Applied Intelligence
https://doi.org/10.1007/s10489-022-04106-x

Label-representative graph convolutional network for multi-label text classification

Huy-The Vu[1] · Minh-Tien Nguyen[1] · Van-Chien Nguyen[2] · Minh-Hieu Pham[3] · Van-Quyet Nguyen[1] · Van-Hau Nguyen[1]

Accepted: 23 August 2022

Abstract
Multi-label text classification (MLTC) is the task that assigns each document to the most relevant subset of class labels. Previous works usually ignored the correlation and semantics of labels resulting in information loss. To deal with this problem, we propose a new model that explores label dependencies and semantics by using graph convolutional networks (GCN). Particularly, we introduce an efficient correlation matrix to model label correlation based on occurrence and co-occurrence probabilities. To enrich the semantic information of labels, we design a method to use external information from Wikipedia for label embeddings. Correlated label information learned from GCN is combined with fine-grained document representation generated from another sub-net for classification. Experimental results on three benchmark datasets show that our model outweighs prior state-of-the-art methods. Ablation studies also show several aspects of the proposed model. Our code is available at https://github.com/chiennv2000/LR-GCN.

Keywords Graph convolutional network · Multi-label classification · Correlation matrix · Label embedding · Label correlation

1 Introduction

Multi-label text classification (MLTC) is a subproblem of text classification that classifies input text/documents into pre-defined classes (labels) [1]. MLTC can be applied to a wide range of applications such as patent classification [2], sentiment analysis [3], and mobile applications [4]. Different from multi-class classification which only assigns one label to the given text, multi-label classification classifies the text into the most relevant multiple labels from the label set [5]. Formally, let $\mathcal{D}$ be a set of n documents, $\mathcal{Y} = \{c_1, c_2, \ldots, c_k\}$ be the label space with k class labels, $\boldsymbol{\mathcal{X}} = \mathbb{R}^{n \times m}$ denotes the m dimensional feature space corresponding to n documents. The task is to learn a mapping function $h : \boldsymbol{\mathcal{X}} \rightarrow 2^{|\mathcal{Y}|}$ from the training set $\mathcal{D} = \{\boldsymbol{x}_i, Y_j \| 1 \leq i \leq n \text{ and } 1 \leq j \leq k\}$, where $\boldsymbol{x}_i \in \boldsymbol{\mathcal{X}}$ is a feature vector of the document i^{th} and $Y_j \subset \mathcal{Y}$ is a set of labels of $\boldsymbol{x}_i$. In MLTC tasks, multiple labels can be assigned to a given document. This results in an increase in the co-occurrence frequency of labels [6]. Therefore, it is desirable to model such label correlation to improve the classification performance of MLTC models.

Recently, deep neural networks for multi-label text classification have been investigated [1, 5–9]. The networks try to automatically learn the text representation by using different architectures such as CNN [1], RNN [5], or BERT-based methods [10, 11]. However, these works often focus on exploring document information and ignore label

✉ Van-Hau Nguyen
haunv@utehy.edu.vn

Huy-The Vu
thevh@utehy.edu.vn

Minh-Tien Nguyen
tiennm@utehy.edu.vn

Van-Chien Nguyen
chien.nv183488@sis.hust.edu.vn

Minh-Hieu Pham
phamhieu30091997@gmail.com

Van-Quyet Nguyen
quyetict@utehy.edu.vn

[1] Hung Yen University of Technology and Education, Hung Yen, Vietnam

[2] Hanoi University of Science and Technology, Hanoi, Vietnam

[3] Foreign Trade University, Hanoi, Vietnam

Published online: 02 November 2022

information. This may lead to information loss. Let's take Table 1 as an example.

We can observe two interesting points from the table. The first one is that some labels appear in the text (e.g. bold words). This shows that label semantics need to be taken into consideration when learning the hidden pattern of text, which can contribute to the performance of multi-label classifiers. The second observation is that labels occur (e.g. bean, oilseed) and co-occur in multiple documents (e.g. corn, grain). This raises a question about how to model correlation among labels effectively, in order to improve classification performance. Prior works [5, 12, 13] attempted to use label information for improving classification performance. However, these methods only focus on exploring label semantics, by using attention mechanisms. Recently, models based on graph neural networks (GCN) have been introduced to capture label correlation [11, 14]. However, modeling label correlation in such GCN-based models is still challenging because of the over-smoothing problem [14]. Furthermore, these models often ignore node embedding initialization [11].

To overcome the issues raised above, we propose a new model that explores both label correlation and label semantics by using the graph convolutional network for the MLTC task in this paper. To capture label correlation, we design a correlation matrix using the occurrence and co-occurrence probabilities of labels. The label correlation is modeled in a modified form of point-wise mutual information, which is widely used in computational linguistics to capture associations between words [15]. This matrix allows the model to effectively propagate label information among GCN nodes. For label semantics, we use word embeddings of labels to initialize node features which are often ignored in other GCN-based models [11, 15]. More importantly, to enhance label embeddings, we propose a simple method using external language resources. Simultaneously, we adopt a BERT-based model (i.e. RoBERTa [16]) as another sub-net to extract fine-grained information from input documents. Both label and document representations are then combined before being fed to a fully-connected neural network for classification. Evaluation results on three benchmark datasets show that our model outweighs prior state-of-the-art classification methods. In addition, ablation studies are also conducted to explore more insights into the proposed model.

In summary, the main contributions of this paper are summarized as follows:

- We propose a new end-to-end trainable multi-label text classification model. The proposed model takes into account the combination of label information learned by GCN and contextual document representations using a BERT-based model.
- We build an effective correlation matrix to guide label information propagated among label nodes. Furthermore, in order to improve label representation learning, we propose to enhance the word embedding of labels by using external language resources to initialize node features.
- We conduct extensive experiments to evaluate the proposed model, then compare it with strong baselines. We also perform ablation studies to deeply analyze several important aspects of the proposed model.

The remainder of this paper is organized as follows: Section 2 overviews related works for multi-label text classification. Section 3 describes the proposed model. Section 4 presents the experiments to validate the effectiveness of the proposed model and also ablation studies to explore some aspects of the model. The conclusion is summarized in Section 5.

Table 1 The text and labels of three samples in the reuters-21578 dataset

Sample	Text	Label
1	CHINA SWITCHES U.S. **WHEAT** TO...The department said outstanding **wheat** sales to China for...Total **corn** commitments for the...	corn, grain, wheat
2	...The USSR has purchased 2.40 mln tonnes of U.S. **corn** for...amounted to 152,600 tonnes of **wheat**, 6,808,100 tonnes of **corn** and 1,518,700 tonnes...	acq, corn, grain
3	...The **corn** wet milling business acquired by the Italian group...it had agreed in principle to sell its European **corn** wet millng business...	corn, grain, oilseed, soybean, wheat

Bold words appear in the label set

2 Related work

The existing models for multi-label classification can be categorized into two main groups: document-based and document-label-based methods.

Document-based methods These algorithms only explore document information. In the early stage, classical machine learning models were proposed to address MLTC [17, 18] which could be solved by either the data transformation approach [17] or the algorithm adaptation approach [18]. However, these methods are limited by the need for tedious feature engineering and analysis to achieve good performance. Furthermore, it is difficult to generalize to new tasks because of the strong dependence on domain knowledge for designing features. In recent years, deep learning-based models have been shown as a powerful tool for MLTC tasks, without the requirement of handcrafted features. In these models, the main idea is to automatically learn the hidden representation of text by using a large number of training samples. One of the first attempts was introduced by [19]. The authors used a convolutional architecture, namely dynamic CNN to take advantage of a changeable width of convolutional layers with dynamic k-max pooling to detect sequences of words for indicating the topics. After that, several architectures have been investigated to learn hidden representations such as CNN [1], LSTM [5, 20], graph convolution neural network (GCN) [7, 15, 21], or transformers [10, 11]. To improve the quality of classifiers, attention mechanisms have been adapted for MLTC [11, 12]. These methods operate on hidden vectors to force classifiers to focus more on important words regarding labels (please refer to Table 1). Although the models mentioned above have achieved quite promising results for MLTC tasks, ignoring label information may lead to information loss [11–13]. Therefore, we argue that MLTC models need to take into account label information.

