Vietnamese Document Classification Using Graph Convolutional Network

Huy-The Vu, Van-Hau Nguyen, Van-Quyet Nguyen, Minh-Tien Nguyen
Faculty of Information Technology
Hung Yen University of Technology and Education
Hung Yen, Vietnam
{thevh, haunv, quyetict, tiennm}@utehy.edu.vn

***Abstract*—Document classification is a classical and fundamental text mining problem for many applications. In such classifiers, text representation is an intermediate step, but it plays an important role in building the models. Recently, graph neural networks have been shown to be a potential method for text presentation since they not only have a rich relational structure but also preserve global word co-occurrence correlation. However, most of them have been proposed for English documents. In this paper, we present a model based on a graph convolutional network for Vietnamese document classification. We first present detailed steps in building the graph from Vietnamese documents and a two-layer GCN architecture for graph embedding. We then propose a method, named PMI filter, to improve the classification accuracy of the model. Furthermore, aspects of the proposed model are also investigated to provide a better understanding of the model behavior. The proposed work is evaluated on two large Vietnamese datasets. In experiments, the proposed model archives better results than its baseline and competitive performance compared to existing feature-selection based methods.**

***Index Terms*—graph neural network, graph convolutional network, document classification**

I. Introduction

Text classification is an essential task of natural language processing (NLP), in which a classifier assigns an input document into pre-defined classes. The main approach to the task is to use hand-crafted features, such as bag-of-words and n-grams, for training a classification model [1], [2]. The outputs of text classification can be applied to numerous NLP applications such as document organization, news filtering, spam detection, opinion mining, and summarization [3]–[6]. It, therefore, demands high-quality text classification systems.

The important intermediate step of text classification is the representation of text. This step converts input text into hidden representation that machine learning models can operate. Over the decades, many studies have investigated this step to improve the quality of the classification [1], [2], [6]–[11]. The recent success of deep learning provides a new way for data representation. Instead of using hand-crafted features, deep learning models can automatically learn hidden features from data by using several architectures such as convolutional neural networks (CNN) [6], long-short term memory (LSTM) [12], or transformers such as BERT [13]. The extension of deep learning models for text classification is to learn data representation on graphs [7], [8], [14]–[16]. This extension applies convolutional neural networks on graphs (called graph convolutional networks - GCN) with the assumption that the structure of graphs provides rich information for learning data representation.

This research was supported by Hung Yen University of Technology and Education, under the grant number UTEHY.L.2020.08

While text classification has received a lot of attention for English, we argue that the investigation of this task for Vietnamese is still an early stage with a few studies [17]–[20]. This makes a boundary for understanding the behavior of machine learning models in a low-resource language. This paper bridges the gap of adapting deep learning models for classifying Vietnamese text. To do that, we introduce a classification model based on GCN [16]. The intuition behind our model is that we refine the graph creation step by filtering the weight of word-word edges of heterogeneous graphs (documents and words). After building the graphs, the model learns to create document embeddings used for classification. This paper makes two main contributions:

- It adopts the model of graph convolutional network based on heterogeneous graphs for Vietnamese document classification. We first present detailed steps in building the graph from Vietnamese documents and a two-layer GCN architecture for graph embedding. We then propose a method, named PMI filter, to improve classification accuracy. To the best of our knowledge, we are the first to apply graph convolutional networks based on heterogeneous graphs for Vietnamese documents.
- It investigates different aspects of the model. The investigation provides a better understanding of the behavior of the model that uses graph convolutional networks on heterogeneous graphs for Vietnamese documents, as well as suggestions for tuning to improve the performance of the model.

The proposed model is evaluated on two large Vietnamese datasets. The evaluation results show that our model improves classification accuracy compared to the baseline and has competitive results compared to other feature-selection based methods.

