DOI 10.15622/ia.2020.19.6.5

N.H. PHAT, N.T.M. ANH

VIETNAMESE TEXT CLASSIFICATION ALGORITHM USING LONG SHORT TERM MEMORY AND WORD2VEC

Phat N.H., Anh N.T.M. **Vietnamese Text Classification Algorithm using Long Short Term Memory and Word2Vec.**

Abstract. In the context of the ongoing forth industrial revolution and fast computer science development the amount of textual information becomes huge. So, prior to applying the seemingly appropriate methodologies and techniques to the above data processing their nature and characteristics should be thoroughly analyzed and understood. At that, automatic text processing incorporated in the existing systems may facilitate many procedures. So far, text classification is one of the basic applications to natural language processing accounting for such factors as emotions' analysis, subject labeling etc. In particular, the existing advancements in deep learning networks demonstrate that the proposed methods may fit the documents' classifying, since they possess certain extra efficiency; for instance, they appeared to be effective for classifying texts in English. The thorough study revealed that practically no research effort was put into an expertise of the documents in Vietnamese language. In the scope of our study, there is not much research for documents in Vietnamese. The development of deep learning models for document classification has demonstrated certain improvements for texts in Vietnamese. Therefore, the use of long short term memory network with Word2vec is proposed to classify text that improves both performance and accuracy. The here developed approach when compared with other traditional methods demonstrated somewhat better results at classifying texts in Vietnamese language. The evaluation made over datasets in Vietnamese shows an accuracy of over 90%; also the proposed approach looks quite promising for real applications.

Keywords: Text Classification, Natural Language Processing, Data Processing, Long short term memory, Word2Vec

1. Introduction. Automatic text classification is intended for processing new documents based on their similarity with the other ones in the training model. In this paper the text classification algorithm is proposed aimed at solving certain issues (classifying topics and positive-negative comments) based on titles. The following example is given to explain the text classification. At that, the news dataset is selected and represented as:

$$N = (n_1, n_2, ..., n_n). \quad (1)$$

The news corresponding labels are:

$$C = (c_1, c_2, ..., c_m). \quad (2)$$

The articles labeling will be performed automatically according to label (C).

The traditional methods often classify documents based on dictionaries. However, the development of deep learning (DL) models has been much more effective and used widely for classifying documents [1, 2].

Besides, the methods for Vietnamese language are quite limited. The most difficult point for processing documents in Vietnamese is to determine word boundaries. In English language words are groups of meaningful characters separated by spaces in sentences. Therefore, it is not difficult to select in a sentence the words in English language. However, word boundaries are not defined for Vietnamese and depend on context of sentences. Several key properties are:

– independent syntax;

– single, complex, repeated, and compound words;

– phrase clustering.

Figure 1 gives an example of Vietnamese phrase clustering [3]; and shows that there exist two ways to understand this sentence. If the second way of words clustering is accepted, the sentence will have not the same meaning as the method.

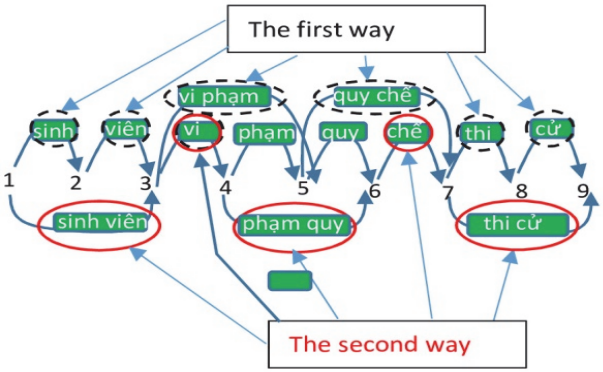

Fig. 1. An example of clustering words in Vietnamese language

Words' clustering in sentences is an important step in preprocessing documents in Vietnamese. If the sentence is identified by the first method, it would be classified as a law. Otherwise, certain misunderstanding would lead to another label. Therefore, the accuracy of phrase clustering is very important. If the clustering of words is not reliable, the label classification of text may be wrong.

In the paper, the long short-term memory (LSTM) model is used at first to classify text since gates of LSTM are able to filter information similar to input ports. Therefore, the data from the past will be adjusted.

Besides, here also proposed to combine LSTM and Word2vec methods to improve the method accuracy.

The rest of the paper is organized as follows. Section 2 describes the related works; Section 3 presents the theoretical basis; Section 4 shows the results of the model use; Section 5 contains conclusions and some new directions proposed for further development.

2. Related Work. The amount of text data must be processed prior to their use. The problem is how to extract information from the data source. The data nature and their characteristics should be clearly understood in order to apply the necessary methodologies. Therefore, time and effort for their classifying will be saved.

DL models have appeared to be quite successful at natural language processing (NLP) [3-12]. Applications of DL to NLP are mentioned as phrase classification algorithm [13] and the main content of Vietnamese text [14].

Currently, there exist many methods to improve English texts classification based on artificial neural (AI) networks [15-25]. In [15] the authors use new LSTM model to classify text. The model with prior training is able to solve the multi-dimensional data processing problem by traditional methods. In [20] the authors use convolutional neural network (CNN) and recurrent neural network (RNN) for classification. In [16] the authors combine two models CNN and bidirectional recurrent neural network (BRNN). In the model the authors use the bidirectional class to replace the pooling class in CNN and help to store the long-term dependencies of input chains.

In addition, the support vector machine (SVM) algorithm is used to reduce required memory [17]. SVM algorithm is well applicable to the text classification problem. It saves memory since only a subset of points is used for training and forming process for new data. Therefore, the necessary points are stored while making decisions in a result of / for decision-making. SVM possesses certain flexibility due to change between linear and non-linear methods. However, it does not clearly indicate the calculus of probability yet, since its classification only focuses on dividing objects into two layers by super-flat.

Many studies have been performed [18, 26-28] in regard to Vietnamese texts classification. Classification of texts in Vietnamese via traditional methods and topics is often based on [18]. Topic model uncovers-abstract documents. This method guarantees stability and provides a relative accuracy. However, it is quite difficult and time-consuming as well as costly. Therefore, SVM algorithm is applied to classifying texts in Vietnamese [26]. This classification is able to adjust the parameters automatically. However, it displays worse results comparing with an

application to English language (the accuracy for Vietnamese language is 80.72% comparing with 89% for English [17]).

In [27] the authors used traditional methods like Na¨ıve Bayes (NB) and maximum entropy (ME). NB is a simple method to solve problems regarding data classification based on statistic. However, its disadvantage is that there is no link between the characteristics. Therefore, ME is used to estimate label probability based on sentence characteristics. Besides, in [27] the authors also use DL method as LSTM and Bi-LSTM for Vietnamese language classification. In [28] the authors employ a combination of LSTM and CNN for separating words in a sentence. This approach helps the model to learn both adjacent and distant data. To the best of our knowledge, the classification of words by subject using LSTM and Word2vec has not been published before.

Based on the analyzed above, the topic based DL model for text classification could be proposed since it has demonstrated certain improvements for classifying documents in Vietnamese. In the model the LSTM and Word2vec methods are combined. As a result, the model improved both accuracy and speed processing for input of data in Vietnamese language.

3. System

3.1. The System Overview. An overview of the system is shown in Figure 2. The system consists of two main blocks, namely training and predicting parts. In two blocks, a combination of two algorithms Word2vec and LSTM is proposed to increase accuracy for classification of texts in Vietnamese. Besides, the preprocessing step suitable for data in Vietnamese is used. The steps details are given in the following sections.

