2020 International Conference on Computational Performance Evaluation (ComPE), North-Eastern Hill University, Shillong, Meghalaya, India. July 2–4, 2020

A Hybrid Classification Approach using Topic Modeling and Graph Convolution Networks

Thoudam Doren Singh
Department of Computer Science and Engineering
National Institute of Technology Silchar
Silchar, India
thoudam.doren@gmail.com

Divyansha
Department of Computer Science and Engineering
National Institute of Technology Silchar
Silchar, India
divyansha1115@gmail.com

Apoorva Vikram Singh
Department of Electrical Engineering
National Institute of Technology Silchar
Silchar, India
singhapoorva388@gmail.com

Abdullah Faiz Ur Rahman Khilji
Department of Computer Science and Engineering
National Institute of Technology Silchar
Silchar, India
abdullahkhilji.nits@gmail.com

***Abstract*—Text classification has become a key operation in various natural language processing tasks. The efficiency of most classification algorithms predominantly confide in the quality of input features. In this work, we propose a novel multi-class text classification technique that harvests features from two distinct feature extraction methods. Firstly, a structured heterogeneous text graph built based on document-word relations and word co-occurrences is leveraged using a Graph Convolution Network (GCN). Secondly, the documents are topic modeled to use the document-topic score as features into the classification model. The concerned graph is constructed using Point-Wise Mutual Information (PMI) between pair of word co-occurrences and Term Frequency-Inverse Document Frequency (TF-IDF) score for words in the documents for word co-occurrences. Experimentation reveals that our text classification model outperforms the existing techniques for five benchmark text classification data sets.**

***Index Terms*—Graph Convolutional Network, Latent Dirichlet Allocation, Text Classification, Topic Modeling**

I. Introduction

With an exponential upsurge in the popularity of internet usage in recent years, loads of unstructured textual data has been made available. Due to this, text classification has taken a center stage in helping agencies and people organize this huge amount of data. Text classification also forms the skeleton for several advanced natural language processing based applications like topic labeling, sentiment analysis, news filtering, spam detection, etc.

A crucial step in text classification is feature extraction and feature representation. Conventional methods of feature representation involve the representation of text as hand-crafted features (e.g. n-grams and bag of words). Bag-of-words (BoW) operates on text considering them as unrelated collection of words. This results in failure of Bag-of-words (BoW) in encoding word order and capturing of syntactic information in word sequences. Recently, deep learning based methods have significantly outperformed these methods by taking into consideration, the sequential intelligence present in the textual data. The important deep learning based text representation methods include techniques like Recurrent Neural Network (RNN) [1], Long-Short Term Memory (LSTM) [2] and Convolutional Neural Network (CNN) [3] that are capable of appreciating underlying syntactic and semantic information in textual corpus. Alternatively, topic modeling has been extended as an effective feature representation technique for textual data. This is generally carried out by constructing a document-topic matrix and feeding it into the model like features.

Recently, graph embeddings or graph neural networks have mustered huge recognition. These graph neural networks have been successful in preserving the semantics and rich relational structures present in the textual data.

This research work aspires to design an extensive multi-class classification model by exploiting an efficient graph based neural network called Graph Convolutional Network (GCN) [4] and a topic modeling based approach Latent Dirichlet Allocation (LDA) [5]. We model a structured graph which comprises of documents and words as nodes by leveraging the text documents. An edge between a word node and a document node has been made using TF-IDF [6] scores while an edge between two different word nodes has been constructed by calculating Point-Wise Mutual Information (PMI) score [7]. The problem, thus, becomes a node classification problem instead of text classification problem.

On top of this, we use LDA to calculate document-topic distribution matrix which had been further used as secondary features for the task. The use of LDA as a feature extraction technique allows us to represent semantics in the data without significantly increasing the dimensionality of the feature matrix. The final enhanced feature matrix is then fed into the classification model for the classification.

Our technique attempts to facilitate text classification problems by enhancing the feature representation of the textual data. To this end, a graph generated from the text documents

is leveraged to create node (word or document) embeddings. Furthermore, topic modeling inspired method LDA has been used to statistically calculate topic distribution in every document and has been used as a feature. As shown in Fig. 1, the novelty of our work lies in using these two different features simultaneously through a novel classification model.

The rest of the paper is organized as the following. Section II discuss about the related works followed by the discussion on text graph convolutional neural network (Text GCN) and Latent Dirichlet Allocation (LDA) models in section III. The section IV discuss about the proposed model. Section V discuss about the experimental setup of the proposed model along with dataset used. Section VI gives the results and discussion on the finding. The paper is concluded with section VII.

II. Related Works

The conventional text classification methods principally focus on two aspects- using classification models and by feature engineering. The advantage of feature engineering is shown in [8] by making a comparative study on classification tasks using k-nearest neighbor classification, Support Vector Machine (SVM) and centroid based classification methods which utilize Latent Semantic Indexing for dimension reduction. A variation of LDA proposing a generative model called Latent Dirichlet Allocation Category Language Model was used in [9].