II. Related Work

Document classification is a traditional natural language processing (NLP) task, which has well investigated with many

studies [1], [2], [15]–[17], [19]. Traditional text/document classification usually uses feature engineering with human involvement. For feature engineering, bag-of-words and TF-IDF (term frequency - inverse document frequency) are well-known indicators. Some refined features were also designed such as using n-grams features for classification [1] or using entities in ontologies [2]. Some studies introduce the way to converting text to graphs and extracting features on graphs [7], [8].

The recent success of deep learning boosts the performance of document classification. Instead of using feature engineering, several methods achieve high accuracy by using hidden features from word embeddings [9], [10]. The advantage of these methods is to automatically learn hidden features from data. Besides the investigation of features, another research direction is to employ deep neural architecture for classification such as CNN for sentence classification [6] or LSTM for multi-task learning [11]. Based on the CNN architecture, several methods were developed for learning on graphs. For example, Kipf and Welling introduced a semi-supervised classification model with graph convolutional networks (GCN) [15]. The model scales linearly when increasing the number of edges and hidden layers while achieves state-of-the-art results on benchmark datasets.

There are a few studies for Vietnamese document classification. Hoang et al. introduced a comparative study for Vietnamese text classification [17]. The authors used two types of features bag-of-words and N-grams. For comparison, SVM, K-NN (K-Nearest Neighbour), and statistical N-grams language model for classification. Experimental results show that the classification can achieve 95% accuracy on 14,000 documents. Hai et al. showed a hybrid feature selection method for classifying Vietnamese text [19]. The authors investigated three feature selection methods: Chi-square, information gain, and document frequency. Based on that, they combined Chi-square and information gain as the feature selection model. Experimental results show that the proposed method helps to improve the accuracy of classification. A recent work introduced a classification model based on neural networks [20]. The model learns features from bag-of-words and keyword extraction and uses a feed-forward network for classification. Experimental results show that the model achieves better accuracy than SVM and random forest.

The work of Yao et al. is perhaps the most relevant to our study [16]. The authors introduced a model for document classification based on graphs by defining a heterogeneous graph that contains documents and words. Edges between documents and words were created by using similarity metrics. The learning process used graph convolutional networks (GCN). We share the idea of using GCN on the heterogeneous graph; however, we extend the model in two points. Firstly, we filter weak word-word edges by adding a low positive threshold. This helps label information of document vertexes are propagated better, as well as improve classification accuracy. Secondly, we dig deeply into several aspects, which provide a better understanding of the model behavior, and suggestions for tuning GCN based models.

III. GCN model for Vietnamese Document Classification

This section presents the proposed model for Vietnamese document classification. As described in Fig. 1, the model composes of two main parts: graph representing and graph convolutional network.

A. Graph Representing

Input documents are pre-processed, including cleaning, tokenizing, and removing stopwords. In experiments, we found that the pre-processing steps enable reducing the size of the graph and improving the performance of the model as well. This comes from using documents and unique words to build vertexes of the graph.

Here, we consider a graph $G = (V, E)$, where V and E are vertexes and edges of the graph G, respectively. V includes documents and unique words, $|V| = n_{docs} + n_{uni.words}$. For the edges of the graph, only ones between document to word and word to word are used. This comes from the suggestion [15] that GCN can propagate information of document labels to the entire graph well, so we do not need to consider document-document edges.

The weight of the document-word edge is represented by using is the term frequency-inverse document frequency (TF-IDF), while point-wise mutual information (PMI) is used to preserve global word co-occurrence information. In [16], the authors found that TF-IDF and PMI are better than TF and word co-occurrence count, respectively. Consequently, the graph G is represented by an adjacency matrix $A \in \mathbb{R}^{n \times n}$ ($n = |V|$), with a element $a_{i,j}$ is defined as (1)

$$a_{i,j} = \begin{cases} TF\text{-}IDF_{i,j} & i \text{ is document, } j \text{ is word} \\ PMI(i,j), & i,j \text{ are words} \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