Document-label-based methods These models benefit from label information to improve the quality of the MLTC task [5, 11–13, 22]. In such models, multi-label classifiers have to not only consider the content of text but also need to explore label information, which is mostly done using attention mechanisms. An important work in this direction was proposed in [22]. This work views text classification as a label-word joint embedding problem. Particularly, each label is embedded in the same space with word vectors by introducing an attention framework. After that, You et al. [5]S introduced label tree-based attention based on RNN for extreme MLTC. The authors combined a multi-label attention mechanism for text and a probabilistic label tree for labels. Xiao et al. [12] presented a label-specific attention network (LSAN) for MLTC. LSAN determines the semantic relationship between labels and documents to construct label-specific document representation by using label semantic information. An interesting idea of using label information was presented in [23]. In this method, each category label is associated with a category description which is generated by hand-crafted templates or using abstractive/extractive models from reinforcement learning. The description is then attended to the most salient texts to improve classification performance. Some other works based on attention mechanisms are hybrid attention [13], attentional ordered recurrent neural network [9], and history-based label attention [8]. As described above, these works mainly consider the semantics of labels by using attention mechanisms to integrate label embeddings into document representation. This may not take full advantage of label information [10].

To alleviate the limitation mentioned above, graph neural network-based models have been introduced to take into account both label semantics and label correlation. Cai et al. [11] introduced a hybrid neural network that combines label information and fine-grained text representation. The model uses BERT for text representation combined with a label graph for integrating label information into the network. However, this work only explores label correlation, ignoring the initialization of node features. MAGNET [24] introduced a graph attention network-based model using both label embeddings as node features and label correlation. We share this idea with MAGNET but our model significantly differs from it. Firstly, we propose a new method to build a correlation matrix from input labels (as explained and compared in Section 3.2.1). Secondly, we use a method to enrich label embeddings before initializing node features. Finally, our subnets of GCN, BERT, and MLP differ significantly from MAGNET.

3 Label representative GCN for muti-label text classification

This section describes the proposed model for multi-label text classification. We first present a high-level view of the proposed model architecture. We then present three main parts of our model in more detail, including label representation learning, document representation learning, and classification.

3.1 Overall model architecture

As shown in Fig. 1, the proposed model consists of two sub-nets for extracting label and document representations and a classifier wrapped on the top. As mentioned, our model shares the idea of [14]. However, it makes three significant differences. First, we create an efficient

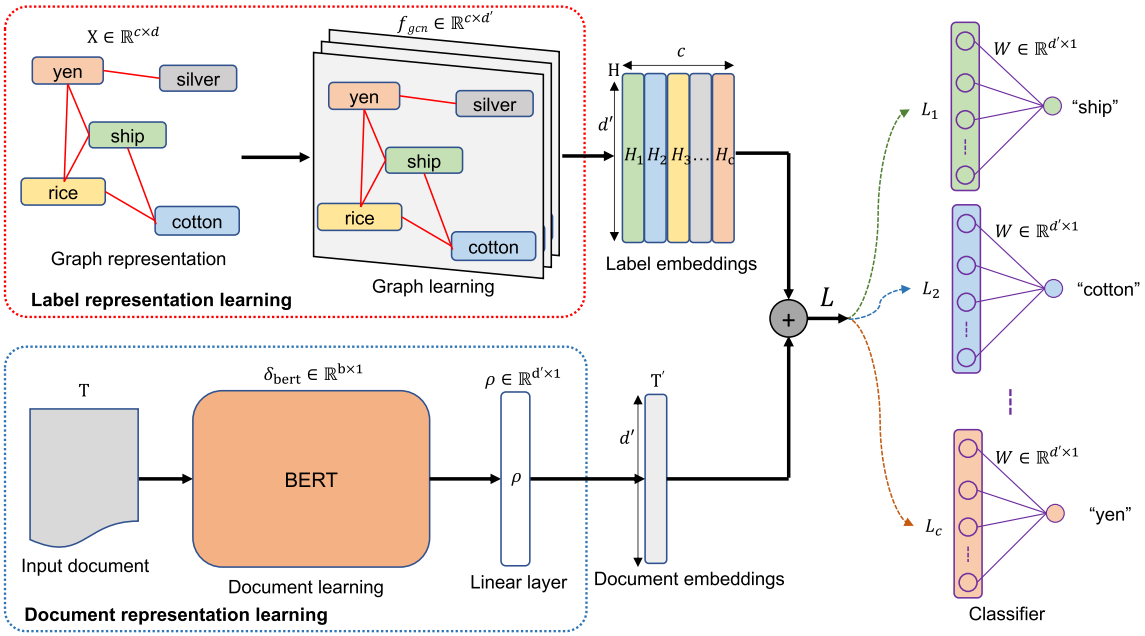


Fig. 1 The overall architecture of our LR-GCN model for multi-label text classification. For label representation learning, the graph $G = (V, E)$ is built from class labels, in which label-word embeddings ($X \in \mathbb{R}^{c \times d}$) are used as the feature of vertices V (c is the number of classes and d is the dimensionality of embedding vectors). Edges E of the graph G (i.e. correlation matrix $A \in \mathbb{R}^{c \times c}$) are built from occurrence and co-occurrence probabilities of labels. A stacked GCN architecture is then adopted to learn over the graph. Its output $H \in \mathbb{R}^{c \times d'}$ is combined with the document representation vector generated from a BERT-based model. These added vectors are fed to a fully-connected neural network $W \in \mathbb{R}^{d' \times 1}$ for classification

correlation matrix, which captures the dependencies among labels. The matrix creation uses the modification of point-wise mutual information for computing the weights of edges in an undirected graph. On the contrary, the work in [14] builds a directed graph with a correlation matrix modeling the label correlation dependency in the form of conditional probability. Secondly, we use RoBERTa [16] to extract fine-grained document information since it has been shown to be a powerful tool for generating document representations [16]. Finally, our model uses a fully-connected neural network as a classifier wrapped on the top instead of using dot-product to calculate predicted output as in [14]. Our preliminary experimental results show that using the classifier improves classification performance.

3.2 Label representation learning

In LR-GCN, we take into account label information regarding both correlation and semantics. This section presents how the correlation matrix and node features are built and then learned using GCN.

3.2.1 Correlation matrix creation

In this paper, we consider a graph $G = (V, E)$, where V and E are vertices and edges of the graph G, respectively. The graph G is represented by a correlation matrix $A \in \mathbb{R}^{c \times c}$ ($c = |V|$, the number of document classes). The correlation matrix plays an important role in propagating label information among vertices in GCN. Therefore, building an effective correlation matrix is a crucial task in our model.

In LR-GCN, we build the correlation matrix in a data-driven manner where modeling label correlation is based on the occurrence and co-occurrence probabilities of labels. In [14], their work modeled the label correlation dependence in the form of conditional probability. MAGNET [24] also adopted this kind of correlation matrix for MLTC tasks. However, using the conditional probability for building the matrix seems to be only suitable for capturing the natural topology structure between label objects in images, as presented in [14]. In contrast, we build the label correlation matrix by using the occurrence and co-occurrence probabilities of labels. Our matrix is based on a modification of point-wise mutual information which is widely used in computational linguistics to capture associations between words [15].

To calculate the edge weight between nodes i and j, we count the occurrence and co-occurrence of labels to approximate the probabilities $p(i)$, $p(j)$ (i.e. occurrence times of label i and j) and $p(i, j)$ (i.e. co-occurrence times of label i and j). So each element A_{ij} of the correlation

matrix is defined as below:

$$A_{ij} = \frac{p(i,j)}{p(i)p(j)} = \frac{\#L_{(i,j)}\#D}{\#L_{(i)}\#L_{(j)}} \quad (1)$$

where $\#L_{(i,j)}$ is the number of documents having a pair of labels i, j, $\#L_{(i)}$ is the number of documents having the label i, and $\#D$ is the total number of documents in the dataset. Intuitively, an edge having a high $A_{i,j}$ value means that its vertices have a high correlation. It should be mentioned that we do not use the $log(.)$ function of A_{ij} in (1), as the original form of point-wise mutual information. Our preliminary experiments show that removing the $log(.)$ enables training time reduction and performance improvement.