Graph neural networks have become widely popular [10]. Renowned neural networks like CNN have been generalized in [11] to work on structured graphs. Kipf and Welling introduced graph convolutional networks(GCN) [4], which was able to achieve state of the art results on benchmark datasets. GCN also shows promising results in various NLP tasks for encoding the sentences like machine translation [12] and relation classification [13]. Many recent studies explored the utility of GCN in text classification like in [14] and [15]. However, they considered a sentence or a document as a graph of word nodes. In contrast TextGCN [16], when constructing a corpus graph regarded both words and documets as nodes hence creating a heterogenous graph. SGCN [17] introduced a simplified GCN architecture by removing excess complexities inherited by GCN layers and empirically showed that the final FC layer shows comparable performance with its GCN counterpart.

Hierarchical Bayesian model that combines topic-based and bigram-based techniques to document modeling has been shown in [18] going beyond CBOW in LDA by introducing a hierarchical Dirichlet language model.A Author-Conference-Topic (ACT) model is introduced in [19]. This model utilizes the topic distribution to explore the inter-dependencies among authors, publication venues, and papers.

The work in [20] enhances the effectiveness of a classification model. It does so by diminishing the number of dimensions or features and trying to map the terms having semantic relationship into the same feature dimension yielding an 11.1% increase in F1 measure over the BOW model. Several other approaches have also been shown using probabilistic models like the one shown in [21]. They proposed a filter-based probabilistic feature selection method for text classification. They also showed that Distinguishing feature selector or DFS could be adapted to another pattern classification problems.

In our work, we use a combination of both the LDA model as well as the GCN approach for text classification.

III. Models

A. Text Graph Convolutional Neural Network (Text GCN)

A Graph Convolution Network (GCN) [4] is a neural network containing multiple layers that has ability to work directly on the graph to generate vector representations of nodes in the form of embeddings based on the edges with neighbor nodes.

The aforementioned graph is a knowledge representation process to schematically model word co-occurrences in various documents by the means of a formal framework facilitating the adaptation of graph convolutions. The graph formed consists of N_i nodes and $\overline{\mathrm{E}}$ edges where N_i corresponds to all the documents and the complete vocabulary present in them while $\bar{\mathrm{E}}$ refers to the weighted relationships between them. The weights are given by:

$$A_{lm} = \begin{cases} \mathrm{PMI}(l,m) & l,\ m\ are\ words, \mathrm{PMI}(l,m) > 0 \\ \mathrm{TF-IDF}_{lm} & l\ is\ document,\ m\ is\ word \\ 1 & l = m \end{cases} \tag{1}$$

The PMI value of a word pair l, m is computed as:

$$\mathrm{PMI}(l,m) = \log \frac{p(l,m)}{p(l)p(m)} \tag{2}$$

$$p(l,m) = \log \frac{p(l,m)}{p(l)p(m)} \tag{3}$$

$$p(l) = \frac{\#\psi(l)}{\#\psi} \tag{4}$$

where PMI refers to Point-Wise Mutual Information [7] calculated between any two co-occurring words. For calculation of PMI, a sliding window $\#\psi$ is used which has a length of 10 words.The total number of sliding window in the documents containing word l and both word l and m is represented by $\#\psi(l)$ and $\#\psi(l,m)$ respectively. $\#\psi$ is total of number of sliding windows in the corpus. A negative value of PMI indicates that no or very little semantic correlation between words and an edge is not created in that case. Conversely, a high positive value of PMI between a pair of words suggests high semantic correlation between them. The Term Frequency-Inverse Document Frequency (TF-IDF) [6] has been utilized to calculate weighted document-word edges.

Once the graph has been built, it has been fed into a k layer GCN (to convolute the features k number of times). The convoluted output achieved by k^{th} layer of GCN will have same feature size as that of label set which are then used as an input to some classification layer (softmax in this case). The