It should be mentioned that $a_{i,j} = 1$ when $i = j$, because we need to sum up all the feature vectors of all neighbors and itself for every vertex. This corresponds to that each vertex in the graph has its own self-loop. To calculate the PMI value, a fixed size window is slid over the whole corpus to determine the global word co-occurrence, as expressed in (2), (3), (4)

$$PMI(i,j) = \log \frac{p(i,j)}{p(i)p(j)} \quad (2)$$

$$p(i,j) = \frac{N_W(i,j)}{N_W} \quad (3)$$

$$p(i) = \frac{N_W(i)}{N_W} \quad (4)$$

where $N_W(i,j)$ is the number of sliding windows having a pair of word i, j, $N_W(i)$ is the number of sliding windows in a corpus having word i, and N_W is the total sliding windows in that corpus. If a pair of words has a positive PMI value, it means that they have a high semantic correlation.

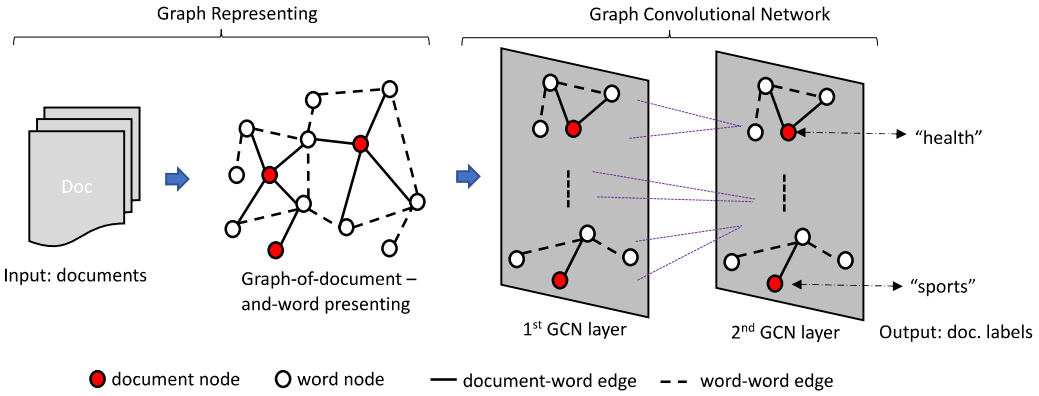

Fig. 1: The proposed model

B. Graph convolutional Network

A multi-layer GCN that is a kind of neural network directly working on a graph is introduced in [15]. The main idea of this model is that each layer captures information of immediate neighbors. When stacking multiple layers together, the model can integrate information from larger neighborhoods. Formally, the l^{th} layer is expressed as (5) and (6):

$$Z^{(l)} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^{(l-1)} W^{(l-1)} \tag{5}$$

$$H^{(l)} = f(Z^{(l)}) \tag{6}$$

where $Z^{(l)}$ is integrated information from neighbors, D is the degree matrix of A ($D_{ii} = \sum_i A_{ij}$), $H^{(l-1)} \in \mathbb{R}^{n \times d}$ is the output map of the previous layer (d is embedding size of the GCN layer), when $l = 0$ (the first layer), $H^{(0)} = X$ ($X \in \mathbb{R}^{n \times m}$ is feature matrix of vertexes, m is the dimension of the feature vectors), $W^{(l-1)}$ is the weight matrix, f is the activation function, e.g. ReLU. An explanation of the GCN layer is described in Fig. 2

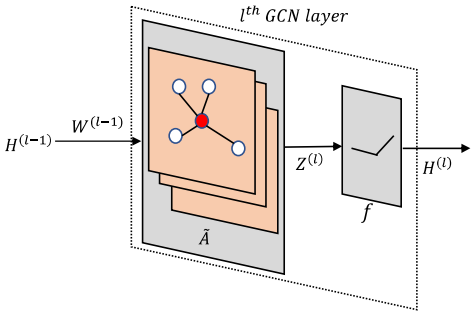

Fig. 2: An explanation of the l^{th} graph convolutional layer

In our model, we use a two-layer GCN, as shown in Fig. 1. This means that the model can propagate the information from a node to its two-hop neighborhood nodes. In our experiments with the Vietnamese datasets (described in Section IV), we found that the model with two layers achieved good results, while adding more layers did not improve the accuracy. This result is consistent with works in [15], [16]. It also means that the graph can present the text data well, with preserving the rich relational structure and global structure information.

