A TOPIC-DRIVEN GRAPH-OF-WORDS CONVOLUTIONAL NETWORK FOR IMPROVING TEXT CLASSIFICATION

Tham Vo
Thu Dau Mot University, Binh Duong, Vietnam

Abstract—In recent times, we have witnessed dramatic progresses and emergence of advanced deep neural architectures in natural language processing (NLP) domain. The advanced sequence-to-sequence (seq2seq)/transformer based architectures have demonstrated remarkable improvements in multiple NLP's tasks, including text categorization. But these advanced deep sequential text embedding techniques have still suffered limitations regarding with the capability of preserving the long-range dependencies between words and documents within the corpus level. Therefore, several graph neural network (GNN) based approaches for text classification task have been proposed recently to cope with this challenge. However, these GNN-based text classification techniques also encountered challenges of integrating global semantic information of texts. The global semantic information like as topic can support to facilitate the textual representation learning process for better classification-driven fine-tuning objective. To deal with these challenges, in this paper we proposed a novel topic-driven GNN-based text embedding approach, called as: GOWTopGCN. Our proposed model enables to simultaneously capture the rich global semantic information and long-range dependent relationships between words and documents. Extensive experiments in benchmark textual datasets show the outperformances of our proposed model in comparing with state-of-the-art transformer-based and GNN-based text classification baselines.

Keywords—neural topic modelling; BERT; GCN; graph-of-words.

I. INTRODUCTION

In general, text classification is considered as a primitive task for most of common problems in the NLP domain. For many years, text classification [1] [2] has been widely studied and applied in multiple real-world applications, like as: online news retrieval, recommendation, spam filtering, textual data filtering etc. The ultimate objective of text classification model is to assign proper label/class for text documents. In the past, most of the traditional textual classification techniques majorly relied on the hand-crafted textual feature engineering approaches, like as: bag-of-words (BOW), n-gram paradigm and their family. These hand-crafted techniques enable to find proper latent representation forms of texts. Thus, these hand-crafted latent representations can be explicitly understood by the computer. However, traditional hand-crafted feature engineering techniques normally encountered limitations [1] [2] [3] [4] related to the simplicity, sparse and low-quality in the achieved text representations. These limitations might lead to the downgrades in the overall accuracy performance of the text classification task. Moreover, the hand-crafted text feature engineering base techniques is also considered as high-cost approach. These methods require sufficient expert knowledge and human interactions to control the textual representation learning process. Thus, they might lack the capability of flexibly applying in the multi-linguistic context. In recent time, the tremendous raise of deep learning in multiple disciplines have shifted the textual feature engineering problem into a new level of automatic and rich contextual representation learning.

In general, the deep neural text embedding schema is originally come from a well-known model in NLP, which is: Word2Vec [5]. From that time on, there are several text embedding models have been introduced, like GloVe [6], fastText [7], Doc2Vec [8], etc. These word embedding models have showed their effectiveness in multiple NLP based applications. However, these recent word embedding based techniques still encountered limitations of preserving the sequential relationships between words to fulfill complex and rich semantic textual representation learning task. Thus, multiple deep neural architectures have been taken in consideration to cope with these limitations. Among advanced deep learning architectures, RNN is considered as the most popular one. The RNN-based text embedding techniques are majorly applied in recent studies to fulfill the sequential textual representation learning task. There are some notable RNN-based models, like as: (e.g. Tree-LSTM [9], RT-LSTM [10], MST-LSTM [11], etc.) have presented significant improvements in both textual embedding and classification problems. There are also outstanding works which demonstrated the usefulness of applying CNN (e.g., Dynamic-CNN [12], VDCNN [13], etc.) in textual representation learning as well as classification task. In this approach, the convolutional and pooling operations are utilized to effectively extract pattern-based textual features for multiple task-driven learning tasks.

Contact author: Tham Vo
Email: thamvth@tdmu.edu.vn
Manuscript received: 7/2021, revised: 11/2021, accepted: 12/2021.

A. Recent progresses in text representation for classification and existing challenges

In recent years, the tremendous raises of auto-encoding (a.k.a. seq2seq) [14], attention mechanism [15] [16] and transformer [17] [18] [19] have shown significant progresses in textual understanding as well as representation learning. Among advanced deep learning based architecture, transformer is considered as the most recent state-of-the-art approach for multi-tasked rich contextual text representation learning. There are remarkable transformer-based models like as: ELMo [17], GPT-2 [18] and BERT [19] which have effectively support to capture rich semantic information from texts in context of large-scale contextual corpora via the pre-training schema. BERT [19] is considered as the most powerful language model which contains multiple pre-training versions which are available for different task-driven analysis and training purposes.

For text classification, the original BERT implementation has successfully demonstrated extraordinary improvements in both rich semantic feature extraction and classification fine-tuning in which can be welly adopted in multi-linguistic implementation context. Recently, there are some notable variants of BERT, like as: VGCN-BERT [20] and BAE [21] have shown potentiality of applying language pre-training language model in text classification task. However, recent transformer-based techniques are also considered as unable to preserve the global semantic information as well as long-range relationships between words and documents within a given text corpus. Thus, they might be incapable to integrate the rich schematic information of text's structures into the classification task-oriented training process. There are several previous studies [22] [23] have presented the usefulness of applying graph-of-words (GOW) paradigm in textual representation and analysis. By modelling the relationships between words and documents in a given text corpus as graph-based structures, we can sufficiently preserve these long-range relationships as well as syntactical structures between words and documents.

B. Our motivations and contributions

Recently, graph neural network (GNN) (e.g., GraphSage [24], GCN [25], GAT [26], etc.) has emerged as a promising direction for dealing graph-structured representation learning problem. Taking the advantages of GNN-based representation learning mechanism, researchers have attempted to find better text graph based transformation and embedding strategy to jointly learn the global structural and rich contextual representations of a given text corpus.

In order to cope with aforementioned challenges, in this paper we proposed a novel topic-aware GCN-based embedding model for text classification, called as: GOWTopGCN. In general, our proposed GOWTopGCN model is a combination between the neural topic modelling (NTM) with GCN-based textual representation learning. Our model enables to simultaneously learn the global semantic information and rich structural representations of texts to efficiently leverage the performance of text classification task. First of all, to effectively learn the topic-word distributions over a given text corpus, we applied the previous NVDM [29] which is an variational auto-encoding (VAE) based topic modelling architecture. The NVDM algorithm use the neural network-based Gaussian evaluation with the VAE framework to effectively achieve the high-quality latent topic-word distributions.

