text{FP}}{\text{TN + FP}}. \tag{5}$$

6. RESULTS AND DISCUSSION

This section first presents the performance comparison between our model and the five other tested models. After that, we show the analysis of our model and the model concatenating the Word2vec and the POS vector before inputting to two channels [20] (IWV) in three aspects: Document visualization, computational time and error analysis.

6.1. Performance comparison

Table 4 and Table 5 present the performance of our proposed model (2CV) with the LSTM-based network and the CNN-based network, respectively. We compare the results of our model with the five other models. The first model is the method using the pre-trained Word2vec model (Pre-train) in which the pre-trained Word2vec vectors are input to two channels of the model and these vectors are not updated during the training process. The second model is the method using a single neural network (CNN or LSTM) to learn features from a pre-trained Word2vec vector (Single_W2V). In other words, the pre-trained Word2vec vector is input to CNN or LSTM and this vector is updated during the training process. The third model is the previous research [20] (IWV) in which the Word2vec vector and the POS

vector are combined before inputting to the neural network for learning a higher-level feature of documents. The fourth model is W2V_SentiWord in which the Word2vec vector is input to one channel and the sentiment word vector [21] is input to the other channel. The last model is W2V_BERT in which two inputs to the two channels are the Word2vec vector and the BERT feature vector [15], respectively.

Table 4. Results of the LSTM-based models

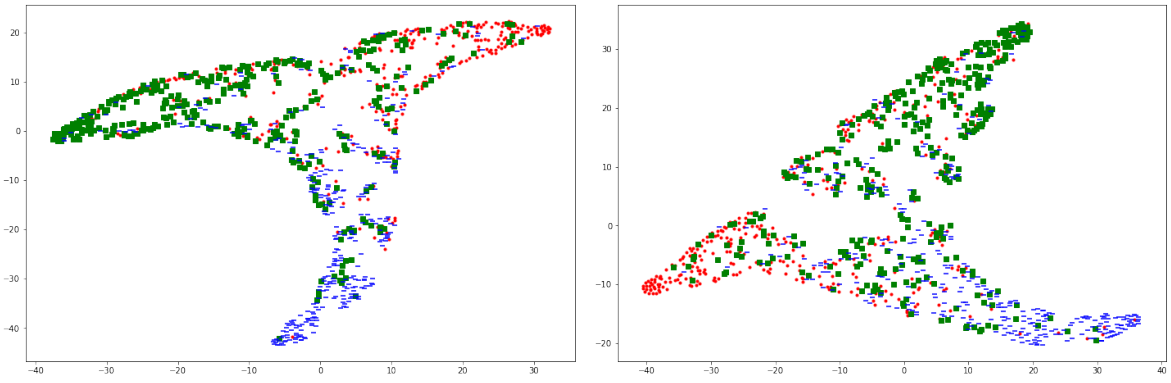
Metrics	Methods	Dataset			
		VLSP	VSFC	Foody	AiVN
ACC	Pre-trained	0.548	0.824	0.819	0.828
	Single_W2V	0.667	0.891	0.873	0.886
	IWV	0.609	0.889	0.882	0.842
	W2V_SentiWord	0.635	**0.899**	0.868	0.892
	W2V_BERT	0.644	0.898	0.873	**0.895**
	2CV	**0.695**	**0.899**	**0.886**	**0.895**
F1	Pre-trained	0.546	0.802	0.819	0.829
	Single_W2V	0.668	0.881	0.873	0.886
	IWV	0.606	0.885	0.882	0.843
	W2V_SentiWord	0.635	**0.893**	0.868	0.892
	BERT-W2V	0.643	0.891	0.873	**0.895**
	2CV	**0.695**	0.890	**0.885**	0.893
AUC	Pre-trained	0.742	0.878	0.900	0.911
	Single_W2V	0.830	0.936	0.944	0.951
	IWV	0.800	0.922	0.945	0.924
	W2V_SentiWord	0.805	0.941	0.936	**0.955**
	BERT_W2V	0.808	**0.942**	0.944	0.954
	2CV	**0.834**	0.939	**0.951**	0.953