Figure 2 illustrates an example of the matrix calculation. Input data includes three samples with six different labels. From input labels, we build a graph, in which nodes are represented by labels (i.e. embedding of label words). To calculate the correlation matrix, for example, the element $A_{oilseed,soybeam}$ (i.e. edge weight between nodes of $oilseed$ and $soybeam$), we count the number of documents having both labels $\#L_{(oilseed,soybeam)} = 1$, the numbers of documents having the label $\#L_{(oilseed)} = 1$, and $\#L_{(soybeam)} = 1$. Consequently, $A_{oilseed,soybeam} = 3.0$.

Finally, as suggested in [25], each node needs to aggregate features of all neighbors and also itself. Therefore, the final correlation matrix is formulated as below:

$$\hat{A}_{ij} = \begin{cases} A_{ij} & \text{if } i \neq j \\ 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

3.2.2 Node feature enrichment

For GCN, when $l = 0$ (the first layer), $H^{(0)} = X$ ($X \in \mathbb{R}^{c \times d}$) is the feature matrix of vertices (where d is the dimension of the feature vectors). In existing GCN-based models, node features are often not initialized (i.e. feature matrix $X = I$ as an identity matrix). However, in our work, since label embeddings that are learned from GCN are then combined with document representation (as shown in Fig. 1), initializing good feature nodes from input labels is an important task. To do this, we can simply employ any word embedding method such as Word2vec [26], Glove [27], and FastText [28]. While such word representations are able to capture some syntactic as well as semantic information, they need to be enhanced, especially for specific domain tasks [29].

Existing methods for enriching word embeddings can be categorized into three groups including joint, retrofitting, and post-specialization methods, as summarized in [29]. Our work shares the idea of the retrofitting methods which use external lexical resources to inject semantic information into pre-trained word vector representations via post-processing techniques but in a different way. In particular, given an input label, we first use Wikipedia-API[1] to retrieve the most relevant Wikipedia document. In our paper, we use external information from Wikipedia because it is an available and huge corpus. After that, we extract only the top two sentences that describe the label. This is based on the observation of text summarization, in which important information is usually mentioned in the top (two or three) sentences [30]. Extracting these sentences allows the length of the label description to meet with requirements of the next step while keeping the most important information relevant to the label, as shown in Fig. 3. Finally, instead of injecting this external information into pre-trained word vector representations as to the retrofitting methods, these sentences are then fed to Sentence-BERT[2] to generate label embeddings. Our experiment results show that this method could improve classification performance compared to other word embeddings [26–28]. This confirms the idea of using external information for enriching label node embeddings.

3.2.3 GCN for label information learning

GCNs are a neural network type that directly works on a graph [25]. In GCN, each layer aggregates the information of immediate neighbors. When stacking multiple layers together, the GCN model can integrate information from higher-order neighborhoods. In our work, GCN layers are considered to be a function $f_{gcn}(.)$ that is based on the correlation matrix to learn label information from the built graph. After aggregating neighbor's features, the output map of label representation learning generated from GCN layers is expressed as in (3):

$$H = f_{gcn}(\hat{A}, X, W_{gcn}) \quad (3)$$

where $f_{gcn}(.)$ is the GCN function, with its learnable weights W_{gcn} over the input feature matrix $X \in \mathbb{R}^{c \times d}$ (d is the dimensionality of label-word embeddings) and the correlation matrix $\hat{A} \in \mathbb{R}^{c \times c}$.

3.3 Document representation learning

This sub-net is designed to learn hidden representations from input documents. After pre-processing, these documents are fed into the model to extract fine-grained information. In general, we can adopt any document representation method, as summarized in [31]. Recently, BERT and its variation have shown to be a powerful tool for document representation [32, 33]. In this work, we employ RoBERTa [33], which is an improvement of BERT for this task [31,

[1] https://pypi.org/project/Wikipedia-API/

[2] https://www.sbert.net/

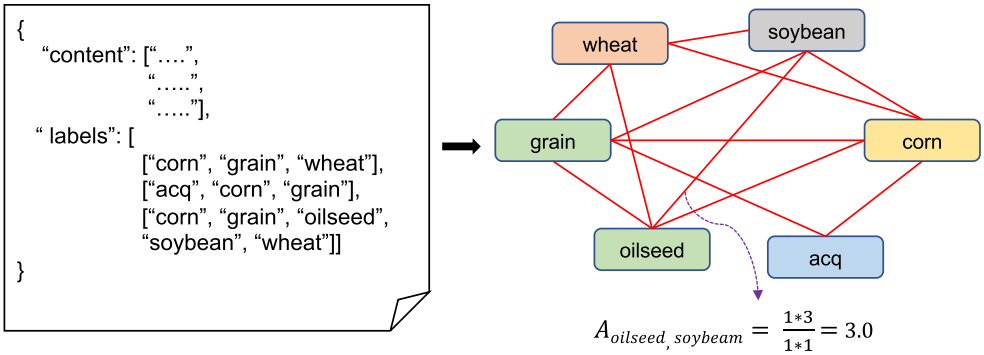


Fig. 2 An example of $A_{i,j}$ calculation

33]. As shown in Fig. 1, the output document representation of the BERT-based model $\delta_{bert}(.)$ is then projected to a linear layer $\rho(.)$, in order to change its dimension from b to d'. Consequently, the final output of the document representation learning part is expressed in (4):

$$T' = \rho(\delta_{bert}(T, \phi_{bert})) \tag{4}$$

where T and ϕ_{bert} are input and parameters of the BERT-based model.

3.4 Classification

After label and document representation learnings, these outputs are combined to generate informative features for classification. As shown in Fig. 1, each column vector of the label embedding matrix represents information of a label. These label embedding vectors are then mapped into the document representation to make it more discriminative. Particularly, each label embedding vector H_i is added with the document embedding T' before being fed into the classifier. Formally, the i^{th} output column vector of the label-document embedding matrix L_i is expressed as

$$L_i = H_i + T' \tag{5}$$

After generating the output map $L \in \mathbb{R}^{c \times d'}$, we use a fully-connected layer $W \in \mathbb{R}^{d' \times 1}$ as a classifier. In preliminary experiments, we found that adding this classifier results in classification performance improvement compared to just using dot-product as [14]. As in [1], we adopt $Sigmoid$ as a nonlinear activation function $\sigma(.)$ which is applied on the last layer to make the final prediction. As suggested in [5], we share a weight matrix W across all labels in order to reduce the number of the proposed model's parameters. Consequently, the predicted output of the L_i label is yielded as below:

$$\hat{y}^i = \sigma(W^T L_i) \tag{6}$$

where L_i is the i^{th} column vector of the output features L. The whole network is trained by using a common loss function for multi-label classification, as expressed as (7):

$$\mathcal{L} = \sum_{i=1}^{c} (y^i log((\hat{y}^i)) + (1 - y^i) log(1 - (\hat{y}^i))) \tag{7}$$

where c is the number of labels, y^i is the ground truth label value, and $\hat{y}^i$ is the predicted probability value which is calculated as (6).

Input label: "wheat"
Output: "wheat is a grass widely cultivated for its seed, a cereal grain which is a worldwide staple food. The many species of wheat together make up the genus Triticum; the most widely grown is common wheat (T. aestivum)"

Fig. 3 An example of text extracted from an input label

4 Experiments

In this section, we conduct experiments to validate the effectiveness of the proposed model. To do that, we first describe datasets that are used in experiments. Then, we present the evaluation metrics, baselines, and settings used for experiments. Finally, we compare the empirical results with SOTA baselines and also analyze ablation studies to show the effectiveness of our method.

4.1 Datasets

For experiments, we use three benchmark datasets designed for multi-label text classification to evaluate the classification performance of our model:

- **Reuters-21578**[3] is one of the most widely used datasets for text classification. It is collected from the Reuters financial newswire service in 1987. In this work, we use ApteMod which is a subset of the Routers dataset for multi-label text categorization with 10,788 documents. This dataset version is highly skewed, with 36.7%

[3] https://www.kaggle.com/nltkdata/reuters

of the documents in the most common class, and only 0.0185% (2 documents) in each of the five least common classes.