Then, these topic-word distributions are fused with BERT [19] -based textual word embeddings with a custom fusion mechanism to produce the final rich semantic word representations. Finally, these rich semantic word embedding matrices are utilized to produce the completed representations for overall documents as well as facilitate the GCN-based textual representation learning upon different constructed text graphs. Finally, the achieved structural embeddings of text via multi-layered GCN architecture are fed into a full-connected layer to conduct the document classification. In general, our contributions in this paper can be summarized as four-folds, which are:

- First of all, we present an integrated topic-driven textual embedding by combining the NTM and pre-trained BERT model to produce a rich semantic representations of words in a given text corpus. Then, the learnt word embeddings are utilized to construct the completed representations of documents as well as assist the multi-typed text graph construction process.
- Next, we demonstrate the multi-typed text graph construction process in which a given text corpus is transformed and represented as different text graphs which are majorly inherited from previous works [4] [27]. These constructed text graphs present multi-typed relationships between words, documents and latent topics which are achieved previously from the neural topic modelling process.
- Then, these multi-typed constructed text graphs are fed into multi-layered GCN architecture to learn the representations of document nodes which are later used for the classification task-driven fine-tuning process at the end. Through the GCN-based propagation learning process with the NTM+BERT based embeddings as the initial node features, we can efficiently preserve both global semantic and structural representations of document nodes. These rich semantic textual representations are explicitly used to leverage the performance of text classification.
- Finally, we conducted extensive experiments in benchmark textual datasets like as: Ohsumed, IMDb, Yelp-2014, DBLP and arXiv to demonstrate the effectiveness of our proposed GOWTopGCN model in comparing with recent state-of-the-art text classification baselines.

In the after all, the left contents of our paper are organized into 4 sections. For the next section, we briefly present literature reviews about recent studies in text classification as well as discussing about the pros/cons of each method. Next, we formally present the methodology and implementation of our proposed GOWTopGCN model in the third section. In the fourth section, we show extensive experiments and comparative studies between our proposed GOWTopGCN model and other baselines. Finally, in the fifth section, we conclude about our achievements in this paper as well as highlight some potential directions for the future works.

II. RELATED WORKS

In recent years, the rapid developments of deep learning have led to significant progresses of text analysis and representation learning techniques. Multiple deep learning based textual embedding and task-driven training strategies have been proposed. These advanced methods have supported to overcome existing challenges of traditional hand-crafted featuring engineering approach.

A. Traditional deep learning based approach for text representation learning

There several studies which utilized different advanced deep neural architectures, such as convolutional neural network (CNN) and recurrent neural network (RNN) (e.g., GRU, LSTM, Bi-LSTM, etc.) to effectively capture rich semantic and sequential information of texts which are later used to fine-tune for different textual analysis tasks. Through experiments in multiple benchmark datasets, deep learning based text representation learning techniques have demonstrated remarkable improvements in multiple primitive tasks of NLP domain. Thus, deep neural textual representation learning has become the mainstream for most of recent studies. Among advanced deep neural architectures, RNN is considered as the most popular neural architecture which enables to learn the long-term sequential information of words within documents to facilitate multiple tasks in NLP. For the classification problem, there are several notable RNN-based methods like as the Tree-LSTM [9], RT-LSTM [10] and MST-LSTM [11] model. In the Tree-LSTM [9] model, Tai, K. S. et al. proposed a tree-based LSTM architecture in which neural cells are arranged following hierarchical tree-structured pattern which support to effectively learn the syntactical relationships between words in text. Similar to that, Zhu, X. et al. proposed the recursive tree-structured LSTM architecture in the RT-LSTM model [10] to jointly learn the syntactical and historical textual representations to facilitate multiple tasks in NLP. To efficiently optimize the RNN-based textual embedding process in context of time-series, Liu, P. et al. proposed a multiple time-scaled LSTM architecture [11]. The proposed MST-LSTM [11] enables to preserve the sequential information of a given text corpus in different timescales for leveraging performance of text classification in both accuracy and scalability aspects.

On the other side, the CNN based deep neural architecture is also widely attended by researchers in textual embedding and classification application. There are notable models such as Dynamic-CNN [12] and VDCNN [13] which utilized the convolutional operations to deeply learn the rich pattern-based features from texts through input word embedding matrices for documents. These CNN-based approaches also demonstrated significant enhancements in text classifications. However, traditional deep learning based textual embedding approach still suffered limitations related to the capabilities of capturing the rich contextual information of texts at different levels.

B. Seq2seq/transformer-based text representation learning approach

Recent emergence of transformer-based techniques like as: ELMo [17], GPT-2 [18] and BERT [19] have provided powerful tools for learning richer contextual information from text for improving the performance of text classification. Multiple variants of pre-training BERT-based models like as: VGCN-BERT [20] and BAE [21] have shown the usefulness of applying pre-training language model in both rich contextual information preserving from texts as well as facilitating the fine-tuning process for text categorization. Although these transformer-based model have proved the effectiveness of applying pre-training language schema on text classification, there are existing problems regarding with the ability of capturing long-range and global semantic information from texts. Text graph transformation and graph neural network representation learning are considered as potential directions for dealing with these challenges.

C. GNN-based text representation learning and classification

Recently, there are outstanding GNN textual embedding based techniques, like as: TextGCN [4], TensorGCN [28], and TG-Transformer [29]. In the TextGCN model [4], Yao, L. et al. proposed a novel multi-typed text graph transformation and GCN-based text graph embedding techniques to effectively capture the long-range structural information of texts which supports to remarkably improve the performance of text categorization in forms of document node classification task. The TextGCN model [4] is considered as the baseline for recent GNN-based textual embedding and classification like as: TensorGCN [28] and TG-Transformer [29]. However, these recent GNN-based models still encountered problems regarding with the capability on integrating with other rich global semantic information of texts like as topic. Thus, they might be still unable to achieve better fine-tuning results for the text classification problem.