C. PMI Filter

In this subsection, we propose a method to improve the classification accuracy of the proposed model. This method comes from an observation that word-word edges play an important role in propagating document label information to the entire graph. Therefore, it should be taken into consideration during the building graph to improve classification accuracy. In this method, we filter word-word edges that have a low PMI value by using a low positive threshold. Particularly, if the PMI value of a word-word edge is less than the threshold, the edge is not added to the graph. This enables label information of document vertexes to propagate to the whole graph well. The detailed computation of this method is described in Algorithm 1.

Algorithm 1: PMI filter pseudo-code

```
/* Input: PMI value of word-word edges PMI_{E(ww)} and
   a low positive threshold θ_th                          */
Input: PMI_{E(ww)}, θ_th
/* Output: word-word edge E^(ww) ∈ G                      */
Output: E^(ww)
/* Filtering calculation                                  */
E^(ww) = {}
foreach pmi_{e_i} ∈ PMI_{E(ww)} do
    if pmi_{e_i} > θ_th then
        E^(ww) += e_i
end
```

The inputs of this method are PMI values and a low positive threshold, while the output is strong edges that will be added to the graph. In this method, choosing the value of the low positive threshold is very important. A low threshold may add edges between word vertexes that are not very related. Consequently, label information of document vertexes may not be propagated to the entire graph. On the contrary, a high threshold may avoid important global word co-occurrence information. Selecting the threshold depends on input data, so a simple way to select an optimal value is to run the model under different thresholds, then choose the best case.

IV. Experiments

This section presents our experiments to evaluate the proposed model for the Vietnamese document classification. We also compared existing words:

- **Text-GCN** [16]: A text classifier using GCN. It is considered to be as a baseline of our model.
- **IG** [19]: Information Gain, a future selection method is commonly used as a criterion in machine learning. A feature with high information gain is a good one for classification.
- **CHI** [19]: based on Chi-Square χ^2 testing. CHI is used to measure the independence of a feature and a category.
- **DF** [19]: Document Frequency, a simple and effective feature selection method. It counts the number of documents in which a term occurs.
- **SIGCHI** [19]: A Hybrid Feature Selection Method of Chi-square and Information Gain. It is based on the combination of the Information Gain and Chi-square

We compare our work with the existing methods mentioned above because of some reasons. First, Text-GCN [16] is also based on GCN model [15] and achieved state-of-the-art classification results for English documents. Basically, our work is the most relevant to this study, so we consider it to be the baseline of our model. Second, the models based on feature selection methods of IG, CHI, DF, and SIGCHI [19] were also proposed to evaluate the same dataset (VNTC). This enables us to compare and explore the capability of graph convolutional embedding for Vietnamese documents.

A. Datasets

In our experiment, we evaluate the proposed model on the Vietnamese Text Classification (VNTC) dataset [17]. We carefully selected this dataset since it a relatively large and sufficient corpus compared to others. It includes about 100,000 documents that were collected and pre-processed from four popular online newspapers. It was divided into two levels:

- *VNTC-10*: contains 10 top categories with 33,759 documents for training and 50,373 documents for testing
- *VNTC-27*: includes 27 child topics of VNTC-10 with 14,375 documents for training and 12,076 documents for testing

The detailed information of the dataset is described in Tables I and II.