- **Arxiv academic paper dataset(AAPD)** was built by [34] for multi-label text classification. It was collected from the abstract and the corresponding subjects in the computer science field from Arxiv.[4] It is a large dataset including 55,840 documents and 54 class labels in total. Each document may have multiple labels.
- **Reuters corpus volume I (RCV1)** [35] is a newswire collection of Reuter's News that was manually categorized for research purposes from 1996-1997. We use this dataset since it is large-scale, with over 800,000 documents. Furthermore, the number of examples for testing is much larger compared to training. This enables evaluating the generalization capability of the proposed model.

These datasets are carefully selected because they are widely used and large-scale, as described above. This enables us to verify the effectiveness of the proposed model. Furthermore, we attempt to use the original version of these datasets for making fair comparisons. In summary, the statistics of the datasets are presented in Table 2.

4.2 Evaluation metrics and baselines

4.2.1 Evaluation metrics

In our experiments, we adopt commonly-used metrics to measure the performance of the proposed method. Like other models for multi-label classification, we use the precision at top k ($P@k$) and the Normalized Discounted Cumulated Gains at top k ($nDCG@k$). Both metrics are defined according to the predicted score vector $\hat{y} \in \mathbb{R}^L$ and the ground truth label vector $y \in \{0, 1\}^L$ as expressed in (8), (9), and (10):

$$P@k = \frac{1}{k} \sum_{l \in rank_k(\hat{y})} y_l \tag{8}$$

$$DCG@k = \sum_{l \in rank_k(\hat{y})} \frac{y_l}{log(l+1)} \tag{9}$$

$$nDCG@k = \frac{DCG@k}{\sum_{l=1}^{\min(k, \|y\|_0)} \frac{1}{log(l+1)}} \tag{10}$$

where $rank_k(\hat{y})$ is the label indexes of the top k highest scores of the current prediction ($\hat{y}$), $\| \ y \ \|_0$ counts the number of relevant labels in the ground truth label vector y. Following prior works [12], we use top k = 1, 3, 5 for both $P@k$ and $nDCG@k$. These metrics are calculated for every single document and then averaged over all of them.

[4] https://arxiv.org/

Apart from $P@k$ and $nDCG@k$, we also use another standard evaluation metric to evaluate our work, *Micro-F1* [36]. This metric considers the overall precision and recall of all the labels, as defined in (11), (12), and (13):

$$Micro - F1 = \frac{2PR}{P + R} \tag{11}$$

$$Precision(P) = \frac{\sum_{t \in S} TP_t}{\sum_{t \in S} TP_t + FP_t} \tag{12}$$

$$Recall(R) = \frac{\sum_{t \in S} TP_t}{\sum_{t \in S} TP_t + FN_t} \tag{13}$$

where TP_t, FP_t, FN_t denote the true-positives, false-positives and false-negatives for the t^{th} label in label set S, respectively.

4.2.2 Baselines

To show the efficiency of the proposed model, we compare it with models that achieved state-of-the-art results on the selected datasets. In order to make fair comparisons, we follow two main rules for choosing the baselines: i) we selected strong baselines that are evaluated on the same version of the datasets (i.e. the same statistics of the datasets as shown in Table 2); ii) we only reused the experimental results instead of reimplementing the baselines in order to keep their best settings and results as proposed. In addition, we also copy the results of some well-known models that are reimplemented and then evaluated on the selected datasets.

Document-based methods

- **XML-CNN** [1] combines the strengths of existing CNN methods and takes multi-label co-occurrence patterns into account. We also copy experiment results of strong deep learning models which were evaluated in this work, including SLEEC, FastXML, Bow-CNN, and Kim-CNN.
- **HTTN** [6] proposes a head-to-tail network to transfer the meta-knowledge from rata-rich tail labels to data-poor tail labels. The model takes advantage of sufficient information among head labels and label dependency between head labels and tail labels.
- **DocBERT** [32] reports SOTA results for document classification by simply fine-tuning the BERT model. This work is selected for comparison because we also adopt a BERT-based model for document representation. We also copy experiment results of CNN and RNN based models which were evaluated in this work, including Kim-CNN, XML-CNN, HAN, LSTM, and KD-LSTM.
- **VLAWE** [37] is a model based on a combination of a document representation based on aggregating

Table 2 Statistics of the datasets

Datasets	D	N	V	M	L	$\hat{L}$	$\tilde{L}$
Reuters-21578	10,788	6,604	1,165	3,019	90	1.23	132.07
AAPD	55,840	46,614	8,226	1,000	54	2.41	2444.0
RCV1	804,414	19,677	3,472	781,265	103	3.18	729.67

D is the number of documents, N is the number of training documents, V is the number of validating documents, M is the number of testing documents, L is the number of class labels, $\hat{L}$ is the average number of labels per document, $\tilde{L}$ is the average number of documents per label

word embedding vectors into document embeddings and SVM as a classifier.

Document-label-based methods

- **LSAN** [12] proposes a label-specific attention network for multi-label text classification. It uses label embeddings to explicitly compute the semantic relations between document words and labels. In this work, the authors also evaluated other strong deep learning methods: DXML considering the label structure from the label co-occurrence graph, SGM applying a sequence generation model from input documents to output labels, AttentionXML building the label-aware document representation, and EXAM exploiting the label text to learn the interaction between words and labels. We also included these methods in our comparison.
- **AttentionXML** [5] is a deep model that is based on a multi-label attention mechanism for capturing the most relevant part of the text to each label, and a label tree allowing to handle millions of labels.
- **HA-Seq2Seq** [8] is a sequence-to-sequence-based model. It introduces history-based label attention to effectively explore informative representations for predicting labels in multi-label text classification.
- **HCSM** [9] is a hierarchical cognitive structure learning model composed of the Attentional Ordered Recurrent Neural Network (AORNN) and Hierarchical Bi-Directional Capsule (HBiCaps). Both modules use a global hierarchical label structure to improve classification performance.
- **HE-AGCRCNN** [21] is a hierarchical taxonomy-aware and attentional graph capsule recurrent CNNs framework for large-scale multi-label text classification. The model uses hierarchical label-dependencies among labels to improve classification accuracy and reduce computational complexity.
- **HG-Transformer** [10] is a hierarchical graph transformer-based deep learning model for large-scale multi-label text classification. In this work, the representation of labels is generated using hierarchical dependencies among labels.
- **LAHA** [13] is based on hybrid attention to exploiting labels for document representation. The model includes three parts: a multi-label self-attention mechanism to detect the contribution of each word to labels, the representation of label structure and document content, and an adaptive fusion method for classification.
- **MAGNET** [24] is a graph attention network-based model. It was proposed to capture the attentive dependency structure among labels by using feature and correlation matrices. Besides, the model uses a BiLSTM to extract text features.

We also select and extract the evaluation results of other competitive models that were reported in the works mentioned above. It should be mentioned that the DocBERT model [32] reported results of base and large BERT versions on both Reuters-21578 and AAPD datasets. In our model, we also use BERT_{base} based model (RoBERTa-base). Therefore, we only compare our model with DocBERT (base version) to make a fair comparison.

4.3 Experimental settings

For our model, we used only one GCN layer for learning label representation. By doing experiments, we found that the model achieved the optimal results (as presented in Section 4.5). We set the embedding size of the GCN layer as 768, after doing experiments with different settings. We adopt RoBERTa-base for generating document representations. We employ a fully-connected neural network (768 input neurons and one output neuron), with its weights W that are shared across all labels, and with Dropout (0.2 rate). We train our model with a batch size of 16 and Adam optimizer with Weight decay (0.01 excluding bias and LayerNorm). We also apply an initialized learning rate of $5e-5$ with a linear schedule.

4.4 Results

Performance evaluation on reuters-21578 We first conduct the performance evaluation on the Reuters-21578. This dataset has the smallest number of documents among the selected datasets, but it is highly skewed. Table 3 reports

Table 3 Results on Reuters-21578 in terms of Micro-F1. Boldface indicates the best method while the underlined one is the second best

Dataset	Models	Micro-F1 (%)
Reuters-21578	Kim-CNN^{+}	80.8
	XML-CNN^{+}	86.2
	HAN^{+}	85.2
	LSTM^{+}	87.0
	KD-LSTM^{+}	88.9
	DocBERT^{+}	89.0
	VLAWE	89.3
	MAGNET	<u>89.9</u>
	LR-GCN (Ours)	**91.6**

Symbol $^{+}$ indicates results are extracted in [32]. The upper group is document-based models while the other for document-label-based ones

evaluation results of different models on Reuters-21578 in terms of Micro-F1. In this evaluation, we use this metric for comparison because it is reported in many works that were evaluated on this dataset. Furthermore, we do not compare classification performance in terms of $P@k$ and $nDCG@k$ since these results are not reported in these baselines.