III. GOWTOPGCN MODEL

In this section, we formally present the methodology of our proposed GOWTopGCN model which is a combination between the topic-aware textual embedding via NTM+BERT and text graph representation learning via GCN. First of all, we present an approach of integrating NTM with BERT to produce an unified latent topic and pre-trained BERT based word embeddings which carry rich global semantic and contextual information of a given text corpus. Then, these word embeddings are utilized in the process of producting completed document representations as well as contructing text graphs for a given corpus. There are four main types of text graphs which are utilized in our approach, which are word-word, word-document, word-topic and document-topic text graphs. Finally, these text graphs are unfied within a single heterorngeous network is fed into a multi-layered GCN archiecture to learn the text graph-structured node representations. The network node embeddings as the last hidden state outputs of a given GCN archiecture are later used in the training process for documet node classfication with a full-connected layer at the end.

A. Topic-driven text representation learning via NTM+BERT

NTM based topic-driven word embedding approach. First of all, our ultimate goal in the utilization of neural

topic modelling is to efficiently extract the latent topics distributions over words ($\mathcal{W}, w \in \mathcal{W}$) and documents in a given text corpus ($\mathcal{D}, d \in \mathcal{D}$). Mainly relying on the well-known LDA topic modelling paradigm, each document, denoted as: (d) has its own distributed latent topic proportion, denoted as: (z_d) and $z_d \in \mathbb{R}^{1\times K}$ with (K) is the number of predefined latent topics. On the other side, at the aspect of topic-word distributions, we have the topic assignments for unique words in a given text corpus, with: (t_w) presents for the topic assignment for a specific (w) word. To efficiently achieve model's parameters following the NTM-based schema, we mainly utilized the NVDM algorithm [27] which is considered as an variational auto-encoding (VAE)-based mechanism which contains two components, including the generative (a.k.a. encoding) network and inference (a.k.a. decoding) network. For the generative network as a neural encoding mechanism, it is designed to encode the input documents into the latent topic distributions. Then, these latent topic representations are passed through the inference based neural network component to reconstruct again the original input document.

Different from the traditional approach of LDA topic modelling approach which majorly depends on the mathematical inference process to estimate model's parameters, the NTM-based approach mainly applied the VAE-based architecture to effectively learn and parameterize the multinomial probabilistic distributions of latent topics. Thorough empirical studies [29] [30] in recent NTM-based techniques have shown the usefulness of applying VAE in topic modelling problem in which can produce better quality of latent topics distributions as well as be more scalable for large-scale text corpora. In general, the latent topic generative process for each input document (d) to produce the topic distribution proportion (z_d) can be formulated as the following (as shown in the equation 1):

$$\begin{aligned} \omega &\backsim \mathcal{N}(\mu_0, \sigma_0^2) \\ z_d &= \mathrm{softmax}(\mathrm{W}_\omega \omega + \mathrm{b}_\omega) \end{aligned} \tag{1}$$

$$\begin{aligned} t_w &\backsim \mathrm{Mult}(z_d) \\ w &\backsim \mathrm{Mult}(\beta_{t_w}) \end{aligned} \tag{2}$$

In the equation 1, the μ_0 and σ_0^2, are the mean and variance parameters of the Gaussian distribution, respectively. Then, the topic distribution proportion (z_d) is produced by parameterizing the prior distributions with a linear neural network layer (with W_ω and b_ω present for the trainable parameters of a given linear layer, respectively) and the softmax function. Then the topic-word distributions, ($\beta \in \mathbb{R}^{K\times|\mathcal{W}|}$) are achieved accordingly (as shown in the equation 2) [29] [30]. In general, the topic-word distributions are represented as an embedding matrix in which each unique word (w) in a given corpus ($\mathcal{D}$) is embedded as a latent topic based embedding vector, as: $\overrightarrow{w}^{\mathrm{NTM}} \in \mathbb{R}^{K\times 1}$.

Contextual word embedding approach via pre-trained BERT. Next, to achieve the contextual information of all unique words in a given corpus ($\mathcal{D}$), we mainly applied the pre-trained BERT model to produce the rich semantic word representations in context of large-scale text corpora. In general, the pre-trained BERT word embedding process can be formulated as the following: $\left\{\overrightarrow{w}_{\mathrm{i}}^{\mathrm{BERT}}\right\}_{\mathrm{i}=1}^{|\mathcal{W}|} = \mathrm{BERT}\left(\{w_{\mathrm{i}}\}_{\mathrm{i}=1}^{|\mathcal{W}|}\right)$. By doing this, we can achieve the BERT-based word embedding vector for each unique word, denoted as: ($\overrightarrow{w}^{\mathrm{BERT}}$) which carries rich contextual information at the corpus level. All BERT-based word embedding vectors form a completed word embedding matrix, denoted as: $\mathcal{X}^{w,\mathrm{BERT}} \in \mathbb{R}^{|\mathcal{W}|\times \mathrm{d}^{\mathrm{BERT}}}$, with ($\mathrm{d}^{\mathrm{BERT}}$) is the dimensionality of BERT-based output embedding vectors. Then, to produce the unified word embedding matrix, denoted as: $\mathcal{X}^{w} \in \mathbb{R}^{|\mathcal{W}|\times \mathrm{d}}$, with: d is the general word embedding output, which carry both global semantic and contextual information of a given text corpus, we defined a custom fusion mechanism as a full-connected neural layer. The general word embedding fusion mechanism can be formulated as the following (as shown in equation 3):

$$\begin{aligned} \mathcal{X}^{w} = \mathrm{W}_{\mathrm{NTM}}^{\mathrm{fuse}}.\,\mathrm{Linear}_{[\mathrm{d}]}(\beta^{\mathrm{T}}) \\ + \mathrm{W}_{\mathrm{BERT}}^{\mathrm{fuse}}.\,\mathrm{Linear}_{[\mathrm{d}]}(\mathcal{X}^{w,\mathrm{BERT}}) \\ + \mathrm{b}^{\mathrm{fuse}} \end{aligned} \tag{3}$$

In this equation, $\mathrm{W}_{\mathrm{NTM}}^{\mathrm{fuse}}$, $\mathrm{W}_{\mathrm{BERT}}^{\mathrm{fuse}}$ and $\mathrm{b}^{\mathrm{fuse}}$ is the trainable weighting and bias parameters which are jointly optimized with others during the training process. The $\mathrm{Linear}_{[\mathrm{d}]}(.)$ is a linear dense-based neural layer which is applied to transform the input embedding matrices (β) and ($\mathcal{X}^{w,\mathrm{BERT}}$) into the unified (d)-dimensional embedding spaces. At the end of this process, we can achieve final word embedding matrix of ($\mathcal{X}^{w}$) which carries global semantic and rich contextual information of a given corpus ($\mathcal{D}$).