Table 5. Result of the CNN-based models

Metrics	Methods	Dataset			
		VLSP	VSFC	Foody	AiVN
ACC	Pre-train	0.522	0.848	0.829	0.863
	Single_W2V	0.619	0.876	0.870	0.888
	IWV	0.544	0.889	0.830	0.870
	W2V_SentiWord	**0.641**	0.883	0.876	0.894
	BERT_W2V	0.611	0.893	**0.880**	0.891
	2CV	0.640	**0.896**	0.874	**0.895**
F1	Pre-train	0.521	0.836	0.829	0.863
	Single_W2V	0.618	0.865	0.870	0.889
	IWV	0.544	0.884	0.830	0.868
	W2V_SentiWord	0.639	0.874	0.876	0.894
	BERT_W2V	0.606	0.879	**0.880**	0.893
	2CV	**0.640**	**0.889**	0.874	0.896
AUC	Pre-train	0.709	0.907	0.906	0.936
	Single_W2V	0.806	0.930	0.940	0.952
	IWV	0.732	0.925	0.909	0.930
	W2V_SentiWord	0.804	0.935	0.944	0.954
	BERT_W2V	0.802	**0.941**	0.948	**0.955**
	2CV	**0.816**	0.936	**0.949**	**0.955**

First, these tables show that the accuracy of 2CV is remarkably better than IWV for both the LSTM-based and CNN-based models on all tested datasets. For example, the ACC scores of 2CV are improved from 0.609 to 0.695 on VLSP compared to the IWV using the LSTM-based model (see Table 4). The F1 score is also improved from 0.606 to 0.695 and the AUC is enhanced from 0.800 to 0.834, respectively, on this dataset [6]. The results on the other datasets are also considerably improved using all three performance metrics. These results show that our proposed models (2CV) are more effective than the IWV model to learn the representation of documents.

Second, the tables also show that 2CV is much better than the Pre-train method. This result is reasonable since the Pre-train representation on a general corpus will not capture enough specific information for the SC problems. For the Single_W2V method, we can see that this technique is significantly improved from the Pre-train method. However, the Single_W2V method is still inferior compared to 2CV. This result shows that combining the POS feature with the Word2vec feature in 2CV helps to improve the accuracy of machine learning on the SC problems.



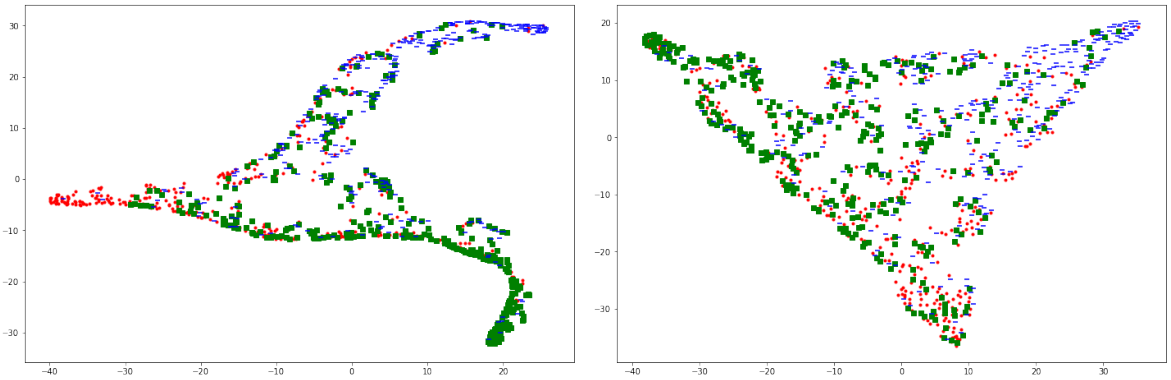
(a) Representation of 2CV (b) Representation of IWV

Figure 6. Latent representation resulting from LSTM model for testing samples in the VLSP dataset

Last, comparing between 2CV and two models using two advanced features, such as, W2V_SentiWord and W2V_BERT, these tables show that 2CV is often better than both W2V_SentiWord and W2V_BERT. For example, both W2V_SentiWord and W2V_BERT are only better than 2CV on two configurations when using LSTM (Table 4) while 2CV is better than these two methods on six configurations. Similar results are also observed when using CNN (Table 5).

The reason for the better performance of our method compared to W2V_SentiWord is that the sentiment word vector in Vietnamese has only one dimension (compared to six in English) and it may not be enough to learn a useful feature for the SC problem. For the W2V_BERT model, one possible reason for its slightly worse performance compared to 2CV is that we only used the word vector of BERT to combine with the W2V vector and ignored

[6]It is noted that the result of our models is slightly lower than the result in [12] on VLSP dataset. The reason could be that in [12], the authors used an ensemble model for SC while our models are single models. In the future we plan to extend our models to become ensemble models using the techniques in [16].