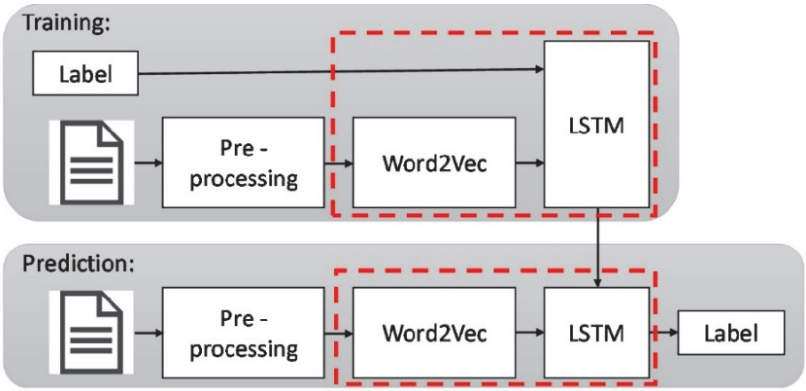

Fig. 2. Diagram of system structure

3.2. Data Collection. The difficulty for classifying is that the dataset of topics in Vietnamese is limited. Datasets usually consist of 50 to 100 raw

texts. In this paper the VNTC dataset [4] suitable for the done research is used. The dataset is updated and information from official electronic newspapers like [29-32] is chosen by users according to each topic. It contains 10 topics of 33756 and 50373 articles for training and testing respectively. Data are selected and aggregated as shown in Figures 3 and 4.

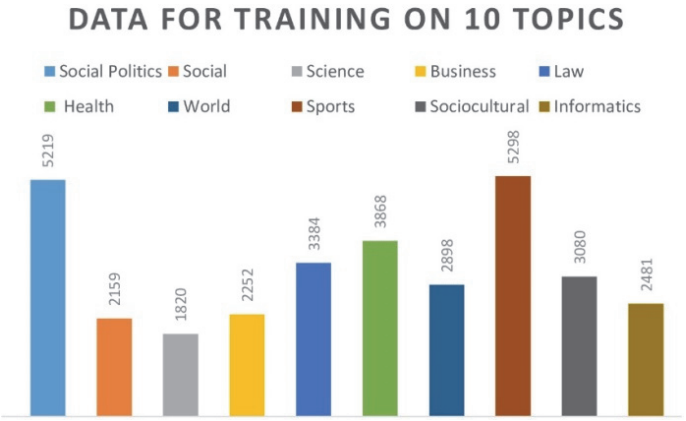

Fig. 3. Result of dataset training [4]

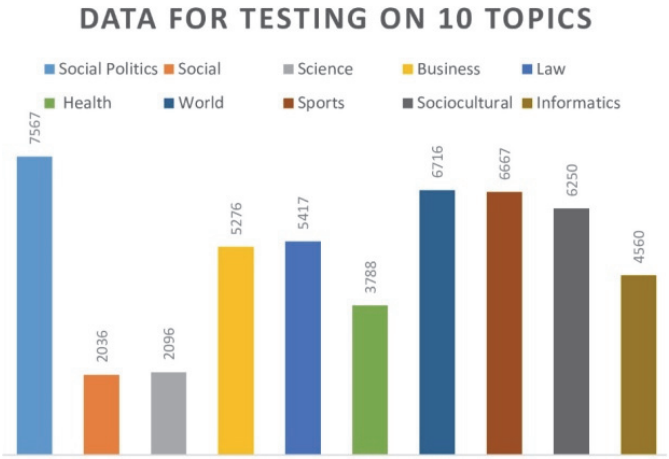

Fig. 4. Result of dataset testing [4]

3.3. Data Preprocessing. At the preprocessing step three small steps namely Vietnamese words separation, data cleaning, stop-word separation are executed as follows.

Vietnamese words separation. As was mentioned in Part 1, the separation of Vietnamese words greatly affects the output results. In recent years the Vietnamese language processing community has grown stronger. There appeared many libraries related to NLP for Vietnamese language that have produced good results [18, 26-28].

Table 1 shows the accuracy of methods for separating Vietnamese words. In the current paper the *vnTokenizer* and *Python* language [33] are used. To separate words the Vietnamese *Pyvi* library for VNTC dataset [33] is used. This library is a combination of maximum matching (MM) algorithm and SVM model to solve both ambiguity and unknown words recognition to improve the shortcomings for other methods. It proposes a language classifier for Vietnamese as follows: O – single word for one language, B – first language for one multi-language word, I – intermediate language for one multi-language word, and E – final language for one multilingual word. MM is considered to be the simplest dictionary based on splitting word; it attempts to match the longest word dictionary. However, this method is not able to solve the problem of ambiguity since it only recognizes words in dictionary. Therefore, the combined system of MM + SVM has reduced the ambiguity by contextual words. In [33] the library has an accuracy of 97,86% for Vietnamese words.

Table 1. Accuracy of Vietnamese words separation methods

Tokenizer	F1-Score (%)
RDRsegmente [34]	97,90
jPTDP-v2 [35]	97,90
UETsegmenter [10]	97,87
vnTokenizer [3]	98,5
JvnSegmenter [36]	97,06

Data cleaning. After separating words the text reveals many special characters and punctuation that simplify the system. To solve the problem, all uppercase letters were converted into lowercase and punctuation was removed.

Stop-word separation. Then the stop-words are to be removed. Stop-words are understood as words not important for classification. Besides, a number of concatenation or quantitative words is not discriminatory at their classifying. In addition, stop-words have no taxonomic values that appear in most documents. Therefore, these words are eliminated to reduce computation time and memory during the processing. At this step the stop-words Vietnamese dictionary [37] are used. Upon these words removing the text will be considered a set of the representation important words.

3.4. Extracting Feature by Word2vec. After removing stop-words the text with the important words will be received. However, there are still many documents' characteristics, and they have to be shorten.

Then words will be vectorized. The vector representation of words plays an important role. Word embedding is responsible for mapping from word or phrase to a real number vector.

In this section the Word2vec model is used. The model represents words as real vectors with specified dimensions. It is an unsupervised learning model, and is one of the first models of word embedding that uses vector each word based on contexts; it maps a set of words to a vector space where each of them is represented by n real numbers. Word2vec is the neural network with only one hidden layer. Input is a large set of words and output are vector spaces. Each single word is assigned a corresponding vector as shown in Figure 5 [38, 39].

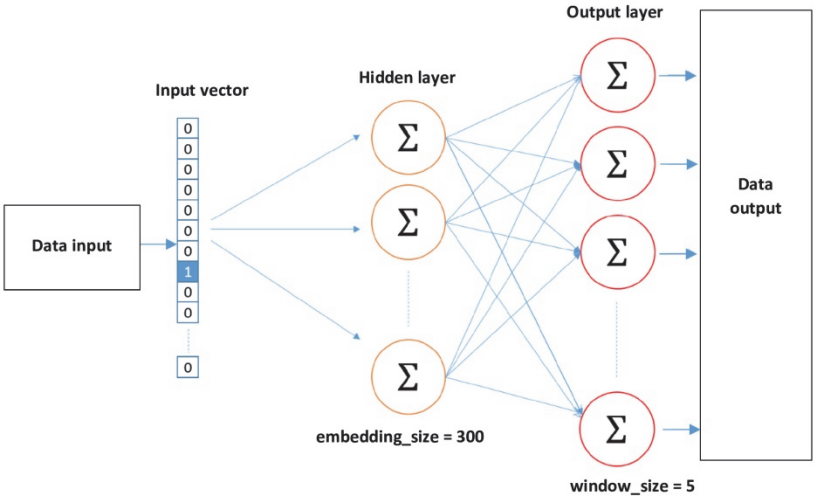

Fig. 5. Proposal of Word2vec model with *embedding_size* = 300 and *window_size* = 5

The model of input is one-hot-vector: Each word will be in a form $x_1, x_2, ..., x_v$, where v is the number of vocabulary. Each word is a vector with a value of "1" equivalent to the order of words in vocabulary and the rest will be of "0" value.