Topic-driven document representation learning via Bi-LSTM. Next, to achieve the completed representations of all documents in ($\mathcal{D}$) from the achieved word embeddings ($\mathcal{X}^{w}$) in previous steps, we utilize a Bi-LSTM architecture to aggregate the sequential information of all (n) occurred words in forms of embedding vectors in ($\mathcal{X}^{w}, \overrightarrow{w} \in \mathcal{X}^{w}$), in each document ($d$), denoted as: $d = \{w_1, w_2, w_3, \dots, w_{\mathrm{n}}\}$. Then, the generated output hidden states of a given Bi-LSTM architecture is concatenate and linearly transformed to achieve a completed document embedding, denoted as: $\vec{d} \in \mathbb{R}^{1\times \mathrm{d}}$. These document embedding vectors form a completed document embedding matrix, denoted as: $\mathcal{X}^{d} \in \mathbb{R}^{|\mathcal{D}|\times \mathrm{d}}$. The general process of document representation learning procedure for a specific document (d), through Bi-LSTM can be formulated as the following (as shown in equation 4):

$$\begin{aligned} \mathcal{H}_d^{\mathrm{RNN},+} &= \mathrm{LSTM}(\{\overrightarrow{w_{\mathrm{i}}}\}_{\mathrm{i}=1}^{\mathrm{n}}, \Theta^{\mathrm{RNN},+}) \\ \mathcal{H}_d^{\mathrm{RNN},-} &= \mathrm{LSTM}(\{\overrightarrow{w_{\mathrm{i}}}\}_{\mathrm{i}=1}^{\mathrm{n}}, \Theta^{\mathrm{RNN},-}) \\ \vec{d} &= \mathrm{Linear}_{[\mathrm{d}]}\left([\mathcal{H}_d^{\mathrm{RNN},+}, \mathcal{H}_d^{\mathrm{RNN},-}]\right) \end{aligned} \tag{4}$$

In the equation 4, $\Theta^{\mathrm{RNN},+}$ and $\Theta^{\mathrm{RNN},-}$ are the trainable parameters of a given Bi-LSTM architecture in both directions and $[.,.]$ is the concatenation operation. Finally, to efficiently use the latent topic embeddings which are composed as a set of probabilistic distributions of observed words in a given corpus ($\mathcal{D}$), we also softly transformed the topic-word distributions (β) into a d-dimensional topic embedding matrix, denoted as the following: $\mathcal{X}^{t} = \mathrm{Linear}_{[\mathrm{d}]}(\beta)$. These transformed d-dimensional latent topic embedding matrix is later used for the text graph representation learning process via GCN which is described in the next section.

B. GCN-based text graph representation learning for classification

1) Multi-typed text graph construction

To present different global structural relationships between words, documents and (K) latent topics which are extracted previously by the NTM, we constructed four main types of text graphs, which are:

- **Word co-occurring text graph**. This text graph contains co-occurring relationships within documents in a given corpus between pairwise words. In general, the word-word links this text graph can be considered as a unigram relation between two consecutive words in a specific document. This type of text graph is formulated as a graph-based structure, denoted as: $\mathcal{G}^{ww} = \{\mathcal{V}^{ww}, \mathcal{E}^{ww}\}$, with: $\mathcal{V}^{ww}$ is a set of unique word nodes in a given corpus ($\mathcal{D}$), or $\mathcal{V}^{ww} = \mathcal{W}$ and $\mathcal{E}^{ww}$ is the set of edges which present for the word co-occurring relationships. The edge's weight is identified as the cosine similarity on the corresponding embedding vectors of two specific (i^{th}) and (j^{th}) words which are achieved previously, denoted as: $\text{cosine}(w_i, w_j) = \frac{\overrightarrow{w_i}\overrightarrow{w_j}}{\|\overrightarrow{w_i}\|.\|\overrightarrow{w_j}\|}$.
- **Word-document text graph**. This text graph presents the relationships between documents and their occurred words, denoted as: $\mathcal{G}^{wd} = \{\mathcal{V}^{wd}, \mathcal{E}^{wd}\}$, with: $\mathcal{V}^{wd} = \{\mathcal{W}, \mathcal{D}\}$ is set of word and document nodes in a given corpus ($\mathcal{D}$) and $\mathcal{E}^{wd}$ presents the word-document occurring relationships with the edge's weight is identified as the TF-IDF score between a specific word and its included document.
- **Word-topic proximity text graph**. This text graph presents for the relationships between a specific words and extracted latent topics through topic modelling approach. The word-topic text graph is formed as: $\mathcal{G}^{wt} = \{\mathcal{V}^{wt}, \mathcal{E}^{wt}\}$, with: $\mathcal{V}^{wt}$, is a set of word and latent topic nodes, as: $\mathcal{V}^{wt} = \{\mathcal{W}, \mathcal{T}\}$ and ($\mathcal{T}$) is a set of indexed latent topics and $\mathcal{E}^{wt}$ presents for a set of word and highest distributed latent topic relationship. The probabilistic topic distribution between a specific word (w) and topic (t) is used as the edge's weight.
- **Document-topic proximity text graph**. Similar to the word-topic text graph, the document-topic text graph presents for the relationships between documents and their highest distributed extracted latent topics, denoted as: $\mathcal{G}^{dt} = \{\mathcal{V}^{dt}, \mathcal{E}^{dt}\}$. The $\mathcal{V}^{wt}$ presents for a set of document and latent topic nodes, as: $\mathcal{V}^{wt} = \{\mathcal{D}, \mathcal{T}\}$, and $\mathcal{E}^{dt}$ is a set of document-highest distributed latent topic relationships. The edge's weight is identified as a highest probabilistic distribution between a specific document (d) and a latent topic (t).

These different constructed text graphs are utilized to form an unified heterogeneous text graph with multi-typed nodes and relationships which presents the overall structural relationships of a given corpus , denoted as: $\mathcal{G}^{\mathcal{D}} = \{\mathcal{G}^{ww}, \mathcal{G}^{wd}, \mathcal{G}^{wt}, \mathcal{G}^{dt}\}$.