Before being fed into the proposed model, the documents are pre-processed. We firstly use regular expressions to clean text based on work in [6]. Since Vietnamese characters are slightly different from English, we changed some expressions to be suitable for Vietnamese. The text is then tokenized using CocCoctokenizer[1]. In English, words are separated by spaces, but Vietnamese words are more complicated [17]. Here, we use the underscore to concatenate morpho-syllables of a Vietnamese word together. We finally remove stopwords in the text by using a Vietnamese stopword list[2]. It should be mentioned that cleaning and removing stopwords are important steps because of graph complexity reduction as well as less required computation. In experiments, we found that the steps also improve the performance of the proposed method. The pre-processed results are summarized in Table III.

[1]https://github.com/coccoc/coccoc-tokenizer

[2]https://github.com/stopwords/vietnamese-stopwords

TABLE I: VNTC-10 with 10 classes contains top categories

No	Topic	Train	Test
1	politics-society	5,219	7,567
2	life	2,159	2,036
3	science & technology	1,820	2,096
4	business	2,552	5,276
5	health	3,384	5,417
6	law	3868	3788
7	world news	2,898	6,716
8	sports	5,298	6,667
9	culture	3,080	6,250
10	informatics	2,481	4,560
Sum		33,759	50,373

TABLE II: VNTC-27 with 27 classes contains child topics of VNTC-10

No	Topic	Train	Test
1	music	900	813
2	eating and drinking	265	400
3	real property	246	282
4	football	1,857	1,464
5	stock	382	320
6	bird flu - influenza	510	381
7	the life in the world	729	405
8	studying abroad	682	394
9	tourist	582	565
10	WTO	208	191
11	family	213	280
12	computer entertainment	825	707
13	education	821	707
14	sex	343	268
15	hackers and viruses	355	319
16	criminal	155	196
17	life space	134	58
18	international business	571	559
19	Beauty	776	735
20	lifestyle	223	214
21	shopping	187	84
22	fine arts	193	144
23	stage and screen	1,117	1,030
24	new computer products	770	595
25	tennis	588	283
26	young world	331	380
27	fashion	412	302
Sum		14,375	12,076

B. Experimental setup

We have taken settings to train and evaluate the proposed work as described in Section III in order to achieve good classification results. The graph convolutional network is trained with two layers since we found in preliminary experiments that increasing the number of GCN layers did not improve the classification accuracy. This is consistent with the conclusion in [15], [16]. The embedding size of the first GCN layer is set to 200, we found that increasing the size does not get better results while requiring more calculation as well as training time. The sizes of the sliding window used for

TABLE III: Statistics of the datasets

Dataset	# Docs	# Training	# Test	# Words	# Nodes	# Classes	Average Length
VNTC-10	84,132	33,759	50,373	54,155	138,287	10	242.99
VNTC-27	26,451	14,375	12,076	33,746	60,197	27	249.65

the PMI calculation are set to 20 and 30 for the VNTC-27 and VNTC-10 datasets, respectively. Other parameters for the training process such as learning rate, dropout rate are kept as [16], while the maximum of 200 epochs using Adam [21] are followed by [15]. For the baseline, settings for Text-GCN are kept the same as the original paper, while evaluation results of IG, CHI, DF, SIGCHI are summarized as reported in [19].

To measure text classification effectiveness, we evaluate the performance of the proposed model in terms of accuracy, precision, recall, f1-score [22].

C. Results

Table IV compares the evaluation results of the models in terms of accuracy on both dataset VNTC-10 and VNTC-27. As shown in the table, our model achieves comparative results compared to the others, especially for the larger dataset.

TABLE IV: Comparison of results in terms of accuracy (%)

Model	VNTC-10	VNTC-27
Text-GCN [16]	91.82	89.73
IG [19]*	91.02	87.93
CHI [19]*	90.94	87.24
DF [19]*	90.92	89.12
SIGCHI [19]*	91.25	**90.27**
Our work	**91.92**	90.07

* selecting the best results of the model (using 100% of features)

First, our model achieves the best result compared to the other ones for VNTC-10. It should be mentioned that this is a big dataset, about 3.16× in terms of the number of documents compared to the other. Here, both our work and Text-GCN are better than feature-selection based methods in [19]. This means that graph convolutional embedding is a good method and also suitable for Vietnamese classification tasks. The improvements come from three main reasons: (1) GCN can preserve relations of both documents-word and global word-word; (2) Through computation at each convolutional layer, each word nodes can gather label information of document nodes from its neighbors. This information is passed to other word nodes. In this way, the word nodes gradually form a key path to propagate label information of document nodes to entitle graphs. This makes the graph more separated and suitable for classification tasks; (3) Since the datasets contain long-text documents (see average length, as shown in Table III), this takes advantage of GCN as explained in [16].