The matrix between the input and hidden layer is W (its dimension is $V \times N$) whose activation function is linear. The matrix between the hidden layer and the output is W' (its dimension is $V \times N$) whose activation function is *softmax*.

Each row of W is N-dimensional vector representing V_w. Each row of W is v_W^T.

Output matrix of the hidden layer is $W' = w'_{i,j}$. The score for each word is calculated as follows:

$$u_j = v'_{wj} * h, \tag{3}$$

where v'_{wj} is j column of W' matrix. Then the *softmax* trigger function is used as follows:

$$P(w_j / w_I) = yi = \frac{\exp(u_j)}{\sum_{j'=1}^{v} \exp(u'_j)} = \frac{\exp(v'_{Wj} v_{WI})}{\sum_{j'=1}^{v} \exp(v'^T W_{j'} v_{WI})}, \tag{4}$$

where v_w and $v_{w'}$ are two vectors.

After training model, the weight of each word is updated continuously.

Therefore, the calculations can be performed by distances. The words appear together in context or synonyms belonging to the same vocabulary.

Word2vec algorithm has two approaches (as shown in Fig. 6) as follows:

– continuous bag of words (CBOW) model gives context (surrounding words) and guesses the appearing probability from destination;

– skip-gram model gives the current word and guesses the probability of context words.

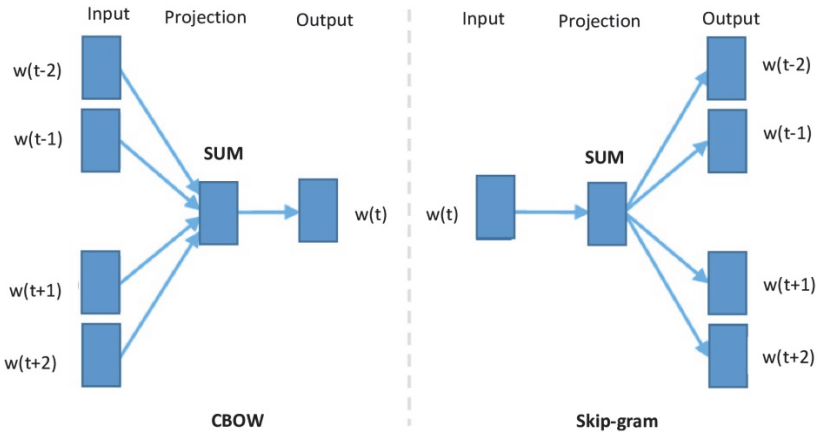

Fig. 6. Two approaches of word2vec algorithm [40]

The neural network architecture of skip-gram and CBOW consists of three layers:

1. *Input* is the input of network and context words around a target or a present word.

2. *Projection* contains the hidden layers of network for parameters calculating.

3. *Output* is a softmax function that calculates the probability for target words or distributes into vocabulary. Based on the feed and back propagation models, the methods would be found to optimize the parameters to predict the results accuracy.

CBOW advantage is that it requires less memory to store large matrices. It is essentially probabilistic since implementation is better than Skip-gram. However, the disadvantage is that words with different meanings are still represented by one vector of words.

3.5. Labeling by LSTM. RNN for NLP has been of interest lately [15-25]. The main idea of RNN is to use sequences of information. In traditional neural networks all inputs and outputs are independent of each other. If inputs go through the hidden layer and the outputs connect among the classes, they will be combined via a function to calculate the current and output layers.

The RNN calculation is performed as follows:

– x_t is the input at t that is one-hot vector corresponding to size of $n \times 1$;

– s_t is the hidden state at t that is calculated based on both the pre-hiding state and input. The front words will affect the output as follows:

$$s_t = f(Ux_t + Ws_{t-1}). \quad (5)$$

The function f is usually a linear function as tang hyperbolic (tanh). The function is used to adjust the information passing through the system. All values are assigned to the range (-1, 1). When vectors go through neural network, they undergo many calculations. In the process, several components become too large. The function will help to refine the difference. To make calculations for the first element, s_{t-1} to 0 or random values should be initialized. The memory is empty without data;

– o_t is the output at t that is a probability vector of predicting words by learning information from all previous inputs as follows:

$$o_t = g(Vs_t), \quad (6)$$

where g is the activation function.

The RNN is described as shown in Figure 7.

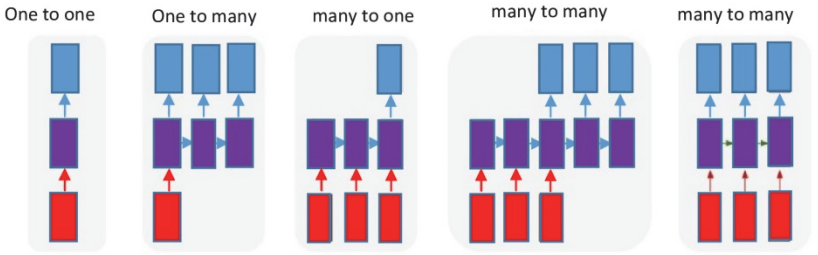

Fig. 7. Types of problems of RNN [41]

– *one to one* is for neural network (NN) and CNN with one input and output. For example, the input is the image and the output identifies whether it is a motorcycle;

– *one to many* is a problem with one input and many outputs. For example, the input is one image and the output is many descriptive words.

– *many to one* has many inputs and one output. For example, the input is multiple images from video and the output is one action.

– *many to many* has many inputs and outputs, for example translating from English into Vietnamese.

LSTM is an extending version of RNN designed to solve long-term dependencies. RNN is a neural network containing a loop. The network is capable of storing information. Information is passed from layer to layer. The output of hidden layers depends on the information. RNN has been commonly used for NLP or sequential data problems. However, its architecture is simple since the ability to link long-distance layers is poor. It is incapable of remembering information from long distance data and, thus, the first elements of input sequence usually has no great effect on the following steps. As a result, RNN is influenced by the derivatives during learning and vanishing gradient. LSTM network is designed to solve the problem; it only remembers relevant and other information to discard them.

The LSTM network consists of many interconnected cells as shown in Figure 8 [21]. The idea of LSTM is to add cell internal state (s_t) and three ports of input and output (f_t forget gate, i_t input, and θ_t output). At each time step, the ports receive the input x_t (representing an element of input sequence) and h_{t-1} that obtains from output of the memory cells from previous time step ($t-1$). The port has the function of selecting information for each different purpose. They are defined as follows:

– *Forgotten gate* removes unnecessary information from s_t;

– *Input port* filters the necessary information to add s_t.

– *Output port* determines information from s_t that is used as output.