2) Heterogeneous text graph embedding via graph convolutional network (GCN)

Majorly inherited from recent GCN-based text embedding techniques [4] [27] [28] for the classification problem, we applied a (k)-layered GCN architecture to learn the structural representations of multi-typed network nodes as the aggregated message-passing between their corresponding relationships. In efficiently integrated with previous learnt topic-driven contextual embeddings of word, document and latent topic nodes, we utilized these embedding matrices as the initial node features. These

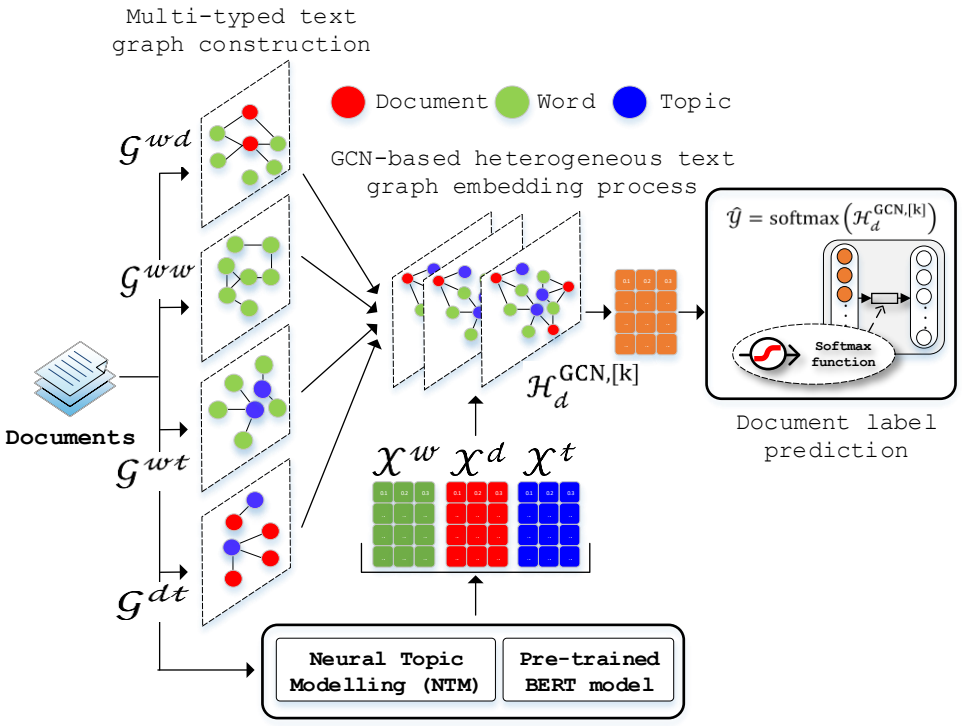

Figure 1. Illustration of the overall architecture of our proposed GOWTopGCN model

node feature matrices are not trainable during the GCN-based propagation learning process. In general, the GCN-based heterogeneous text graph embedding process can be formulated as the following (as shown in the equation 5 and 6):

$$\mathcal{H}^{\mathrm{GCN},[0]} = \mathrm{ReLu}\big(\mathrm{W}^{\mathrm{GCN},[0]}.\mathcal{X} = \{\mathcal{X}^{w}, \mathcal{X}^{d}, \mathcal{X}^{t}\}.\hat{\mathcal{A}}\big) \quad (5)$$

$$\mathcal{H}^{\mathrm{GCN},[\mathrm{l}]} = \mathrm{ReLu}\big(\mathrm{W}^{\mathrm{GCN},[\mathrm{l}-1]}.\mathcal{H}^{\mathrm{GCN},[\mathrm{l}-1]}.\hat{\mathcal{A}}\big) \quad (6)$$

In these equations, the ($\hat{\mathcal{A}}$) is the normalized adjacency matrix of a given heterogeneous text graph, defined in GCN [23] and ($\mathrm{W}^{\mathrm{GCN}}$) is the trainable weighting parameter matrix of each GCN-based layer. Through the graph-structured embedding information which is propagated through different (k) layers, we achieved the final document node embeddings at the last (k^{th}) output hidden states which are then fed to a classification task-driven full-connected layer with the softmax function to conduct text categorization, denoted as: $\hat{\mathcal{Y}} = \mathrm{softmax}\big(\mathcal{H}_d^{\mathrm{GCN},[\mathrm{k}]}\big)$, with ($\mathcal{H}_d^{\mathrm{GCN},[\mathrm{k}]}$) presents for the out hidden states as GCN-based embedding vectors of document nodes. Finally, the cross-entropy loss strategy is applied to train the overall GOWTopGCN architecture as the following: $\mathcal{L} = -\sum_{\mathrm{i}\in\mathrm{T}}\sum_{\mathrm{c}\in\mathrm{C}}\mathcal{Y}(\mathrm{c})\log\hat{\mathcal{Y}}(\mathrm{c})$, with (T) is a set of annotated documents for training purposed which are categorized into (C) different classes/labels. To efficiently optimize model's parameters, we apply the stochastic gradient descent (SGD) strategy upon the Adam optimizer with a predefined learning rate (η). The Figure 1 illustrates the overall architecture and associated components of our proposed GOWTopGCN model in this paper

IV. EXPERIMENTS AND DISCUSSIONS

In this section, we demonstrate extensive experiments in benchmark text datasets, like as: Ohsumed, IMDb, Yelp-2014, DBLP and arXiv to demonstrate the effectiveness of our proposed GOWTopGCN model in comparing with recent state-of-the-art text classification baselines, like as: GloVe [6], fastText [7], VDCNN [13], BERT [16] and TextGCN [4].