(a) Representation of 2CV (b) The representation of IWV

Figure 7. Latent representation resulting from CNN model for testing samples in the VLSP dataset

its document vector [7]. For the other tested models, it is often based on only one channel (Single W2V) or the pre-trained features (Pre-trained and IWV) are not updated during the training process thus their performance is often inferior to 2CV.

6.2. Visualization analysis

Figure 6, and Figure 7 visualize the document representation of 2CV and IWV resulting from the LSTM-based and CNN-based models trained on the VLSP dataset, respectively. For the sake of visualization, we project the document representation into two-dimension vectors by the t-SNE method [27]. In these figures, three classes of samples in the VLSP dataset are represented by blue minus, red circle, and green square, respectively. Figure 6 (a), and Figure 7 (a) are the representation resulting from our proposed method (2CV) with the LSTM-based and CNN-based networks, respectively. The rest are results from the IWV method. We can observe that the document representation resulting from our proposed technique is more distinguishable than those of the IWV method. The document representations of the proposed model tend to polarize the classes more clearly. Thus, the representation of the 2CV method can fascinate the classification algorithms more effectively.

6.3. Computational time analysis

Table 6. Comparison of training and testing time for VLSP dataset

	Training time (second)	Testing time (second)
2CV LSTM	1977	8.4
IWV LSTM	1166	6.5
2CV CNN	3182	11
IWV CNN	838	5.4

[7] In the future, we will conduct more experiments to verify if using the document vector of BERT can further improve its performance in SC.

Table 6 presents the computational time for training and testing of 2CV and IWV performing on VLSP. The statistics are recorded on a personal computer of the following configuration: Core i5-7400 CPU, 8 GB RAM, Ubuntu 18.04. It can be seen that, although the accuracy of 2CV are more satisfactory than IWV, this method takes longer training and testing time than IWV.

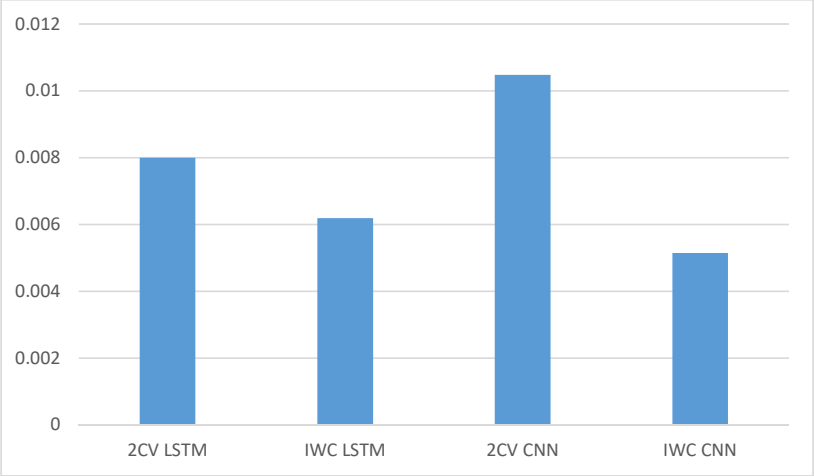


Figure 8. Time to predict a sample of 2CV and IWV

We also estimate the average time to predict a sample using 2CV and IWV. The results show that the predicted time for a sample by the IWV method is faster than the 2CV method. In particular, the 2CV based on CNN consumes the highest prediction time.

6.4. Error analysis

This subsection presents some samples in which our model predict correctly while IWV is incorrect. This result is listed in Table 7.

It can be seen from this table that our model is better than IWV when the structure of the input documents is complicated. For example, cases 1, 15, 18 contain the negative words ("không" and "k") and our model is better than IWV in these cases. Similarly, cases 10 and 13 are the short sentences that contain many sentiment words of opposite meaning and this makes it difficult for IWV. Other cases are the documents of neutral label. Since these document does not contain the sentiment words, it is difficult to predict for all models including our model. However, our model is still better than IWV in predicting the neutral documents.