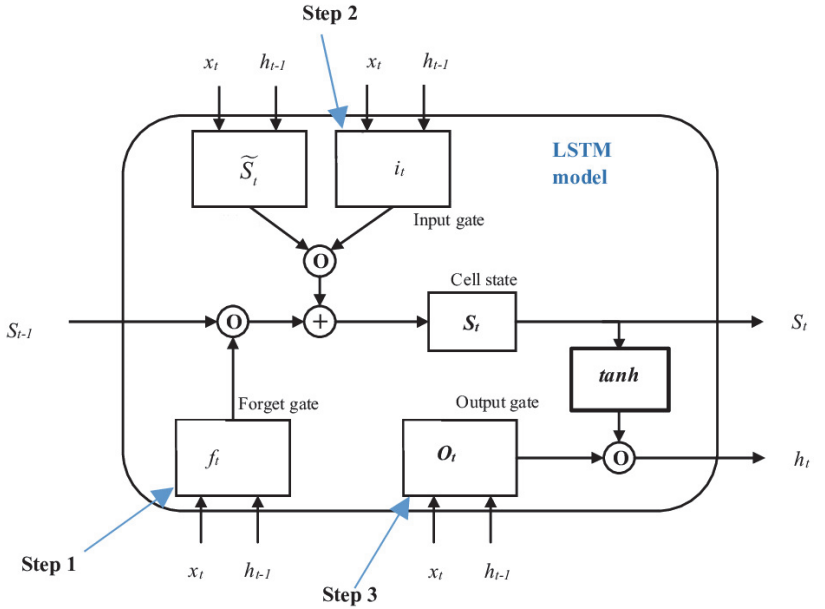

Fig. 8. An example of LSTM model

During the implementation, s_t and h_t are calculated as follows:

At the first step, LSTM cell determines the information that needs to be removed from s_t at $t-1$. The value of f_t is calculated based on x_t, h_{t-1} and b_f. The sigmoid function converts all activation values between "0" and "1" as follows:

$$f_t = \sigma(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f). \tag{7}$$

At the second step, LSTM cell determines the information that needs to be added to s_t. The step involves two calculations for $\tilde{s}_t$ and f_t. $\tilde{s}_t$ represents information that can be added to s_t. as:

$$\tilde{s}_t = tanh(W_{\tilde{s},x}x_t + W_{\tilde{s},h}h_{t-1} + b_{\tilde{s}}). \tag{8}$$

i_t of input port at time t is calculated:

$$i_t = \sigma(W_{i,x}x_t + W_{i,h}h_{t-1} + b_i). \tag{9}$$

In the next step, s_t. is calculated as follows:

$$s_t = f_t * s_{t-1} + i_t * \tilde{s}_t. \tag{10}$$

Finally, h_t is:

$$o_t = \sigma(W_{o,h} h_{t-1} + b_o); \tag{11}$$

$$h_t = o_t * tanh(s_t), \tag{12}$$

where $W_{\tilde{s},x}, W_{\tilde{s},h}, W_{f,x}, W_{f,h}, W_{i,x}$ and $W_{i,h}$ are weight matrices in each LSTM cell and $b_f, b_{\tilde{s}}, b_i$, and b_o are bias vectors.

3.6. Convolution Neural Network (CNN). CNN has a number of advantages for image and text processing. Therefore, it is studied in regard to NLP application. However, the biggest disadvantage of RNN is that it takes a too much time to train a model. Therefore, the researchers hope to use CNN to reduce training time while still achieving the same results as those of RNN.

CNN is a type of artificial neural network (ANN) with multiplayer perceptron's, namely convolution and pooling. It is understood as a convolutional class that transforms an input into a different output. CNN simply includes a few layers of convolution combining nonlinear activation functions as *ReLU* or *tanh* to create abstract of information for the next layer.

In the feed-forward neural network (FNN) model, the layers are directly connected to each other through weight (w). These layers are called the fully connected or affine layers. In CNN model layers are linked through convolution. Specifically, the next layer is a convolution result from the previous layer since local connections exist.

CNN is a collection of convolution classes that overlaps and used for nonlinear activation functions similar to ReLU and then for activating weights in nodes. Each class after the activation function will generate abstract of information for the next one. In the FNN model each input is used for subsequent layers.

In training process CNN automatically learns values through filter classes. There are two aspects to consider; they are location invariance and compositionality. If this object is projected under different angles, the accuracy of the algorithm will be significantly affected with the same object. Pooling layer is the immutability for translation, rotation, and scaling. Local aggregation gives levels of information representation from lower to higher and abstract through the convolution. The composite layer is inserted periodically after each pair of convolutions and the nonlinear activation layer. It reduces the spatial dimensions of a sample and the total parameters of network. Therefore, the

aggregate layer makes sense to control the overload in network. There are two types of algorithms common in aggregation classes as follows:

– *MaxPooling* extracts the maximum value of child function;

– *AveragePooling* extracts the average value of sub-functions.

RNN is a specialized neural network that processes a series of values. At the network application, the text to be a two-dimension matrix is considered where rows are the tokens of sentence, and their values are the vector values of each token. The size of input matrix is as follows: *height = len(tokens)* and *width = dimension(wordembedding)*. At convolution implementation, the filter (also two-dimension matrix) is used with *widthvalue = In put Matrix*, and height value is usually 3, 4 or 5. The application of CNN to NLP can be described as shown in Figure 9.

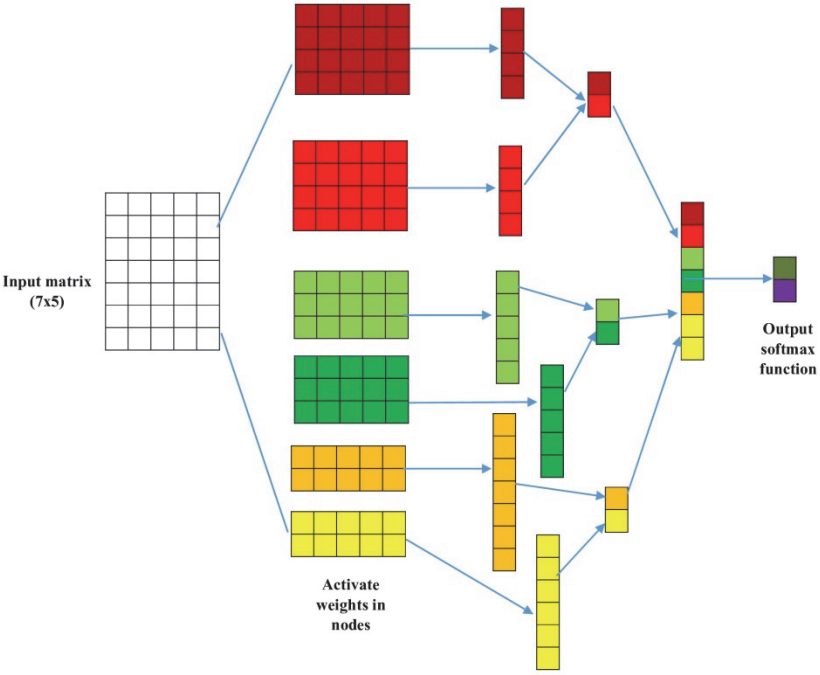

Fig. 9. Diagram of applying CNN to NLP model [42]

4. Simulation and Results

4.1. Setup

4.1.1. Dataset. VNTC dataset is used [4] including 33756 and 50373 articles for training and testing sets, respectively. The input will be separated from the words via *pyvi* library [43] and cleaned stop-words (special characters and punctuation).

4.1.2. Pre-processing Data. All documents after separating from via *pyvi* library are processed into words for the next processing step. The function of auxiliary words will be omitted to increase the performance as well as to reduce the number of characteristics for classification model.

4.1.3. Extracting Feature. The text is converted into-vector to create a directory for using document representations. Each word will be assigned a natural value. They are placed with the corresponding numbers. Each text is converted into an array of natural numbers that are assigned fixed words. It will be converted into a vector form consisting of one number and "0"elements.

Word2vec will be used to improve text features. It was found out that the pre-train of the *gensim* library is not suitable for the considered dataset. Therefore, the dataset was built by the authors. The Word2vec model was subjected to retraining in order to improve the problem with $embedding_{size}$ = 300, $window_{size}$ = 5 for 80000 documents. Figure 10 is the result of training model using $t - SNE$.