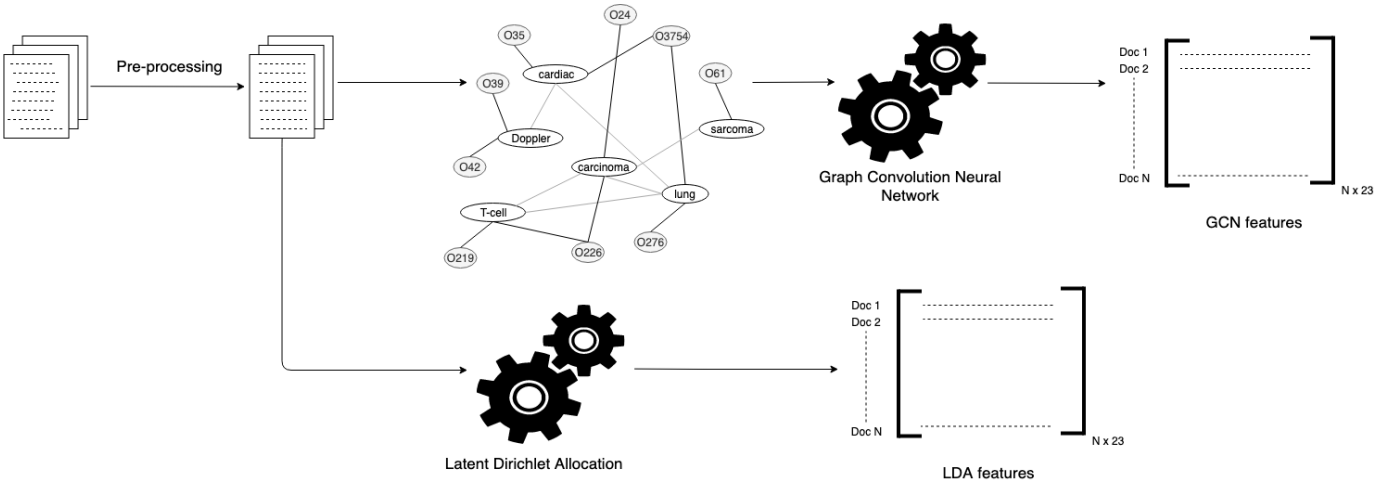

Fig. 1: Feature Extraction (Example taken from ohsumed corpus).

mathematical representation of a k layer Text GCN (where k = 2) can be shown as:

$$Z = \text{softmax}\left(\tilde{An}\,\text{ReLU}\left(\tilde{An}X\psi_0\right)\psi_1\right) \tag{5}$$

where,

$$\tilde{An} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \tag{6}$$

In the equations shown above, A refers to the adjacency matrix of the the addressed graph while $\tilde{An}$ is the representative of a normalized symmetric adjacency matrix for the same. D refers to the degree matrix of graph[1]. We take it to be a diagonal matrix (square matrix) which possesses number of dimensions equal to number of nodes in the graph. In simple terms, the input matrix is one hot encoded matrix of each node of the graph. It is noteworthy that all the diagonal elements in the adjacency matrix are ones since all the nodes are self-connected.

In this model, we use Simplified Graph Convolution Network (SGCN) [16], a special variant of GCN that tries to reduce the "excess complexity" associated with GCN. It tends to do so by demolishing the non-linearities from the model architecture by removal of non-linear transitions between GCN layers. Thus, it manages to collapse the final function in a single linear transformation.

B. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a model that has the ability to calculate the distribution of topics over a subjected text corpus making it a probabilistic model. The fundamental idea is that each and every document is treated as a combination of different topics where each topic is attributed by a set of words.

LDA is capable of generating feature vectors out of the documents by providing a document-topic matrix. Each column is representative of the topics that a document contains. The topics' distribution index on a document, therefore, serves the purpose of features of the document vectors. These are referred to as "Topic features". These topic features can represent the high-dimensional features present in a textual data i.e. words effectively in low dimensionality i.e. topics.

[1] $D_{ij} = \sum_i A_j$; ψ_1 and ψ_0 are the learnable filter weights for second and first layers respectively, that has to be trained. X represents the feature matrix of all the nodes to be given as input.

IV. The Proposed Model

In this work, we establish a novel technique to effectively classify the documents into classes based on their labels. To this end, we create semantic rich features by using Text GCN and topic modeling based approach LDA which are then fed them into a novel classification model, as shown in Fig. 1.

Text GCN adds a whole new dimension to the field of document classification by converting the text corpus into a graph and transforming the problem statement into node classification instead of text classification. It is highly proficient in capturing semantics present in the text by the means of the edges present between the nodes (documents and words). To further enrich the semantics captured, we make use of topic modeling approach LDA which is a distinguished probabilistic generative model exhibiting competent performances in numerous NLP based tasks. These semantics rich features are used to make the prediction of the label of concerned documents.

A novel classification model is utilized to undertake the classification task of the documents. The presented model uses a merge layer to combine the features generated using LDA and Text GCN. The merged features are fed into a dense layer that condenses the number of features while preserving the semantics contained in them. The number of features after being condensed is equivalent to the total number of labels of the documents present in the dataset. The output from this layer is passed through several other layers before getting fed to softmax which provides us with the probability distribution of all the labels for each individual document. Along with this, we use a skip-connection between last GCN layer and second dense layer. Also, skip-connections between every

TABLE I: Summary statistics of datasets

Dataset	Docs	Training	Texts	Words	Nodes	Classes
MR	10,662	7,108	3,554	18,764	29,4263	2
Ohsumed	7,400	3,357	4,043	14,157	21,557	23
R52	9,100	6,532	2,568	8,892	17,992	52
R8	7,674	5,485	2,189	7,688	15,362	8
20NG	18,846	11,314	7,532	42,757	61,603	20

consecutive layer after the first dense layer has been deployed. To prevent overfitting, we use dropout in every layer ensuring that the model trains without overfitting. A schematic view of the classification model can be seen in Fig. 1. We conduct elaborate experimentation to achieve the best performance of the classification model, the results of which are shown in Table II.