As shown in the table, our model outperforms the others. The proposed model achieved a better result of 1.7% compared to MAGNET which is the second-best. This may come from several significant differences between them. First, MAGNET built the correlation matrix in the form of conditional probability as proposed in [14]. However, this form of calculation seems to be only suitable for capturing natural dependencies of object labels in images as explained in [14]. In contrast, our work models label correlation in a modified form of point-wise mutual information which is widely used for capturing word-word association in natural language processing [15]. Consequently, the proposed correlation matrix benefits from our model does not require additional fine-grained schemes such as attention for GCN as in MAGNET or re-weighted as in [14]. Second, our model benefits from RoBERTa which was confirmed in many works [31, 33] to be effective in extracting fine-grained document information, whereas MAGNET applied BiLSTM to learn word embeddings. The experimental results confirmed the effectiveness of the proposed model.

Surprisingly, VLAWE [37] is a combination of a document representation based on aggregating word embedding vectors into document embeddings and SVM as a classifier is slightly better than DocBERT (the base version) [32]. It should be mentioned that [32] also reported Micro-F1 of 90.70% when running BERT$_{large}$ on this dataset. While this result is higher than both VLAWE and MAGNET, it is lower than ours which only adopts a base version of the BERT-based model (i.e. RoBERTa base). This validates the effectiveness of the proposed model using GGN for capturing both label correlation and semantics.

Performance evaluation on AAPD Next, we evaluate the performance of the proposed model on the AAPD dataset. For Micro-F1, our model once again achieves better results compared to DocBERT [32] which achieved the second best, as shown in Table 4. In terms of $P@k$ and $nDCG@k$, our model achieves the best results in almost all metrics compared to the other works, as shown in Table 5. LSAN outperforms other methods (lines 1^{st}-11^{th} in Table 5) on AAPD. However, its results are lower than our work in almost all metrics, except for $P@5$. It should be noticed that we evaluated our model on this dataset without using label embeddings because word labels in this dataset are not available. As shown in Section 4.5, we argue that we can improve the performance of the proposed model by using the proposed label embedding method. One possible reason is that our model benefits from the correlation and semantics information of labels that are learned by GCN.

Performance evaluation on RCV1 Finally, we verify the effectiveness of the proposed model by evaluating it on the

Table 4 Results on AAPD and RCV1 in terms of Micro-F1

Dataset	Models	Micro-F1 (%)
AAPD	Kim-CNN^{+}	51.40
	XML-CNN^{+}	68.70
	HAN^{+}	68.00
	LSTM^{+}	70.50
	KD-LSTM^{+}	72.90
	DocBERT^{+}	<u>73.40</u>
	AttentionXML‡	71.50
	HA-Seq2Seq‡	72.00
	LR-GCN (Ours)	**74.03**
RCV1	RCNN$^{\div}$	68.60
	XML-CNN$^{\div}$	69.50
	HAN$^{\div}$	69.60
	HLSTM$^{\div}$	67.30
	HCSM$^{\div}$	<u>85.80</u>
	HR-DGCNN-3$^{\circledast}$	76.20
	HE-AGCRCNN$^{\circledast}$	77.80
	LR-GCN (Ours)	**88.03**

For each dataset, boldface indicates the best method while the underlined one is the second best. Results are extracted: $^{+}$ in [32], ‡ in [8], $^{\circledast}$ in [21], $^{\div}$ in [9]. For each dataset, the upper group is document-based models while the other for document-label-based ones

Table 5 Results on AAPD and RCV1 in terms of $P@k$ and $nDCG@k$

Datasets	Models	P@1(%)	P@3(%)	P@5(%)	nDCG@3(%)	nDCG@5(%)
AAPD	XML-CNN*	74.38	53.84	37.79	71.12	75.93
	HTTN$^{\times}$	83.84	59.92	40.79	79.27	82.67
	DXML*	80.54	56.30	39.16	77.23	80.99
	SGM*	75.67	56.75	35.65	72.36	75.35
	AttentionXML*	83.02	58.72	40.56	78.01	82.31
	EXAM*	83.26	59.77	40.66	79.10	82.79
	LSAN*	85.28	61.12	**41.84**	80.84	84.78
	SLEEC†	81.96	57.48	38.99	77.65	81.59
	LAHA†	84.48	60.72	41.19	80.11	83.70
	LR-GCN (Ours)	**86.50**	**62.43**	41.66	**82.52**	**85.48**
RCV1	Bow-CNN$^{\perp}$	96.40	81.17	56.74	92.04	92.89
	Kim-CNN$^{\perp}$	93.54	76.15	52.94	87.26	88.20
	XML-CNN$^{\perp}$	96.86	81.11	56.07	92.22	92.63
	HTTN$^{\times}$	95.86	78.92	55.27	89.61	90.86
	DXML*	94.04	78.65	54.38	89.83	90.21
	SGM*	95.37	81.36	53.06	91.76	90.69
	AttentionXML*	96.41	80.91	56.38	91.88	92.70
	EXAM*	93.67	75.80	52.73	86.85	87.71
	LSAN*	96.81	81.89	56.92	92.83	93.43
	SLEEC$^{\perp}$	95.35	79.51	55.06	90.45	90.97
	FastXML$^{\perp}$	94.62	78.40	54.82	89.21	90.27
	HR-DGCNN$^{\div}$	95.29	50.32	55.38	90.02	90.28
	HG-Transformer$^{\div}$	95.80	80.98	55.96	90.03	91.96
	LR-GCN (Ours)	**97.13**	**84.29**	**58.45**	**94.98**	**95.38**

For each dataset, boldface indicates the best method while the underlined one is the second best. Results are extracted: * in [12], † in [13], $^{\times}$ in [6], $^{\perp}$ in [1], $^{\div}$ in [10]. For each dataset, the upper group is document-based models while the other for document-label-based ones

RCV1 dataset. A challenge of this dataset is that it is a large-scale dataset in which training samples are much fewer than testing samples. It should be noticed that although work in [8] (also in many other works) reported its results on RCV1 dataset, it uses almost all of the samples for training (i.e. 802,414/1,000/1,000 for training/testing/validation) instead of using the same split as the original dataset. Therefore, we skip reporting these results for fair comparisons.

For RCV1, our model outperforms all prior SOTA methods in all metrics. In terms of Micro-F1 the proposed model outperforms HCSM [9] (the second-best) and HE-AGCRCNN [21] by large margins of 2.23% and 10.23%, respectively. It should be mentioned that HE-AGCRCNN outperformed many traditional (e.g. HR-LR, HR-SVM) and deep learning models (e.g. HR-DGCNN-3, Capsule-B), as presented in [21]. In terms of $P@k$ and $nDCG@k$, our model is superior with a large margin compared to LSAN which keeps showing better results compared to deep-learning models and even transformer-based models in almost all metrics (i.e. except for $P@1$). XML-CNN follows LSAN with a tiny gap. These show the effectiveness of the proposed model for dealing with the multi-label classification problem.

As mentioned above, while both our model and LSAN use document and label information, our method outperforms LSAN in all performance metrics. There are two possible reasons for this: (1) LSAN did not use label correlation. This model only used label semantics to explicitly determine the semantic relation between each word-label pair via an attention mechanism. In contrast, apart from considering the label semantics, our model benefited from label correlations, in which the proposed label correlation matrix is built to guide the information propagation among nodes in the graph. This further confirms our idea that using GCN to learn over the proposed correlation matrix is helpful for representing label information; (2) while LSAN adopted Bi-LSTM for input text representation, we employed RoBERTa which has been shown as a powerful tool for extracting fine-grained document information [31, 33].

4.5 Ablation studies

In this section, we investigate other aspects of the model, including the effects of the FCNN component, the correlation matrix, node embedding methods, sensitive parameters of GCN, and the size of training data. These studies are conducted on the Reuters-21578 dataset because of available label words and the faster training and testing time.