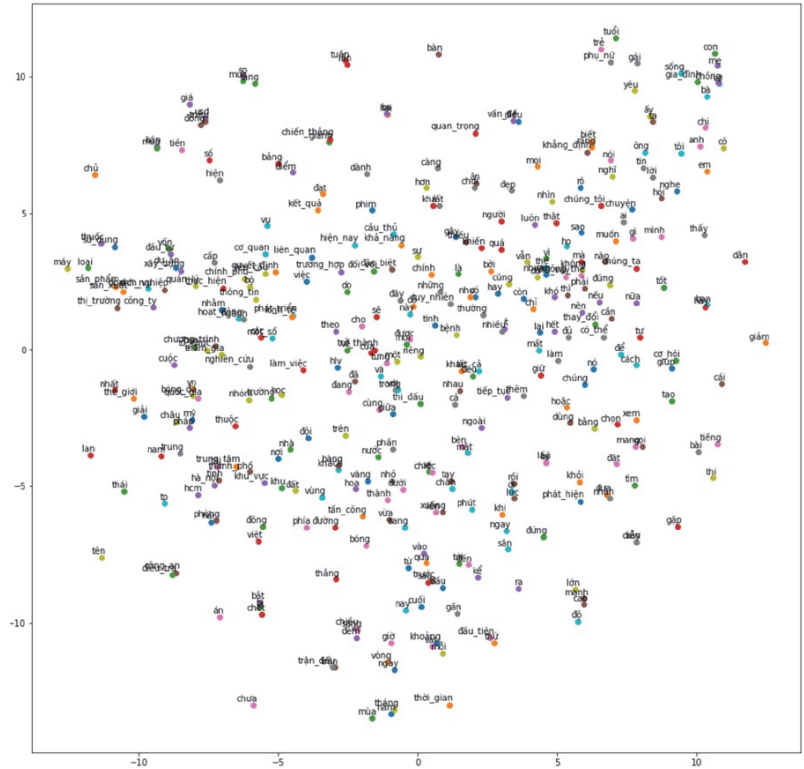

Fig. 10. Selection words of vector space from Word2vec

4.1.4. Text Classification. Since a newsletter usually ranges from 500 to 1000 words, the following was chosen: the maximum length of each document is 400 and the size of word embedding is 300 in order to reduce the input of network. The choice of text size and vector space to solve the multi-dimensional data problem of traditional methods is very important. Therefore, was selected the training of the model with a hidden class of 128 units and use of *RMSProp* with a learning rate of 0.001 for the optimal function and a dropout of 0.4.

The embedding layer with a 300-dimensional vector is used to represent a word. Then, was used spatial dropout 1D to remove 1D feature of maps and increase independence between them. The LSTM class is used with 128 memory units. The model of output layer consists of 10 topics. Activation function is used as *softmax* to classify multiple layers and loss function is categorized as cross_entropy. In addition, *batchsize* is set to 64 and *epochs* is 35. Detailed flowchart of LSTM algorithm is shown in Figure 11.

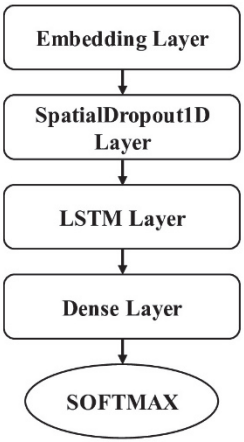

Fig. 11. Algorithm flowchart for LSTM model

4.1.5. Evaluation Criteria. The precision, recall, and F1-score criteria were used to evaluate the model performance. The parameters are shown in Table 2 [44] where:

– TN is the result where model accurately predicts the negative class;

– TP is the result where model accurately predicts the positive class;

– FN is the result where model incorrectly predicts the negative class;

– FP is the result where model incorrectly predicts the positive class.

Table 2. Confution matrix

Labels	Negative prediction	Positive prediction
Negative act	True negative(TN)	False positive(FP)
Positive act	False negative(FN)	True positive(TP)

Precision is the ratio of positive points determining by *TP* and *FP* as:

$$Precision = \frac{TP}{TP + FP}. \quad (13)$$

Recall is the ratio of positive points determining by *TP* and *FN* as:

$$Recall = \frac{TP}{TP + FN}. \quad (14)$$

F1-score is the harmonic mean of precision and recall (assuming these two quantities are non-zero) determining by the expression:

$$\frac{2}{F1} = \frac{1}{Precision} + \frac{1}{Recall}. \quad (15)$$

4.2. Results. The results of the model are shown in Tables 3 and 4.

Table 3. Comparing the results of LSTM model and the proposal

Model	Precision (%)	Recall (%)	F1-score (%)
LSTM	92.36	91.81	92.09
The proposal (LSTM + Word2Vec)	95.55	95.93	95.74

Table 4. Comparing the results of CNN model and the proposal

Model	Precision (%)	Recall (%)	F1-score (%)
CNN	84.48	83.02	83.25
The proposal (CNN+ Word2Vec)	84.13	84.89	84.01

Table 3 shows the results of classification for the text in Vietnamese. The accuracy of model using LSTM with Word2vec is 4% higher than that

using LSTM. The result is achieved by pre-training of the Word2vec model. Besides, Word2vec is able of preventing the overfitting that reduces the number of training parameters and improves accuracy.

Also was executed the model based on CNN and CNN + Word2vec as shown in Table 4. Two important features of CNN are local sensing and weight sharing. When CNN algorithm is applied to the problem, the input is considered as a two-dimensional matrix of size 400 × 300. In the case under consideration the fixed length of each text is 400 and size of each word is 300. In the CNN model is used the Conv1D class with a filter of 128 and kernel-size of 5. For the Pooling class, we use *MaxPool1D* with $pool - size = 2$. In the next class, we continue to use *Conv1D* with a filter of 256. Finally, we use *GlobalAvgPool1D* with the *softmax* activation function.

The results given in Tables 3 and 4 show that the accuracy of model based on LSTM + Word2vec is much better than that of CNN. CNN demonstrates great advantages in image processing. However, the model has low results in text classification. It can be seen that LSTM gives good results in text data processing.

Besides, the received results are also compared with other methods results. The results are shown in Table 5, Figures 12 and 13.

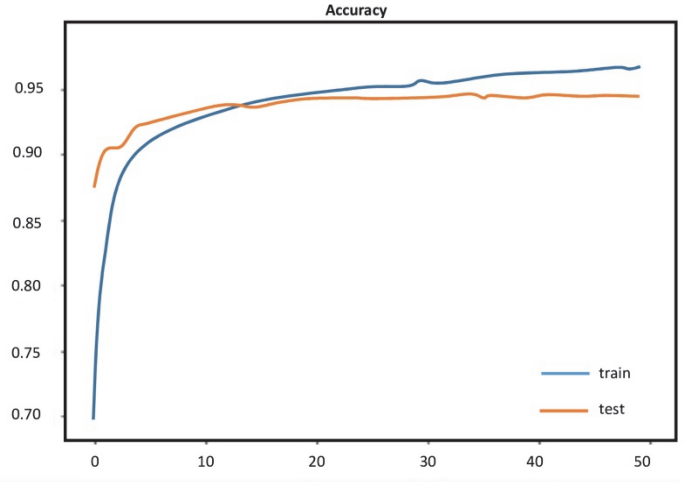

Fig. 12. Loss value of proposal's model

It could be seen in Table 5, that while using the same number of extracting features the proposed approach gains a better accuracy than the other methods.