A. Dataset usage and experimental setups

1) Dataset description

To evaluate the accuracy performances of different text classification baselines, we mainly used different standard textual datasets which have been widely utilized in previous works [13] [4] [19], including:

- **Ohsumed:** is considered as a popular text corpus which contains medical scientific articles that are gathered from the well-known MEDLINE online library. This dataset contains > 7.4K documents which are arranged into 23 categories.
- **IMDb:** this dataset contains customer's reviews with the corresponding rating scores from 1 to 10 about movies which are published in the IMDb platforms. This dataset contains about 50K reviews in forms of length-varied textual documents.
- **Yelp-2014**: similar to the IMDb, the Yelp is also considered as the large-scaled customer's review dataset which belongs to the Yelp challenge (https://www.yelp.com/dataset). This dataset contains > 8M reviews of customers on the over 160K local businesses/services which arranged into a ranking range of [1-5]. For this dataset we randomly selected 500K customer's reviews for conducting experiments in this paper.
- **DBLP**: is also a large-scaled text corpus which contains abstract contents of > 5M scientific papers in computer science domain. For experiments in this paper, we randomly selected about 260K papers in this dataset which are labelled into 12 main topics/areas of computer science domain, following the ACM CCS-2012 classification (https://dl.acm.org/ccs).
- **arXiv**: contains > 730K pre-printing scientific articles which are published in the arXiv online library platform (https://arxiv.org/). For the experiments in this paper, we randomly select 600K documents. These scientific articles are categorized into 8 domains, like as: economics, mathematics, physics, quantitative biology, computer science, etc. following the arXiv category taxonomy (https://arxiv.org/category_taxonomy).

2) Textual pre-processing steps and latent topic extraction.

To effectively handle textual pre-processing steps like as: stop-word removal, word tokenization, stemming, etc. we mainly applied the Stanford CoreNLP library (https://stanfordnlp.github.io/CoreNLP/). For the implementation of NTM model, we reused the original source code of Miao, Y. et al. in the NVDM model [21] to achieve the topic-word and document-topic distributions in these given datasets with the number of latent topics are set to 10 for all datasets ($\mathrm{K} = 10$).

3) Model implementations and configurations

For the implementation of our proposed GOWTopGCN, we mainly use the Python programming language with the supports of the well-known PyTorch machine learning framework (https://pytorch.org/). For the implementation of multi-layered GCN based architecture in our proposed model, we mainly utilized the geometric neural network architecture libraries which are provided at the extended PyTorch-Geometric plugin (https://pytorch-geometric.readthedocs.io/). For all the experiments in this paper, we set up the proposed GOWTopGCN model and other comparative baselines (described in section 4)) in a single server with Intel Xeon SKL-SP 4210 CPU and 64Gb RAM.

Pre-trained BERT usage. For the implementation of pre-trained BERT model [16], we mainly used the original pre-trained BERT (large/uncased version) which is officially released by Google at this repository (https://github.com/google-research/bert). We kept the default BERT-based word embedding vector size as: 768 ($\mathrm{d}^{\mathrm{BERT}} = 768$). For the dimensionality of general embedding vector after the linear transformation process, we set it as: 256 ($\mathrm{d}^{wdt} = 256$). For the number of LSTM-based cells in the Bi-LSTM architecture which is used for producing the completed document embedding (as

described in section III.B.1)), we set it as: 256 ($h^{LSTM} = 128$). For the number of GCN-based layers in our heterogeneous text graph representation learning (as described in section III.B.2)), we configured it as: 3, ($k^{GCN} = 3$) and the dimensionality of network node embedding is set to 256 ($d^{GCN} = 256$). Table 1 shows other configurations of our proposed GOWTopGCN for experiments in this paper.

Table 1. Configurations of GOWTopGCN model for experiments

Model's configuration	Value
The BERT-based word embedding size (d^{BERT})	768
Dimensionality of general word, document and topic embedding vector after transformation (d^{wdt}) (described in section III.B.1))	256
Number of LSTM-based cells for document embedding through Bi-LSTM archiecture (h^{LSTM}) (described in section III.B.1))	128
The number of GCN-based layer for heterogeneous text graph embedding process (k^{GCN}) (described in section III.B.2))	3
Default model's optimizer.	Adam
Default learning rate (η).	1×10^{-4}
Default dropout rate for all neural network arachiectures.	0.5

4) Experimental setups and evaluation methods

For datasets (as listed in sub-section IV-A-1) which are used for experiments in our paper, majorly inspired from empirical studies in previous works [4] [29], we applied the splitting strategy for these datasets as shown in Table 2. Specifically, for the training processes of different deep learning based text representation learning and classification, we divide the training set of each dataset into two sub-sets, model training (90%) and validation/development (10%). Then, the trained models are assessed the accuracy performance with the testing set under the F-1 evaluation metric. For each model, we run the evaluation 10 times and reported the average outputs in each dataset as the final results.

Table 2. Detailed information about dataset splits for all experiments in this paper

Dataset	Training set size	Testing set size	N.o Classes
Ohsumed	3,357	4,043	23
IMDb	25,000	25,000	10
Yelp-2014	250,000	250,000	10
DBLP	100,000	163,921	12
arXiv	300,000	300,000	8

B. Comparative text classification methods

To compare the performance of our proposed GOWTopGCN model with other text classification baselines, we also implemented several methods for text classification, which are:

- **GloVe** [6]: is a well-known word embedding technique which enables to capture the rich local contextual information of words within a text corpus. To apply the GloVe model [6] for the text classification task, we utilized the large-scaled pre-trained GloVe model to achieve the embedding vector of each document. Then the document embeddings are produced by taking the average of all occurred word embeddings. Finally, the achieved document embeddings are combined with the SVM classifier to handle the text classification task.
- **fastText** [7]: similar to the utilization of pre-trained GloVe model, we also used the pre-trained fastText [7] word embedding platform. For experiments in this paper, we used fastText to produce the completed document representations similar to the utilization of GloVe model. Then, these achieved document embedding are later use to facilitate the SVM classifier for handling classification task.
- **VDCNN** [13]: is recently proposed by Conneau, A. et al. for deeply learning the rich semantic pattern-based features of texts in which can be fine-tuned for different tasks including the classification. In the VDCNN model, the input textual data is deeply analyzed to extract textual latent featured at the character level. For experiment in this paper, we mainly utilized the original implementation of VDCNN model of Conneau, A. et al. with 29 convolutional layers and fine-tuned for text document classification task.
- **BERT** [19]: is considered as the most popular and powerful transformer-based architecture for rich contextual text representation. In the BERT model, Devlin, J. et al. [19] proposed a flexible transformer-based architecture in which can be fine-tuned for different tasks, including classification. For the implementation of BERT in our experiments, we reused the pre-trained BERT model which is official released by Devlin, J. et al. and set up for handling text classification task.
- **TextGCN** [4]: is a recent state-of-the-art GNN-based approach for rich schematic textual embedding and text classification task. In the TextGCN model, Yao, L. et al. proposed a novel heterogeneous text graph construction technique to preserve different global structural relationships between words and documents in a given text corpus. For experiments with a specific dataset, different constructed text graphs are fed to a multi-layered GCN architecture to achieve the document node embeddings through the graph-based propagation learning procedure. Finally, the achieved document node embeddings are utilized in the text classification process through a full-connected layer at the end. The TextGCN is considered as our main competitor in this paper.