7. CONCLUSIONS

In this paper, we have designed a new model based on deep neural networks for sentiment analysis. In our model, two features, i.e., Word2vec and POS are simultaneously learned using two deep networks and then are combined to form a new higher-level feature. The

Table 7. Some samples which 2CV and IWV result in different prediction

Dataset	Reviews	IMV	2CV	Label
VLSP	*1. Nếu đúng như thế thì không ai mua đâu.*	Neu	Neg	Neg
	2. Cấu hình ổn nhưng khi sử dụng thật thất vọng. Có dại mới mua.	Pos	Neg	Neg
	3. Wow, giá nó rẻ hơn 25% so với những sp cùng phân khúc luôn kìa.	Neu	Pos	Pos
	4. Mình bảo đảm bạn ko bao h có giá đó.	Neg	Neu	Neu
	5. Luôn có thiện cảm với chất lượng của htc.	Neg	Pos	Pos
VSFC	*6. Tính thực tế cũng cao so với việc thi lý thuyết lấy điểm.*	Neg	Pos	Pos
	7. Chỉ có thầy wzjwz320 dạy , em cảm thấy học và thi không ăn nhập gì với nhau.	Pos	Neg	Neg
	8. Với lại đặc thù sinh viên trường em nó khác cô ạ.	Neg	Neu	Neu
	9. Hy vọng thầy có thể hướng dẫn tốt hơn các kỹ năng của môn học.	Pos	Neg	Neg
	10. Giảng viên dạy chậm , dễ tiếp thu.	Neg	Pos	Pos
Foody	*11. Mình ăn ở đây 5 lần r ý . 2 lần đi với đám bạn , 2 lần đi với cô , 1 lần đi với nhỏ bạn thân ! Ngon mà giá hợp lí nữa ..*	Neg	Pos	Pos
	12. Hôm nay phát hiện quán này , nhìn bên ngoài chưa có gì thu hút nhưng khi mở cửa vào bên trong rất thích không gian quán . Đặc biệt là nước uống giá phù hợp ..	Neg	Pos	Pos
	13. Quán ăn ngon , hơi bị đông dù mình đến buổi trưa . Phần 2 người ăn đầy đủ . Cô chủ dễ thương , lúc ra tính tiền cho 2 cái kẹo :).	Neg	Pos	Pos
	14. Ăn ở đây rùi . Khá là ngon . Nhưng mà đợi hơi lâu nhé ... Vì quá đông !.	Pos	Neg	Neg
	15. Quán nhỏ . . Menu k đa dạng . . Nhiều món ăn k hấp dẫn . . Bánh căn k ngon . . Chỉ có món bánh tráng Thái Lan vs chuối là ngon và yaourt trà xanh thì đc . ..	Pos	Neg	Neg
AiVN	*16.Hàng ghi made in mỹ nhưng mã vạch lại ở Việt Nam . Nhiều Hạt đã mốc*	Pos	Neg	Neg
	17. Lần đầu mua thấy đẹp mua thêm cái sau cũng là loại quần đó mà vừa rộng vừa xấu ... Chào tạm biệt.	Pos	Neg	Neg
	18. Không ổn cho lắm . . nói đường làm nẻo . Sản phẩm không tốt.	Pos	Neg	Neg
	19. Thời gian giao hàng rất nhanh . đi lúc 8h nvậy có trễ không mà đồ ăn không đc tươi như mấy buffet ở chỗ khác.	Neg	Pos	Pos

higher-level feature is then passed through a Softmax function for classification. We have carried out extensive experiments to evaluate the performance of the proposed method on four Vietnamese datasets. The experimental results demonstrate that our proposed method can enhance the ACC, F1, and AUC scores for SC compared to two single models and three state-of-the-art ensemble methods.

One limitation of our proposed model is that it takes longer processing time compared to the previous model (IWV). The reason is that we used two deep neural networks to learn the document representation, thereby increasing training and testing time. However, the prediction time of our proposed model for a single document is negligible and can be

applicable to many real-world applications. In the future, we will extend the application of 2CV to analyze the user's opinion in other languages, especially English, and other domains like industry, business, healthcare, politics and academics.

ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2019.05.

REFERENCES

[1] L. Arras, G. Montavon, K.-R. Müller, and W. Samek, "Explaining recurrent neural network predictions in sentiment analysis," *arXiv preprint arXiv:1706.07206*, 2017.

[2] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, "Neural sentiment classification with user and product attention," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1650–1659.