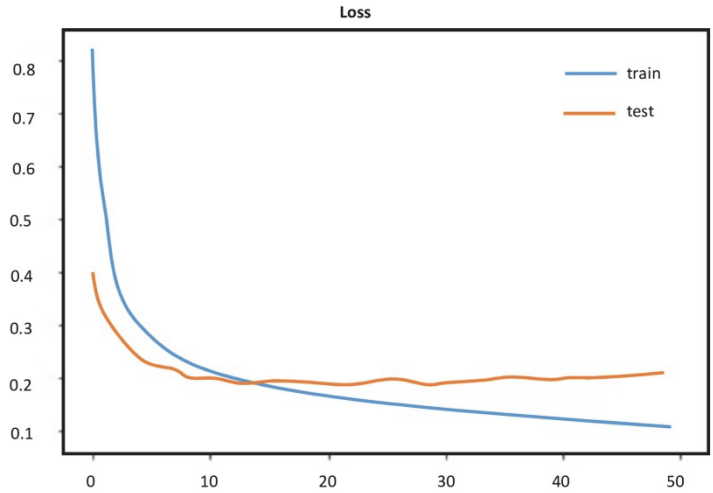

Fig. 13. Accuracy value of proposed model

Table 5. Comparing the results of proposed method with some other methods

Method	Technology	Data	Number of extracting features	Accuracy (%)
Proposed	LSTM, Wor2vec, and CNN	Vietnam data (VNTC) [4]	300	93.8
[18]	Topic modeling (Naïve Bayes theory)	VLSP (20000 sentences)	829 and 339	from 83.00 to 94.07
[19]	Support vector machine (SVM)	4162 documents [29]	7721	80.72
[45]	Naïve Bayes, Maxent, LSTM, and Bi-LSTM	Vietnamese students of feed- back corpus for sentiment analysis (16000 feed-back) [47]	300	from 81.2 to 89.6
[4]	SVM,kNN, and NGram	VNTC [4]	N/A	93.4; 84.67; and 97.1
[47]	SVM, Random forest, SVC, and neural network	VNTC [4]	N/A	96.52; 99.21; 99.22; and 99.75

5. Conclusion. The documents classification plays an important role in exploiting big data. The paper presented the method of text classification based on LSTM and CNN models. LSTM algorithm overcomes the vanishing gradient problem that helps in identifying the associations between sentence characteristics, key words, and contextual words. The proposed approach contribution is the use of the Word2vec model for classifying text based on title. Besides, the features learning was executed by Word2vec method that helps to link them together to improve the model efficiency. In the Word2vec method, was used a 300-dimension to reduce the number of characteristics in comparison with other methods. It helps to increase processing speed and avoid a curse of dimensionality.

Also, was performed the model without using Word2vec. The simulation (model) shows that the results are better with Word2vec use. The selection of characteristics from Word2vec helps the model to select appropriate features and to improve accuracy. Also were compared the models using LSTM and CNN. The results show that LSTM is better than CNN.

However, some disadvantages were identified as follows:

– training time of model is longer than that of other methods (SVM and CNN);

– the method will not work correctly if the sentences are too long.

In the future, therefore, the plan is to improve the classification model by combining other DL models to change derivative disappearance.

References

1. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural computation.* 1997. vol. 9. pp. 1735–1780.
2. Sak H., Senior A., Beaufays F. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv preprint arXiv:1402.1128.2014.
3. Phuong L.-H., Nguyen H., Roussanaly A., Ho T. A hybrid approach to word segmentation of Vietnamese texts. *Lecture Notes in Computer Science.* 2013. vol. 5196. pp. 240–249.
4. Hoang V.C.D., Dinh D., Nguyen N. le, Ngo H.Q. A comparative study on Vietnamese text classification methods. 2007 IEEE International Conference on Research, Innovation and Vision for the Future. 2007. pp. 267–273.
5. Ngo Q.H., Dien D., Winiwarter W. A hybrid method for word segmentation with English-Vietnamese bilingual text. 2013 International Conference on Control, Automation and Information Sciences (ICCAIS). 2013. pp. 48–52.
6. Jindal P., Jindal B. Line and word segmentation of handwritten text documents written in Gurmukhi script using mid point detection technique. 2015 2nd International Conference on Recent Advances in Engineering Computational Sciences (RAECS). 2015. pp. 1–6.
7. Gao Y. et al. Wacnet: Word segmentation guided characters aggregation net for scene text spotting with arbitrary shapes. 2019 IEEE International Conference on Image Processing (ICIP). 2019. pp. 3382–3386.
8. Charoenpornsawat P., Schultz T. Improving word segmentation for Thai speech translation. 2008 IEEE Spoken Language Technology Workshop. 2008. pp. 241–244.
9. Yu C. et al. Term extraction from Chinese texts without word segmentation. 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT). 2017. pp. 1–4.

10. Nguyen T., Le A. A hybrid approach to Vietnamese word segmentation. 2016 IEEE RIVF International Conference on Computing Communication Technologies, Research, Innovation, and Vision for the Future (RIVF). 2016. pp. 114–119.
11. Zhang Z. et al. Effective subword segmentation for text comprehension. *IEEE/ACM Transactions on Audio, Speech, and Language Processing.* 2019. vol. 27. no. 11. pp. 1664–1674.
12. Bal A., Saha R. An improved method for handwritten document analysis using segmentation, baseline recognition and writing pressure detection. *Procedia Computer Science*. 2016. vol. 93. pp. 403–415.
13. Nguyen T.V., Tran H.K., Nguyen T.T.T., Nguyen H. Word segmentation for Vietnamese text categorization: An online corpus approach. RIVF06. 2005. vol. 172. pp. 1–6.
14. Nguyen T., Lung V.D. Extracting the main content of Vietnamese scientific documents based on the structure. *Vietnam Journal of Science and Technology (VJST)*. 2014. vol. 52. no. 3. pp. 269–280.
15. Xiao L., Wang G., Zuo Y. Research on patent text classification based on word2vec and LSTM. 2018 11th International Symposium on Computational Intelligence and Design (ISCID). 2018. vol. 01. pp. 71–74.
16. Hassan A., Mahmood A. Efficient deep learning model for text classification based on recurrent and convolutional layers. 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). 2017. pp. 1108–1113.
17. Sarkar A., Chatterjee S., Das W., Datta D. Text classification using support vector machine. International *Journal of Engineering Science Invention.* 2015. vol. 4. no. 11. pp. 33–37.
18. Linh B.K. et al. Vietnamese text classification based on topic modeling. 9th Fundamental and Applied IT Research (FAIR). 2016. vol. 01. pp. 532–537.
19. De T.C., Khang P.N. Classify text with supported vector learning machine and decision tree. *Can Tho University Journal of Science*. 2012. vol. 21. pp. 269–280.
20. Radhika K., Bindu K.R. A text classification model using convolution neural network and recurrent neural network. *International Journal of Pure and Applied Mathematics*. 2018. vol. 119. pp. 1549–1554.
21. Fischer T., Krauss C. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*. 2018. vol. 270. no. 2. pp. 654–669.
22. Sebastiani F. Machine learning in automated text categorization. *ACM Computing Surveys*. 2001. vol. 34. pp. 1–47.
23. Yasotha R., Charles E.Y.A. Automated text document categorization. 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS). 2015. pp. 522–528.
24. Farhoodi M., Yari A. Applying machine learning algorithms for automatic Persian text classification. 2010 6th International Conference on Advanced Information Management and Service (IMS). 2010. pp. 318–323.
25. Krendzelak M., Jakab F. Text categorization with machine learning and hierarchical structures. 2015 13th International Conference on Emerging eLearning Technologies and Applications (ICETA). 2015. pp. 1–5.
26. Giang N.L., Hien N.M. Classification of Vietnamese documents using support vector machine. *NU Journal of Science: Computer Science and Communication Engineering.* 2005. pp. 1–10.
27. Nguyen P., Hong T., Nguyen K., Nguyen N. Deep learning versus traditional classifiers on Vietnamese students' feedback corpus. 2018 5th NAFOSTED Conference on Information and Computer Science (NICS). 2018. pp. 75–80.
28. Vo Q., Nguyen H., Le B., Nguyen M. Multi-channel LSTM-CNN model for Vietnamese sentiment analysis. 9th International Conference on Knowledge and Systems Engineering (KSE). 2017. pp. 24–29.
29. Vnexpress, The most read Vietnamese newspaper. 2020. Available at: https://e.vnexpress.net/ (accessed: 05.12.2019).
30. Tuoitre, Tuoitre news. 2020 Available at: https://tuoitre.vn/ (accessed: 05.12.2019).