For the unique configurations of these listed above comparative techniques, we configured them as the same in their original papers in which these techniques achieve the highest accuracy performances. For other

configurations which are similar to our GOWTopGCN model, we set them as the same as described in Table 1.

C. Experimental results and discussions

The Table 3 shows the experimental outputs for text classification task in multiple datasets, like as: Ohsumed, IMDb, Yelp-2014, DBLP and arXiv by using different techniques. In general, as shown the results, our proposed GOWTopGCN model explicitly achieved better performances than recent state-of-the-art baselines. Specifically, in comparing with traditional pre-trained word embedding approach like as GloVe and fastText, the GOWTopGCN model significantly outperforms averagely 54.69%. In comparing with recent advanced deep learning based techniques like as: VDCNN and BERT, our proposed model also obtained better performance approximately 25.67% (VDCNN) and 12.32% (BERT) for all datasets. For our main competitor which is the TextGCN model, GOWTopGCN also slightly improve the accuracy performance about 5.97% in terms of F1 evaluation metric for all datasets.

Table 3. Experimental outputs in terms of F1 evaluation metric for text classification task via different methods in benchmark datasets

Model/dataset	Ohsumed	IMDb	Yelp-2014	DBLP	arXiv
GloVe	0.42892	0.32812	0.46782	0.39712	0.42078
fastText	0.48891	0.46821	0.59092	0.40882	0.43781
VDCNN	0.65921	0.51921	0.57821	0.52782	0.44672
BERT	0.71921	0.50892	0.62902	0.60921	0.58921
TextGCN	0.68021	0.52467	0.67891	0.67612	0.67892
GOWTopGCN	**0.72812**	**0.55892**	**0.70812**	**0.71029**	**0.72681**

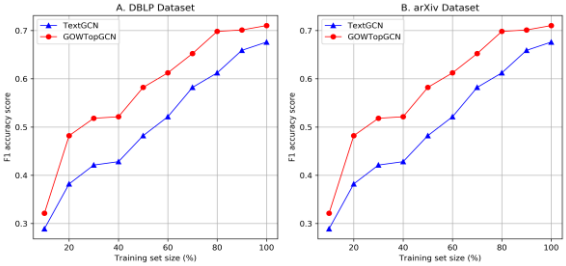

Figure 2. Experimental studies on the stability of TextGCN and GOWTopGCN models for text classification task in the DBLP and arXiv datasets

Model's stability is also considered as an important aspect which should be carefully evaluated to ensure the proposed algorithm can work well on different training set size situations. We have varied the training set size (%) of two large-scale datasets which are DBLP and arXiv from 10% to 100% and reported the changes in the classification outputs of our proposed GOWTopGCN and TextGCN models. As shown from the experimental results in Figure 2, both two GOWTopGCN and TextGCN models are considered as quite stable with different size of training set. The accuracy performances of both two models gradually raised accordingly with the increases in the volume of data which is used for training. These experimental output prove the fact that our proposed GOWTopGCN model can be affordable to be applied in the context of low training resources.

In general, comparative results between our GOWTopGCN model and other baselines in text classification task have demonstrated the effectiveness of our proposed ideas in this paper. GOWTopGCN is a combination between the neural topic modelling and the rich structural and contextual representation learning in texts via BERT and GCN in which can effectively assist for the fine-tuning process in text classification problem.

D. Model parameter sensitivity analysis

In this section, we conducted extensive experiments in evaluating the effects of important model's parameters in the overall accuracy performance of our proposed GOWTopGCN model. First of all, we studied the influence of dimensionality of node embedding vector (d^{GCN}) parameter on the text classification performance in large-scale datasets, like as: arXiv and DBLP. We have varied the values of this parameter within range [10-400] and reported the changes in the accuracy performances for the text classification task. As shown from the experimental outputs in Figure 3-A, it showed that our model is quite insensitive with this parameter in which the proposed GOWTopGCN model achieved a stable performance with the value of (d^{GCN}) parameter > 200. Similar to empirical studies on the (d^{GCN}) parameter, we also evaluated the influence of number of GCN-based layers (k^{GCN}) parameter, on the overall performance of our model. As shown from the experimental outputs in Figure 3-B, our proposed GOWTopGCN model gained the stability with value of (k^{GCN}) parameter > 2 and the increase in number of GCN-based layers doesn't affect much on the overall accuracy performance of our model.

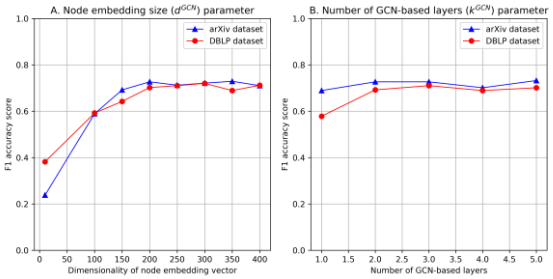

Figure 3. Ablation studies on the influences of network node embedding vector size (d^{GCN}) and number of GCN-based layers (k^{GCN}) in our proposed GOWTopGCN model

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed a novel topic-driven contextual and structural text embedding approach for text classification task, called as GOWTopGCN. Our proposed GOWTopGCN model is an integration between neural topic modelling with the rich contextual and structural text representation learning approach by using BERT and GCN. First of all, to efficiently extract latent topic distributions over words and documents, we utilize the variational auto-encoding (VAE) mechanism upon the

Gaussian latent topic modelling paradigm. Then, the achieved latent topic distributions are integrated with BERT-based embedding strategy to produce rich global semantic representations of words and documents in a given text corpus. Then, these rich semantic representations are utilized in the process of multi-typed text graph construction and graph-structured representation learning via the multi-layered GCN architecture. Extensive experiments in benchmark datasets demonstrate the effectiveness and outperformances of our proposed GOWTopGCN model in comparing with recent baselines for text classification task. For our future works, we intended to apply our proposed GOWTopGCN model in different primitive tasks of NLP domain, like as sentiment analysis and spam filtering.