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[4] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proceedings of The 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 513–520.

[5] Y. Goldberg, "A primer on neural network models for natural language processing," *Journal of Artificial Intelligence Research*, vol. 57, pp. 345–420, 2016.

[6] Z. Hailong, G. Wenyan, and J. Bo, "Machine learning and lexicon based methods for sentiment classification: A survey," in *2014 11th Web Information System and Application Conference*. IEEE, 2014, pp. 262–265.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May–Jun. 2015, pp. 103–112. [Online]. Available: https://www.aclweb.org/anthology/N15-1011

[9] Y. Kim, "Convolutional neural networks for sentence classification," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 08 2014.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[12] P. Q. N. Minh and T. The Trung, "A lightweight ensemble method for sentiment classification task," 11 2016.

[13] R. Moraes, J. F. Valiati, and W. P. G. Neto, "Document-level sentiment classification: An empirical comparison between svm and ann," *Expert Systems with Applications*, vol. 40, no. 2, pp. 621–633, 2013.

[14] A. Mudinas, D. Zhang, and M. Levene, "Combining lexicon and learning based approaches for concept-level sentiment analysis," in *Proceedings of The First International Workshop On Issues Of Sentiment Discovery And Opinion Mining*. ACM, 2012, p. 5.

[15] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: Pre-trained language models for Vietnamese," *arXiv preprint*, vol. arXiv:2003.00744, 2020.

[16] H.-Q. Nguyen and Q.-U. Nguyen, "An ensemble of shallow and deep learning algorithms for vietnamese sentiment analysis," in *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*. IEEE, 2018, pp. 165–170.

[17] H. T. Nguyen, H. V. Nguyen, Q. T. Ngo, L. X. Vu, V. M. Tran, B. X. Ngo, and C. A. Le, "Vlsp shared task: Sentiment analysis," *Journal of Computer Science and Cybernetics*, vol. 34, no. 4, pp. 295–310, 2018.

[18] G. Rao, W. Huang, Z. Feng, and Q. Cong, "Lstm with sentence representations for document-level sentiment classification," *Neurocomputing*, vol. 308, pp. 49–57, 2018.

[19] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, pp. 14–46, 2015.

[20] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi, "Sentiment analysis based on improved pre-trained word embeddings," *Expert Systems with Applications*, vol. 117, pp. 139–147, 2019.

[21] B. Shin, T. Lee, and J. D. Choi, "Lexicon integrated cnn models with attention for sentiment analysis," *arXiv preprint arXiv:1610.06272*, 2016.

[22] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2006, pp. 1015–1021.

[23] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015.

[24] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective lstms for target-dependent sentiment classification," *arXiv preprint arXiv:1512.01100*, 2015.

[25] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1422–1432.

[26] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 1555–1565.

[27] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html

[28] K. Van Nguyen, V. D. Nguyen, P. X. Nguyen, T. T. Truong, and N. L.-T. Nguyen, "Uit-vsfc: Vietnamese students' feedback corpus for sentiment analysis," in *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 2018, pp. 19–24.

[29] Q.-H. Vo, H.-T. Nguyen, B. Le, and M.-L. Nguyen, "Multi-channel lstm-cnn model for vietnamese sentiment analysis," in *2017 9th international conference on knowledge and systems engineering (KSE)*. IEEE, 2017, pp. 24–29.

[30] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, "Vncorenlp: A vietnamese natural language processing toolkit," *arXiv preprint arXiv:1801.01331*, 2018.

[31] X.-S. Vu and S.-B. Park, "Construction of vietnamese sentiwordnet by using vietnamese dictionary," *arXiv preprint arXiv:1412.8010*, 2014.

[32] J. Xu, D. Chen, X. Qiu, and X. Huang, "Cached long short-term memory neural networks for document-level sentiment classification," *arXiv preprint arXiv:1610.04989*, 2016.

[33] Y. Yin, Y. Song, and M. Zhang, "Document-level multi-aspect sentiment classification as machine comprehension," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2044–2054.

[34] S. Zhai and Z. M. Zhang, "Semisupervised autoencoder for sentiment analysis," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[35] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[36] X. Zhou, X. Wan, and J. Xiao, "Attention-based lstm network for cross-lingual sentiment classification," in *Proceedings of The 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 247–256.

Received on February 15, 2020
Revised on July 29, 2020