31. Thanhnien, Thanhnien online newspaper. 2020. Available at: https://thanhnien.vn/a (accessed: 05.12.2019).
32. NLD, Nguoilaodong online newspaper. 2020. Available at: https://nld.com.vn/ (accessed: 05.12.2019).
33. Trung T.V. Python Vietnamese Core NLP Toolkit. 2019. Available at: https://github.com/trungtv/pyvi (accessed: 05.12.2019).
34. Nguyen D.Q. et al. A fast and accurate Vietnamese word segmenter. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). 2018. pp. 2582–2587.
35. Nguyen D.Q., Verspoor K. An improved neural network model for joint pos tagging and dependency parsing. Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. 2018. pp. 1–11.
36. Nguyen C.-T. et al. Vietnamese word segmentation with CRFs and SVMs: An investigation. Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation 2006. pp. 215–222.
37. Le V.-D. Detailed explanation of Word2Vector Skip-gram. 2015. Available at: http://www.programmersought.com/article/8383114826/ (accessed: 05.12.2019).
38. Ma L., Zhang Y. Using word2vec to process big text data. 2015 IEEE International Conference on Big Data (Big Data). 2015. pp. 2895–2897.
39. Barazza L. How does Word2Vec's Skip-Gram work? 2017. Available at: https://becominghuman.ai (accessed: 19.02.2017).
40. Landthaler J. et al. Extending thesauri using word embeddings and the intersection method. ASAIL@ ICAIL. 2017. vol. 8. no. 1. pp. 112–119.
41. An S. Recurrent Neural Networks. 2017. Available at: https://www.cc.gatech.edu/ san37/post/dlhc-rnn/ (accessed: 10.10.2019).
42. Zhang Y., Wallace B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification // arXiv preprint arXiv:1510.03820. 2015.
43. Le V.-D. Vietnamese stopwords, 2015. Available at: https://github.com/stopwords/vietnamese-stopwords (accessed: 05.12.2019).
44. Ting K.M. Confusion Matrix. Boston. MA: Springer US. 2010. pp. 209–209.
45. Nguyen P., Hong T., Nguyen K., Nguyen N. Deep learning versus traditional classifiers on Vietnamese students' feedback corpus. 2018 5th NAFOSTED Conference on Information and Computer Science (NICS). 2018. pp. 75–80.
46. Nguyen K.V. et al. UIT-VSFC: Vietnamese students' feedback corpus for sentiment analysis. 2018 10th International Conference on Knowledge and Systems Engineering (KSE). 2018. pp. 19–24.
47. Van T.P., Thanh T.M. Vietnamese news classification based on bow with key-words extraction and neural network. 2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES). 2017. pp. 43–48.

Phat Huu Nguyen – Ph.D., Dr.Sci., Lecturer, School of Electronics and Telecommunications, Hanoi University of Science and Technology (HUST). Research interests: digital image and video processing, wireless networks, ad hoc and sensor network, and intelligent traffic system (ITS) and internet of things (IoT). The number of publications – 55. phat.nguyenhuu@hust.edu.vn; 1, Dai Co Viet str., Hanoi, Viet Nam; office phone: +84(243)869-2242; fax: +84(243)869-2242.

Anh Nguyen Thi Minh – Bachelor, School of Electronics and Telecommunications, Hanoi University of Science and Technology (HUST). Research interests: natural language processing, artificial intelligence applications. The number of publications – 1. anh.ntm165774@sis.hust.edu.vn; 1, Dai Co Viet str., Hanoi, Viet Nam; office phone: +84(243)869-2242; fax: +84(243)869-2242.

Acknowledgements. This research is carried out in the framework of the project funded by the Ministry of Education and Training (MOET), Vietnam under the grant B2020-BKA-06. The authors would like to thank the MOET for their financial support.

УДК 004.9 **DOI** 10.15622/ia.2020.19.6.5

Х.Н. Фат, Н.Т.М. Ань

АЛГОРИТМ КЛАССИФИКАЦИИ ВЬЕТНАМСКОГО ТЕКСТА С ИСПОЛЬЗОВАНИЕМ ДОЛГОЙ КРАТКОСРОЧНОЙ ПАМЯТИ И WORD2VEC

Фат Х.Н., Ань Н.Т.М. **Алгоритм классификации вьетнамского текста с использованием долгой краткосрочной памяти и Word2Vec.**

Аннотация. В условиях текущей четвертой промышленной революции вместе с развитием компьютерных технологий увеличивается и количество текстовых данных. Следует понимать природу и характеристики этих данных, чтобы применять необходимые методологии. Автоматическая обработка текста экономит время и ресурсы существующих систем. Классификация текста является одним из основных приложений обработки естественного языка с использованием таких методов, как анализ тональности текста, разметка данных и так далее. В частности, недавние достижения в области глубокого обучения показывают, что эти методы хорошо подходят для классификации документов. Они продемонстрировали свою эффективность в классификации англоязычных текстов. Однако по проблеме классификации вьетнамских текстов существует не так много исследований. Последние созданные модели глубокого обучения для классификации вьетнамского текста показали заметные улучшения, но тем не менее этого недостаточно. Предлагается автоматическая система на основе длинной краткосрочной памяти и Word2Vec моделей, которая повышает точность классификации текстов. Предлагаемая модель продемонстрировала более высокие результаты классификации вьетнамских текстов по сравнению с другими традиционными методами. При оценке данных вьетнамского текста предлагаемая модель показывает точность классификации более 90%, поэтому может быть использована в реальном приложении.

Ключевые слова: классификация текста, естественная языковая обработка, обработка данных, длинная краткосрочная память, Word2Vec

Фат Хуу Нгуен – д-р техн. наук, преподаватель, факультет электроники и телекоммуникаций, Ханойский научно-технический университет. Область научных интересов: цифровая обработка изображений и видео, беспроводные сети, одноранговые и сенсорные сети, интеллектуальная транспортная сеть (ITS) и Интернет вещей (IoT). Число научных публикаций – 55. phat.nguyenhuu@hust.edu.vn; Дай Ко Вьет, 1, Ханой, Вьетнам; р.т.: +84(243)869-2242; факс: +84(243)869-2242.

Ань Нгуен Тхи Минь – бакалавр, факультет электроники и телекоммуникаций, Ханойский научно-технический университет. Область научных интересов: обработка естественного языка, приложения искусственного интеллекта. Число научных публикаций – 1. anh.ntm165774@sis.hust.edu.vn; ул. Дай Ко Вьет, 1, Ханой, Вьетнам; р.т.: +84(243)869-2242; факс: +84(243)869-2242.

Поддержка исследований. Данное исследование проводится в рамках проекта, финансируемого Министерством образования и науки Вьетнама в рамках (грант B2020-BKA-06).

Литература

1. *Hochreiter S., Schmidhuber J.* Long short-term memory // Neural computation. 1997. vol. 9. pp. 1735–1780.