On both datasets, our model performs better than Text-GCN. It should be noticed that both models use the same graphs built from input documents in each dataset (i. e. the same graph representing, see Fig. 1). There are two main reasons why we can achieve these results: (1) Our model takes advantage of the PMI filter method (see Section III-C). As mentioned before, word-word edges are bridged to propagate label information of document vertexes. Therefore, if the weight value of these edges is small, the paths will be weak, leading to degradation on the performance of the model. Especially, the number of word-word vertexes on both Vietnamese datasets is much higher than almost ones used in [16]. To find the threshold, we first calculate the weight average of all word-word edges having positive PMI value on the whole graph. We then run the proposed model under the variation of different thresholds. From there, we found that the threshold should be set around the average value, as shown in Fig. 3 (a); (2) we increase the size of the sliding window. This comes from that the size of the whole corpus is large. Therefore, a small window size could not capture enough global word-word information. In experiments, we found that increasing the window size did not improve classification accuracy when running on VNTC-27 because of its smaller corpus size. This is in contrast with VNTC-10, as shown in Fig. 3. The main reasons mentioned above also explained for experimental results as summarized in Table V, where we can get more insight performance of both models. As shown in the table, our model is better in almost all performance parameters when compared with the Text-GCN.

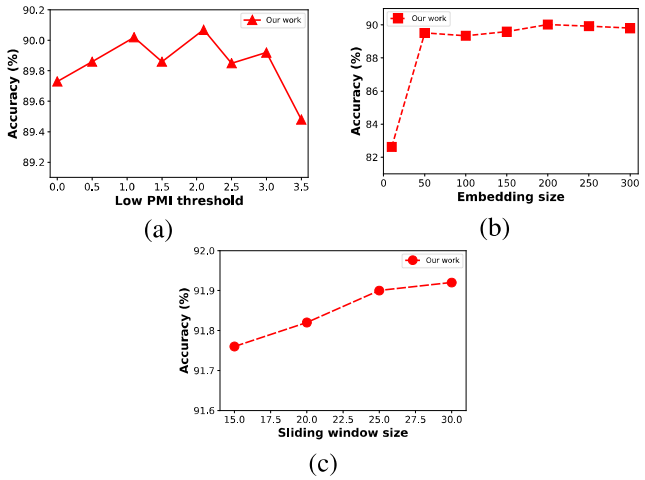

Fig. 3: Accuracy under variation of: (a) positive threshold on VNTC-27 (b) embedding size on VNTC-27 and (c) sliding window size on VNTC-10.

Fig. 3 describes the impacts of PMI threshold, embedding size, and sliding window size on the classification accuracy of the model. As shown in Fig. 3 (a), a small weight of word-word edge should be removed to improve performance. This trend of accuracy is consistent with the analysis of the proposed PMI filter method (see Section III-C). Fig. 3 (b) illustrated that a small embedding size of the first GCN layer

TABLE V: Comparison of results in terms of Micro-average and Macro-average (%)

Dataset	Model	Micro			Macro		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
VNTC-10	Text-GCN [16]	91.82	91.82	91.82	90.04	**89.76**	89.87
	Our work	**91.92**	**91.92**	**91.92**	**90.49**	89.50	**89.91**
VNTC-27	Text-GCN [16]	89.73	89.73	89.73	87.89	87.64	87.44
	Our work	**90.07**	**90.07**	**90.07**	**88.15**	**88.14**	**87.92**

could dramatically degrade accuracy. This is because label information of document vertexes could not propagate to the entire graph. In contrast, increasing embedding size did not improve accuracy while increasing the amount of computation as well as training time. For changing the size of the window, we found that it depends on the corpus size, with large corpus it should be increased and vice versa, as shown in Fig. 3 (c).