VI. ACKNOWLEDGEMENT

This research is funded by Thu Dau Mot University, Binh Duong, Vietnam under grant number DT.21.2-062.

REFERENCES

[1] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D., "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.

[2] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J., "Deep Learning--based Text Classification: A Comprehensive Review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1-40, 2021.

[3] Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., ... and Iyengar, S. S., "A survey on deep learning: Algorithms, techniques, and applications," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1-36, 2018.

[4] Yao, L., Mao, C., and Luo, Y., "Graph convolutional networks for text classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[5] Mikolov, T., Chen, K., Corrado, G., and Dean, J., "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations (ICRL)*, 2013.

[6] Pennington, J., Socher, R., and Manning, C. D., "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

[7] Mikolov, T., Grave, É., Bojanowski, P., Puhrsch, C., and Joulin, A., "Advances in Pre-Training Distributed Word Representations," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, 2018.

[8] Le, Q., and Mikolov, T., "Distributed representations of sentences and documents," in *International conference on machine learning, PMLR*, 2014.

[9] Tai, K. S., Socher, R., and Manning, C. D., "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015.

[10] Zhu, X., Sobihani, P., and Guo, H., "Long short-term memory over recursive structures," in *International Conference on Machine Learning (PMLR)*, 2015.

[11] Liu, P., Qiu, X., Chen, X., Wu, S., and Huang, X. J., "Multi-timescale long short-term memory neural network for modelling sentences and documents," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015.

[12] Blunsom, P., Grefenstette, E., and Kalchbrenner, N., "A Convolutional Neural Network for Modelling Sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

[13] Conneau, A., Schwenk, H., Cun, Y. L., and Barrault, L., "Very deep convolutional networks for text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017.

[14] Sutskever, I., Vinyals, O., and Le, Q. V., "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems*, 2014.

[15] Bahdanau, D., Cho, K., and Bengio, Y., "Neural machine translation by jointly learning to align and translate," *3rd International Conference on Learning Representations, ICLR*, 2015.

[16] Vaswani, Ashish, et al., "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[17] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L., "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

[18] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I., "Improving language understanding by generative pre-training," *OpenAI*, 2018.

[19] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K., "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*, 2019.

[20] Lu, Z., Du, P., and Nie, J., "VGCN-BERT: Augmenting BERT with Graph Embedding for Text Classification," in *Advances in Information Retrieval*, 2020.

[21] Garg, S., and Ramakrishnan, G., "BAE: BERT-based Adversarial Examples for Text Classification," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[22] Rousseau, F., Kiagias, E., and Vazirgiannis, M., "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*, 2015.

[23] Wang, C., Song, Y., Li, H., Zhang, M., and Han, J., "Text classification with heterogeneous information network kernels," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[24] Hamilton, W. L., Ying, R., and Leskovec, J., "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[25] Kipf, T. N., and Welling, M., "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[26] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y., "Graph Attention Networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[27] Miao, Y., Grefenstette, E., and Blunsom, P., "Discovering discrete latent topics with neural variational inference," in *International Conference on Machine Learning (PMLR)*, 2017.

[28] Liu, X., You, X., Zhang, X., Wu, J., and Lv, P., "Tensor graph convolutional networks for text classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[29] Zhang, H., and Zhang, J., "Text Graph Transformer for Document Classification," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[30] Bashri, M. F., and Kusumaningrum, R., "Sentiment analysis using Latent Dirichlet Allocation and topic polarity wordcloud visualization," in *5th International Conference on Information and Communication Technology (ICoIC7)*, 2017.

ỨNG DỤNG MẠNG TÍCH CHẬP ĐỒ THỊ VÀ MÔ HÌNH CHỦ ĐỀ ĐỂ NÂNG CAO HIỆU QUẢ PHÂN LỚP VĂN BẢN

Tóm tắt—Trong thời gian gần đây, chúng ta đã chứng kiến sự tiến bộ đáng kể của các kiến trúc deep neural nâng cao trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Dù vậy, các kỹ thuật này vẫn tồn tại nhiều hạn chế liên quan đến khả năng duy trì sự phụ thuộc trong phạm vi dài giữa các từ và tài liệu. Do đó, một số phương pháp tiếp cận dựa trên mạng nơ ron đồ thị từ (GNN) để phân lớp văn bản đã được đề xuất gần đây để giải quyết các thách thức này. Tuy nhiên, các kỹ thuật này cũng gặp phải nhiều khó khăn trong việc tích hợp thông tin ngữ nghĩa toàn cục của văn bản. Các thông tin này có thể hỗ trợ để tạo điều kiện thuận lợi cho quá trình học biểu diễn văn bản nhằm đạt được mục tiêu điều chỉnh phân lớp tốt hơn. Để vượt qua những khó khăn này, trong bài báo này, chúng tôi đề xuất một phương pháp mới để nhúng văn bản dựa trên mạng GNN theo chủ đề, GOWTopGCN. Mô hình được đề xuất của chúng tôi cho phép biểu diễn thông tin ngữ nghĩa toàn cục phong phú và các mối quan hệ phụ thuộc trong phạm vi dài giữa các từ và tài liệu. Các thử nghiệm với các bộ dữ liệu văn bản chuẩn đã chứng minh được sự vượt trội của mô hình được đề xuất so sánh với các mô hình hiện đại gần đây về phân lớp văn bản dựa trên GNN.

Từ khóa—neural topic modelling; BERT; GCN; đồ thị từ.

AUTHORS' BIOGRAPHIES

Tham Thi Hong Vo received the M.S. degree in Computer Science from University of Information Technology (UIT), Ho Chi Minh, Vietnam in 2009. She received the PhD degree from Lac Hong University, Dong Nai, Vietnam in 2021. She is working at Institute of Engineering and Technology, Thu Dau Mot University, Binh Duong, Vietnam. Her researches focus on Text Mining, Data Stream Analysis, Deep Learning, Information Network Analysis and Mining, Text Representation Learning.