4.5.1 Effects of the fully-connected neural network component

As mentioned above, one of our improvement is that we add a fully-connected neural network on top of the model as a classifier. In this study, we test the proposed work without this component to measure its contribution to classification performance. Particularly, each label embedding vector H_i will be combined using dot-product with the document embedding T' to produce the predicted score, as applied in other works [14, 24]. As shown in Fig. 4, the proposed model (LR-GCN) is superior compared to the model with dot-product, especially 2.1% in terms of Micro-F1. One of the reasons for this may be that embeddings of labels and the document are not good enough to directly output the predicted score. They demand a subnet to continuously learn their representations and predict final results.

4.5.2 Effects of different correlation matrix methods

We first evaluate our model with different edge-weight calculation methods that are used for building the correlation matrix. This enables us to show the effectiveness of the proposed correlation matrix. In this study, we consider another correlation matrix, named Edge=1, where the edge weight of two co-occurrence label nodes is assigned by 1. As shown in Fig. 5, the proposed correlation matrix shows an improvement compared to the other. One possible reason is that $A_{i,j}$=1 for every pair of co-occurrence label nodes is not enough for representing label correlation. This confirms the role of the proposed correlation matrix in propagating label information among nodes in GCN.

4.5.3 Effects of node embedding methods

In our work, we propose to use external language resources to enrich label word embeddings. In this study, we conduct experiments on the proposed model under other node embedding methods. To do that, we first test the proposed model without using node features, named None (i.e. feature matrix $X = I$ as an identity matrix). After that, we further investigate the performance of the proposed model under popular word embedding methods such as Word2vec [26], Glove [27], FastText [28]. Figure 6 shows two main points: (1) the proposed node embedding with enrichment shows better results compared to the others. This confirms the idea of using external information for enriching label node embeddings and also enables thinking of applying this idea for other applications such as specific domains (e.g. medical and legal documents) and low resource languages (e.g. Vietnamese, Japanese) where powerful embeddings may not be available; (2) The evaluation results also confirm that using label embeddings as node features improves performance classification.

4.5.4 Effects of sensitive parameters of GCN

In this study, we first evaluate our model under various GCN layer numbers including one layer with an embedding dimension of 768, two layers with embedding dimensions of 768 and 1024, three layers with embedding dimensions of 768, 1024, 1024, and four layers with embedding dimensions of 768, 1024, 1024, 2048. As shown in Fig. 7a, the model achieves optimal results with only one GCN layer while increasing the layer number leads to degradation of classification performance. This may come from that we use the powerful word embedding method that is trained on a large text corpus to generate node embeddings. After that,

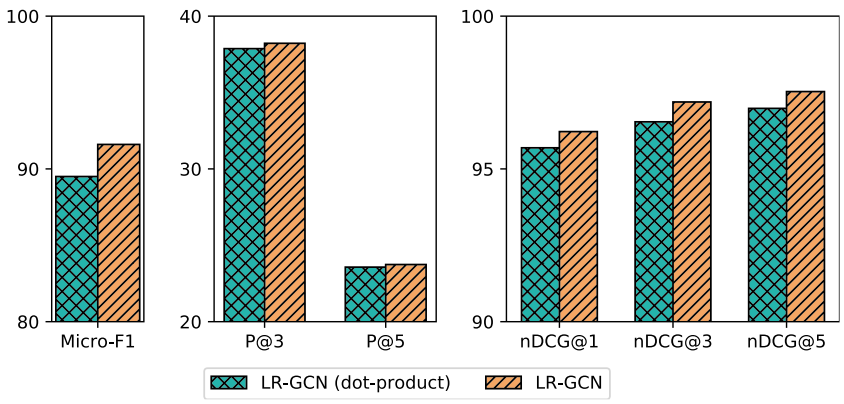


Fig. 4 Comparison of different methods for building the classifier

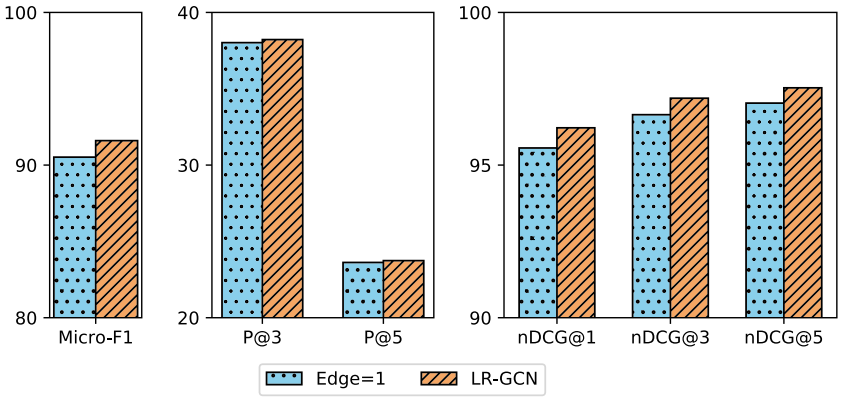


Fig. 5 Comparison of different methods for building the correlation matrix

the embedding of every single node in the graph will be gradually accumulated from its intermediate neighborhoods when increasing the number of GCN layers. This leads to being over smoothing. This performance drop is consistent with results that are mentioned in [25] and [15].

We then vary the GCN embedding dimension and measure the classification performance. As shown above, our model achieved good results with only one GCN layer. In this study, we perform the model with one layer under varying embedding dimensions of 256, 512, 768, 1024, and 2028. As shown in Fig. 7b, the model reaches optimum performance at the dimension of 768. Furthermore, we can observe that too low dimensional embeddings may not propagate label information to the whole graph well, while highly dimensional embeddings do not improve performance.

4.5.5 Effects of the size of training data

Finally, we test the proposed work with different training data proportions to explore the sensitivity of our correlation matrix construction as well as the model. In this study, we also evaluated several best-performing models including XML-CNN, AttentionXML, and LSAN. For these baselines, we keep the settings as presented in their papers. Only for LSAN, we used Word2vec for word embeddings due to lacking pre-trained embeddings provided in its source. Figure 8 compares evaluation results of the models with data proportions of 0.05, 0.10, 0.25, 0.50, and 0.75. As shown in the figure, our model achieves competitive results compared to the other ones. Especially, the proposed model is consistently higher than the baselines at lower data percentages (less than 0.25). This means that the proposed correlation matrix as well as our model are not sensitive to training data proportions and can be applied in practical cases, in which only limited training samples are provided.

5 Conclusion

In this paper, we proposed a new model for multi-label text classification. Our model explores label correlation and semantics by using graph convolutional networks. To do that, we design an effective correlation matrix that is based on the occurrence and co-occurrence probabilities of labels. For node features, we enrich label embeddings using