V. Conclusion

This paper has presented a GCN based classifier for Vietnamese document classification. We first describe steps for building the graph from Vietnamese documents and a two-layer GCN model. We then propose a method, named PMI filter, to improve classification accuracy. In addition, different aspects of the proposed model are also investigated to provide more insights into the model behavior. Perhaps this work is the first attempt to apply GCN on heterogeneous graphs for Vietnamese documents. In experiments, our model archives better results than its baseline and competitive performance compared to existing feature-selection based methods. In future work, we intend to work on methods to tackle the problem of the computational complexity that is linear in the number of graph edges. Besides, the performance improvement of the model is also considered.

Acknowledgment

This research was supported by Hung Yen University of Technology and Education, under the grant number UTEHY.L.2020.08

References

[1] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 90–94*, 2012.

[2] V. Chenthamarakshan, P. Melville, V. Sindhwani, and R. D. Lawrence, "Concept labeling: Building text classifiers with minimal supervision," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (AAAI), pp. 1225-1230*, 2011.

[3] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," *Mining Text Data, 163-222*, 2012.

[4] Z. Zeng, Y. Deng, X. Li, T. Naumann, and Y. Luo, "Natural language processing for ehr-based computational phenotyping," *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 16(1), 139-153*, 2018.

[5] M.-T. Nguyen, D.-V. Tran, C.-X. Tran, and M.-L. Nguyen, "Summarizing web documents using sequence labeling with user-generated content and third-party sources," in *International Conference on Applications of Natural Language to Information Systems, pp. 454-467*, 2018.

[6] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751.

[7] F. Rousseau, E. Kiagias, and M. Vazirgiannis, "Text categorization as a graph classification problem," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 1702–1712*, 2015.

[8] Y. Luo, Özlem Uzuner, and P. Szolovits, "Bridging semantics and syntax with graph algorithms—state-of-the-art of extracting biomedical relations," *Briefings in bioinformatics 18, no. 1: 160-178*, 2017.

[9] D. Shen, G. Wang, W. Wang, M. R. Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin, "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pp. 440–450*, 2018.

[10] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, "Joint embedding of words and labels for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pp. 2321–2331*, 2018.

[11] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), pp. 2873-2879*, 2016.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation, 9(8), pp. 1735-1780*, 1997.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171-4186*, 2019.

[14] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu, "Relational inductive biases, deep learning, and graph networks," in *arXiv preprint arXiv:1806.01261*, 2018.

[15] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proceedings of the 5th International Conference on Learning Representations*, ser. ICLR '17, 2017.

[16] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *n 33rd AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019, pp. 7370–7377.

[17] V. C. D. Hoang, D. Dinh, N. le Nguyen, and H. Q. Ngo, "A comparative study on vietnamese text classification methods," in *2007 IEEE International Conference on Research, Innovation and Vision for the Future*, 2007, pp. 267–273.

[18] G.-S. Nguyen, X. Gao, and P. Andreae, "Text categorization for vietnamese documents," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, vol. 3, pp. 466-469*, 2009.

[19] T. H. Nguyen, N. H. Nghia, D. L. Tuan, and V. T. Nguyen, "A hybrid feature selection method for vietnamese text classification," in *2015 Seventh International Conference on Knowledge and Systems Engineering (KSE)*, 2015, pp. 91–96.

[20] T. P. Van and T. M. Thanh, "Vietnamese news classification based on bow with keywords extraction and neural network," in *2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES), pp. 43-48*, 2017.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *In Interna- tional Conference on Learning Representations (ICLR)*, 2015.

[22] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, p. 1–47, Mar. 2002.