2. *Sak H., Senior A., Beaufays F.* Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition // arXiv preprint arXiv:1402.1128.2014.
3. *Phuong L.-H., Nguyen H., Roussanaly A., Ho T.* A hybrid approach to word segmentation of vietnamese texts // Lecture Notes in Computer Science. 2013. vol. 5196. pp. 240–249.
4. *Hoang V.C.D., Dinh D., Nguyen N. le, Ngo H.Q.* A comparative study on Vietnamese text classification methods // 2007 IEEE International Conference on Research, Innovation and Vision for the Future. 2007. pp. 267–273.
5. *Ngo Q.H., Dien D., Winiwarter W.* A hybrid method for word segmentation with english- vietnamese bilingual text // 2013 International Conference on Control, Automation and Information Sciences (ICCAIS). 2013. pp. 48–52.
6. *Jindal P., Jindal B.* Line and word segmentation of handwritten text documents written in Gurmukhi script using mid point detection technique // 2015 2nd International Con- ference on Recent Advances in Engineering Computational Sciences (RAECS). 2015. pp. 1–6.
7. *Gao Y. et al.* Wacnet: Word segmentation guided characters aggregation net for scene text spotting with arbitrary shapes // 2019 IEEE International Conference on Image Processing (ICIP). 2019. pp. 3382–3386.
8. *Charoenpornsawat P., Schultz T.* Improving word segmentation for Thai speech translation // 2008 IEEE Spoken Language Technology Workshop. 2008. pp. 241–244.
9. *Yu C. et al.* Term extraction from Chinese texts without word segmentation // 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT). 2017. pp. 1–4.
10. *Nguyen T., Le A.* A hybrid approach to Vietnamese word segmentation // 2016 IEEE RIVF International Conference on Computing Communication Technologies, Research, Innovation, and Vision for the Future (RIVF). 2016. pp. 114–119.
11. *Zhang Z. et al.* Effective subword segmentation for text comprehension // IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2019. vol. 27. no. 11. pp. 1664–1674.
12. *Bal A., Saha R.* An improved method for handwritten document analysis using segmentation, baseline recognition and writing pressure detection // Procedia Computer Science. 2016. vol. 93. pp. 403–415.
13. *Nguyen T.V., Tran H.K., Nguyen T.T.T., Nguyen H.* Word segmentation for Vietnamese text categorization: An online corpus approach // RIVF06. 2005. vol. 172. pp. 1–6.
14. *Nguyen T., Lung V.D.* Extracting the main content of Vietnamese scientific documents based on the structure // Vietnam Journal of Science and Technology (VJST). 2014. vol. 52. no. 3. pp. 269–280.
15. *Xiao L., Wang G., Zuo Y.* Research on patent text classification based on Word2Vec and LSTM // 2018 11th International Symposium on Computational Intelligence and Design (ISCID). 2018. vol. 01. pp. 71–74.
16. *Hassan A., Mahmood A.* Efficient deep learning model for text classification based on recurrent and convolutional layers // 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). 2017. pp. 1108–1113.
17. *Sarkar A., Chatterjee S., Das W., Datta D.* Text classification using support vector machine // International Journal of Engineering Science Invention. 2015. vol. 4. no. 11. pp. 33–37.
18. *Linh B.K. et al.* Vietnamese text classification based on topic modeling // 9th Fundamental and Applied IT Research (FAIR). 2016. vol. 01. pp. 532–537.
19. *De T.C., Khang P.N.* Classify text with supported vector learning machine and decision tree // Can Tho University Journal of Science. 2012. vol. 21. no. a. pp. 269–280.

20. *Radhika K., Bindu K.R.* A text classification model using convolution neural network and recurrent neural network // International Journal of Pure and Applied Mathematics. 2018. vol. 119. pp. 1549–1554.
21. *Fischer T., Krauss C.* Deep learning with long short-term memory networks for financial market predictions // European Journal of Operational Research. 2018. vol. 270. no. 2. pp. 654–669.
22. *Sebastiani F.* Machine learning in automated text categorization // ACM Computing Surveys. 2001. vol. 34. pp. 1–47.
23. *Yasotha R., Charles E.Y.A.* Automated text document categorization // 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICI-CIS). 2015. pp. 522–528.
24. *Farhoodi M., Yari A.* Applying machine learning algorithms for automatic Persian text classification // 2010 6th International Conference on Advanced In-formation Manage- ment and Service (IMS). 2010. pp. 318–323.
25. *Krendzelak M., Jakab F.* Text categorization with machine learning and hierarchical structures // 2015 13th International Conference on Emerging eLearning Technologies and Applications (ICETA). 2015. pp. 1–5.
26. *Giang N.L., Hien N.M.* Classification of Vietnamese documents using support vector machine // VNU Journal of Science: Computer Science and Communication Engineering. 2005. pp. 1–10.
27. *Nguyen P., Hong T., Nguyen K., Nguyen N.* Deep learning versus traditional classifiers on Vietnamese students' feedback corpus // 2018 5th NAFOSTED Conference on Information and Computer Science (NICS). 2018. pp. 75–80.
28. *Vo Q., Nguyen H., Le B., Nguyen M.* Multi-channel LSTM-CNN model for Vietnamese sentiment analysis // 9th International Conference on Knowledge and Systems Engineering (KSE). 2017. pp. 24–29.
29. Vnexpress, The most read Vietnamese newspaper. 2020. URL: https://e.vnexpress.net/ (дата обращения: 05.12.2019).
30. Tuoitre, Tuoitre news. 2020. URL: https://tuoitre.vn/ (дата обращения: 05.12.2019).
31. Thanhnien, Thanhnien online newspaper. 2020. URL: https://thanhnien.vn/a (дата обращения: 05.12.2019).
32. NLD, Nguoilaodong online newspaper. 2020. URL: https://nld.com.vn/ (дата обращения: 05.12.2019).
33. *Trung T.V.* Python Vietnamese Core NLP Toolkit. 2019. URL: https://github.com/trungtv/pyvi (дата обращения: 05.12.2019).
34. *Nguyen D.Q. et al.* A fast and accurate Vietnamese word segmenter // Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). 2018. pp. 2582–2587.
35. *Nguyen D.Q., Verspoor K.* An improved neural network model for joint post tagging and dependency parsing // Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. 2018. pp. 1–11.
36. *Nguyen C.-T. et al.* Vietnamese word segmentation with CRFs and SVMs: An investigation // Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation 2006. pp. 215–222.
37. *Le V.-D.* Detailed explanation of Word2Vector Skip-gram. 2015. URL: http://www.programmersought.com/article/8383114826/ (дата обращения: 05.12.2019).
38. *Ma L., Zhang Y.* Using word2vec to process big text data // 2015 IEEE International Conference on Big Data (Big Data). 2015. pp. 2895–2897.
39. *Barazza L.* How does Word2Vec's Skip-Gram work? 2017. URL: https://becominghuman.ai (дата обращения: 19.02.2017).
40. *Landthaler J. et al.* Extending thesauri using word embedding's and the inter-section method // ASAIL@ ICAIL. 2017. vol. 8. no. 1. pp. 112–119.

41. *An S.* Recurrent Neural Networks. 2017. URL: https://www.cc.gatech.edu/san37/post/dlhc-rnn/ (дата обращения: 10.10.2019).
42. *Zhang Y., Wallace B.* A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification // arXiv preprint arXiv:1510.03820. 2015.
43. *Le V.-D.* Vietnamese stopwords, 2015. URL: https://github.com/stopwords/vietnamese- stopwords (дата обращения: 05.12.2019).
44. *Ting K.M.* Confusion Matrix. Boston // MA: Springer US. 2010. pp. 209–209.
45. *Nguyen P., Hong T., Nguyen K., Nguyen N.* Deep learning versus traditional classifiers on Vietnamese students' feedback corpus // 2018 5th NAFOSTED Conference on Information and Computer Science (NICS). 2018. pp. 75–80.
46. *Nguyen K.V. et al.* UIT-VSFC: Vietnamese students' feedback corpus for sentiment analysis // 2018 10th International Conference on Knowledge and Systems Engineering (KSE). 2018. pp. 19–24.
47. *Van T.P., Thanh T.M.* Vietnamese news classification based on bow with key-words extraction and neural network // 2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES). 2017. pp. 43–48.