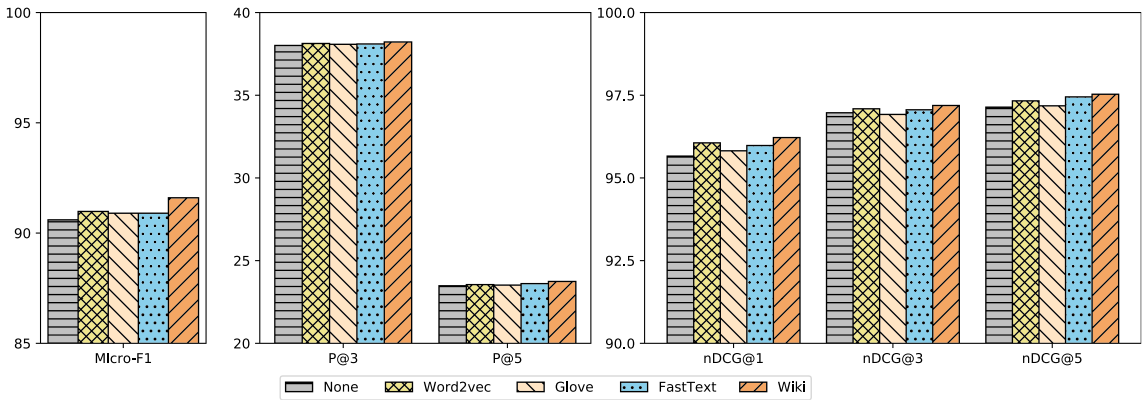
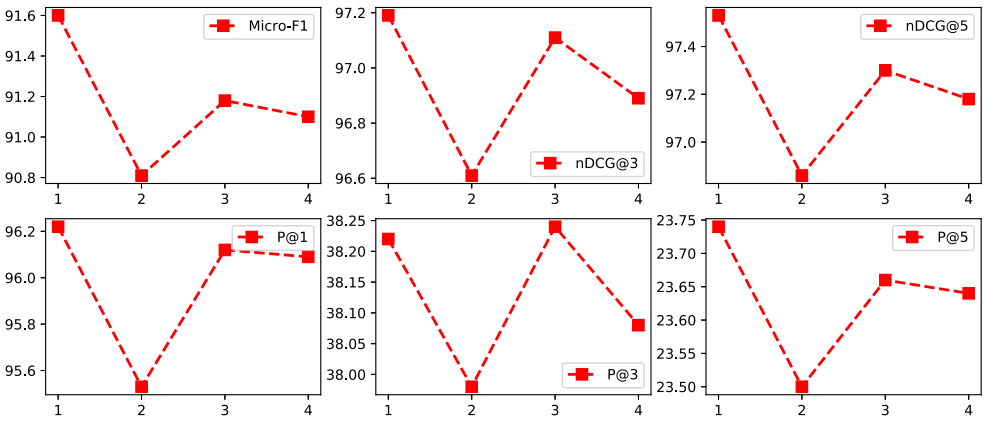
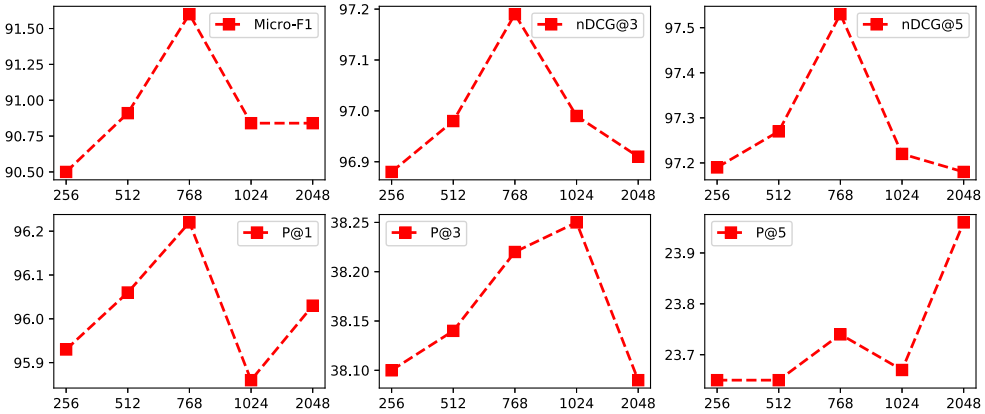


Fig. 6 Comparison of different node embedding methods

Fig. 7 Test parameter sensitivity of GCN



(a) Test performance (%) by varying GCN layer numbers (i.e. 1, 2, 3, 4)



(b) Test performance (%) by varying the GCN embedding dimension (i.e. 256, 512, 768, 1024, 2048)

external language resources. Correlated label information learned from GCN is combined with fine-grained document representation generated from a sub-net for classification. Experimental results show three important points. Firstly, our model outweighs prior state-of-the-art methods. This validates the effectiveness of the proposed model. Secondly,

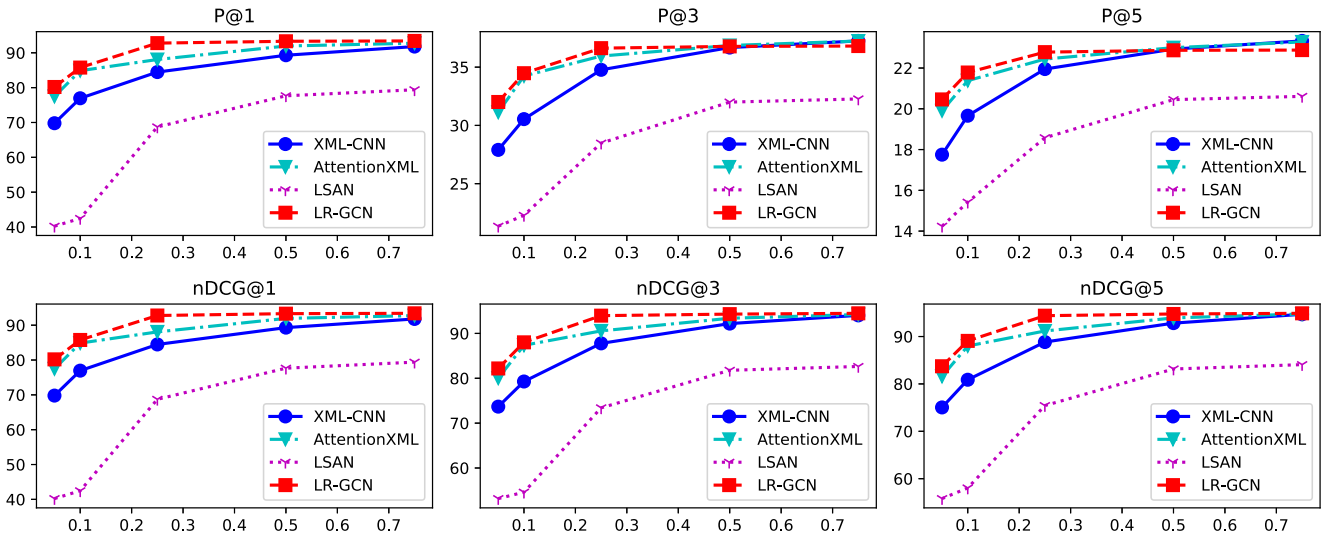


Fig. 8 Test performance (%) by varying training data proportions

the proposed correlation matrix and enriching node embeddings improve classification performance. Finally, increasing the number and dimensionality of GCN layers does not improve the quality of classification.

Acknowledgment This research is funded by Hung Yen University of Technology and Education under grant number UTEHY.L.2020.08.

References

1. Liu J, Chang W-C, Wu Y, Yang Y (2017) Deep learning for extreme multi-label text classification. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval. Association for computing machinery, SIGIR '17, pp 115–124. https://doi.org/10.1145/3077136.3080834
2. Tang P, Jiang M, Xia BN, Pitera JW, Welser J, Chawla NV (2020) Multi-label patent categorization with non-local attention-based graph convolutional network. Proc AAAI Conf Artificial Intell 34(05):9024–9031. https://doi.org/10.1609/aaai.v34i05.6435
3. Huang B, Guo R, Zhu Y, Fang Z, Zeng G, Liu J, Wang Y, Fujita H, Shi Z (2022) Aspect-level sentiment analysis with aspect-specific context position information. Knowl-Based Syst 243:108473. https://doi.org/10.1016/j.knosys.2022.108473
4. Liu W, Wang H, Shen X, Tsang I (2021) The emerging trends of multi-label learning. IEEE Trans Pattern Anal Mach Intell, pp 1–1, https://doi.org/10.1109/TPAMI.2021.3119334
5. You R, Zhang Z, Wang Z, Dai S, Mamitsuka H, Zhu S (2019) Attentionxml: label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In: Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, neurIPS 2019, december 8-14, 2019, vancouver, BC, Canada, pp 5812–5822
6. Xiao L, Zhang X, Jing L, Huang C, Song M (2021) Does head label help for long-tailed multi-label text classification. Proc AAAI Conf Artificial Intell 35(16):14103–14111
7. Peng H, Li J, He Y, Liu Y, Bao M, Wang L, Song Y, Yang Q (2018) Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In: Proceedings of the 2018 World Wide Web Conference. International world wide web conferences steering committee, WWW '18, pp 1063–1072
8. Xiao Y, Li Y, Yuan J, Guo S, Xiao Y, Li Z (2021) History-based attention in seq2seq model for multi-label text classification. Knowl-Based Syst 224:107094. https://doi.org/10.1016/j.knosys.2021.107094
9. Wang B, Hu X, Li P, Yu PS (2021) Cognitive structure learning model for hierarchical multi-label text classification. Knowl-Based Syst 218:106876. https://doi.org/10.1016/j.knosys.2021.106876
10. Gong J, Teng Z, Teng Q, Zhang H, Du L, Chen S, Bhuiyan MZA, Li J, Liu M, Ma H (2020) Hierarchical graph transformer-based deep learning model for large-scale multi-label text classification. IEEE Access 8:30885–30896. https://doi.org/10.1109/ACCESS.2020.2972751
11. Cai L, Song Y, Liu T, Zhang K (2020) A hybrid bert model that incorporates label semantics via adjustive attention for multi-label text classification. IEEE Access 152183-152192:8
12. Xiao L, Huang X, Chen B, Jing L (2019) Label-specific document representation for multi-label text classification. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). Association for computational linguistics, pp 466–475, https://doi.org/10.18653/v1/D19-1044, https://www.aclweb.org/anthology/D19-1044
13. Huang X, Chen B, Xiao L, Yu J, Jing L (2021) Label-aware document representation via hybrid attention for extreme multi-label text classification. Neural Process Letters
14. Chen Z-M, Wei X-S, Wang P, Guo Y (2019) Multi-label image recognition with graph convolutional networks. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR). pp 5172–5181, https://doi.org/10.1109/CVPR.2019.00532
15. Yao L, Mao C, Luo Y (2019) Graph convolutional networks for text classification. In: In 33rd AAAI conference on artificial intelligence (AAAI-19), pp 7370–7377
16. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: a robustly optimized bert pretraining approach. In: arXiv:1907.11692
17. Tsoumakas G, Vlahavas I (2007) Random k-labelsets: an ensemble method for multilabel classification. In: Kok JN, Koronacki J, Mantaras RLD, Matwin S, Mladenič D, Skowron A (eds) Machine Learning: ECML 2007. Springer, pp 406–417
18. Zhang M-L, Zhou Z-H (2007) Ml-knn: a lazy learning approach to multi-label learning. Pattern Recogn 40(7):2038–2048
19. Kalchbrenner N, Grefenstette E, Blunsom P (2014) A convolutional neural network for modelling sentences. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (Vol 1: long papers). Association for computational linguistics, Baltimore pp 655-665
20. Tai KS, Socher R, Manning CD (2015) 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 (Vol 1: long papers). Association for computational linguistics, pp 1556–1566. https://doi.org/10.3115/v1/P15-1150, https://www.aclweb.org/anthology/P15-1150
21. Peng H, Li J, Wang S, Wang L, Gong Q, Yang R, Li B, Yu PS, He L (2021) Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification. IEEE Trans Knowl Data Eng 33(6):2505–2519
22. Wang G, Li C, Wang W, Zhang Y, Shen D, Zhang X, Henao R, Carin L (2018) Joint embedding of words and labels for text classification. In: Proceedings of the 56th annual meeting of the association for computational linguistics (Vol 1: long papers). Association for computational linguistics, pp. 2321–2331. https://doi.org/10.18653/v1/P18-1216, https://aclanthology.org/P18-1216
23. Chai D, Wu W, Han Q, Wu F, Li J (2020) Description based text classification with reinforcement learning. In: III HD, Singh A (eds) Proceedings of the 37th international conference on machine learning. Proceedings of machine learning research. PMLR, vol 119, pp 1371–1382, https://proceedings.mlr.press/v119/chai20a.html, https://dl.acm.org/doi/10.5555/3524938.3525066, Accessed 23 March 2022
24. Pal A, Selvakumar M, Sankarasubbu M (2020) Magnet: multi-label text classification using attention-based graph neural network. In: ICAART (2), pp 494–505
25. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: International conference on learning representations (ICLR)
26. Mikolov T, Chen K, Corrado Gs, Dean J (2013) Efficient estimation of word representations in vector space. Proc Workshop ICLR, vol 2013
27. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014

conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
28. Joulin A, Grave E, Bojanowski P, Douze M, Jégou H, Mikolov T (2016) Fasttext.zip: compressing text classification models. CoRR, arXiv:1612.03651
29. Biesialska M, Rafieian B, Costa-jussà MR (2020) Enhancing word embeddings with knowledge extracted from lexical resources. In: Proceedings of the 58th annual meeting of the association for computational linguistics: student research workshop. Association for computational linguistics, pp 271–278, https://doi.org/10.18653/v1/2020.acl-srw.36, https://aclanthology.org/2020.acl-srw.36
30. Narayan S, Cohen SB, Lapata M (2018) Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Association for computational linguistics, pp 1797–1807, https://doi.org/10.18653/v1/D18-1206. https://aclanthology.org/D18-1206
31. Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J (2021) Deep learning–based text classification: a comprehensive review. ACM Comput Surv, vol 54(3), https://doi.org/10.1145/3439726
32. Adhikari A, Ram A, Tang R, Lin J (2019) Docbert: Bert for document classification. In: arxiv:1904.08398
33. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: a robustly optimized bert pretraining approach. arXiv:1907.11692
34. Yang P, Sun X, Li W, Ma S, Wu W, Wang H (2018) SGM: Sequence generation model for multi-label classification. In: Proceedings of the 27th international conference on computational linguistics. Association for computational linguistics, pp 3915–3926, http://aclanthology.lst.uni-saarland.de/C18-1330.pdf, Accessed 23 March 2022
35. Lewis DD, Yang Y, Rose TG, Li F (2004) Rcv1: a new benchmark collection for text categorization research. J Mach Learn Res 5:361–397
36. Yang Y (1999) An evaluation of statistical approaches to text categorization. Inf. Retr 1(1–2):69–90. https://doi.org/10.1023/A:1009982220290
37. Ionescu RT, Butnaru A (2019) Vector of locally-aggregated word embeddings (VLAWE): a novel document-level representation. In: Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies, Vol 1 (long and short papers). Association for Computational Linguistics, pp 363–369, https://doi.org/10.18653/v1/N19-1033, https://www.aclweb.org/anthology/N19-1033

Huy-The Vu received a Ph.D. degree in computer science and engineering from The University of Aizu, Japan in 2019. His current research interests include natural language processing, deep learning, and neuromorphic systems. He is currently working as a lecturer at the Department of Computer Science, Hung Yen University of Technology and Education.

Dr. Minh-Tien Nguyen received a Ph.D. degree in Information Science from Japan Advanced Institute of Science and Technology, Japan in 2018. His research interests include text summarization, information extraction, natural language processing, and deep learning. He published nearly 50 scientific papers. He also served as reviewers of high-quality international conferences and journals. He was a co-organizer of the Special Session on Deep Learning for Intelligent Systems at the International Conference on Knowledge and Systems Engineering (KSE 2018 and 2019) and Special Session on Deep Learning and Applications for Industry 4.0 at the International Conference on Computational Collective Intelligence (ICCCI 2020). He is currently the lecturer at the Faculty of Information and Technology, Hung Yen University of Technology and Education, Vietnam and is a senior AI researcher at Cinnamon AI, Vietnam.

Van-Chien Nguyen is currently a computer science student at Hanoi University of Science and Technology. His research interests are focusing on Machine Learning, Natural Language Processing, and their intersection.

Dr. Van-Quyet Nguyen received the Ph.D degree in Computer Science from the Chonnam National University, South Korea, in 2019. He received the M.Sc in Computer and Information Science from the Hanoi University of Science and Technology (HUST), Ha Noi, Viet Nam, in 2013. He received the B.Sc degree in Information Technology from the Hung Yen University of Technology and Education, Hung Yen, Viet Nam, in 2009. His research interests are in Big Data Analytics, Big Graph Processing, Machine Learning, and Internet of Things. He has over 10 years research and working experience in Big Data analytics and its applications. He is currently a lecturer at the Hung Yen University of Technology and Education.

Dr. Van-Hau Nguyen received an Engineer's degree in Applied Informatics Mathematics, and Master's degree in Information Technology in 2003 and 2006, respectively, at Hanoi University of Science and Technology. In 2015, he obtained the Ph.D. degree in Computer Science from the Artificial Intelligence lab at Technische Universitaet Dresden, Germany. He is currently the director of the AI center at the Faculty of Information and Technology, Hung Yen University of Technology and Education, Vietnam. He has published more than 30 papers in international conferences and journals. His research interests include: Automated Reasoning, Machine Learning, Artificial Intelligence, and IoT.