V. Experimentation

We conduct our experimentation on five widely used benchmark datasets: Ohsumed, R8 and R52 of Reuters 21578, 20-Newsgroups (20NG) and Movie Review (MR). A detailed analysis of datasets can be seen in Table I.

- Ohsumed corpus [22] is taken from MEDLINE database. Every document present in the dataset may have single or multiple associated labels out of the total 23 label classes. Since, we are interested in single-label based classification of the text, we eliminate documents belonging to multiple classes. This leaves us with 7,400 documents associated only to a single class that were used for the evaluation.
- R52 and R8 are parts of Reuters[2] 21578 dataset. R52 has 52 categories while R8 has 8 of them.
- 20-NG dataset[3] consists of 18,846 documents that are divided among 20 different categories.
- MR dataset[4] consists of movie reviews where each review includes one sentence. It is a dataset for binary sentiment classification.

The preprocessing of the data started off with tokenization of text. The words having a frequency less than 5 are removed along with the stopwords mentioned in the NLTK.

For generation of features using LDA, the number of topics are selected to be equal to the number of labels in the dataset. The parameters are tuned to achieve best results. It has been done by setting number of documents that are to be iterated through in every update is one i.e. online iterative learning has been used; in each training chunk, the number of documents to be used is 100. During training, the number of passes through the corpus is equal to 10 and minimum probability mandatory for a topic to be present in a document to be appreciated equals to 0. For the rest of the parameters, default settings have been used.

We compare our hybrid model with three baseline text classification models (LSTM, CNN, Bi-LSTM [23]) and two state of the art models (TextGCN and SGCN) in text classification task:

- **LSTM**: We use the last hidden state to represent the whole text. Word embeddings were not pre-trained for this model.
- **CNN**: We utilise Convolutional Neural Network without any pre-trained word embeddings.
- **Bi-LSTM**: Bi-directional LSTM model with pre-trained emebddings is used.
- **Text GCN**: A Graph Convolutional Network customized for text is used.
- **SGCN**: A simplified GCN model for removing unnecessary complexities was used.

For our classification model, we use one layer of text GCN to convolute over the text documents. The output yields a feature matrix having same number of features as number of labels. To incorporate semantic wealth of topic features into these features, we pass both of them through a custom merge layer that combines these features and outputs a feature matrix of size twice the number of labels. This output is then fed as input into a dense layer that condenses the feature matrix back to the size of number of labels. This matrix is passed through multiple dense layers before being used as an input to a softmax layer. Additionally, we use multiple skip-connections between different layers to make our model more efficient. Batch size of 128 has been used to feed the data into the model. LBFGS has been used to optimize the model. To avoid overfitting of the model, we tune our model for different values of dropout. To deal with class imbalance problems (as in 20-NG) in classification task, we also use focal loss [24] given by FL.

VI. Results and Discussions

From the experimentation conducted (refer to section V), it can be observed that our proposed approach has the potential to achieve robust text classification results. In the Table II,the highlighted accuracies are the maximum that were acheived on respective datasets. It can be observed that our model was able to give state of the art performances for all datasets except MR. MR dataset has movie reviews and has 2 classes. The documents are labeled with respect to their overall sentiment polarity making it a binary classification task. LDA is used for topic modeling and is only suitable when discrete topics exists in the collection of documents. Topic modeling cannot capture polarity and fails to generate good features for MR dataset giving poor accuracy. It can be, therefore, inferred that

[2]https://www.cs.umb.edu/ smimarog/textmining/datasets/

[3]http://qwone.com/ jason/20Newsgroups/

[4]http://www.cs.cornell.edu/people/pabo/movie-review-data/

TABLE II: Accuracy on document classification task.

Model	20NG	R8	R52	Ohsumed	MR
CNN	0.7693	0.9402	0.8537	0.4387	0.7498
LSTM	0.6571	0.9368	0.8554	0.4113	0.7506
BI-LSTM	0.7318	0.9631	0.9054	0.4927	**0.7768**
TextGCN	0.8634	0.9707	0.9356	0.6836	0.7674
SGCN	0.8851	0.9724	0.9402	0.6851	0.7593
Hybrid	0.8642	**0.9762**	0.9424	**0.6987**	0.4764
Hybrid-FL	**0.8921**	0.9749	**0.9431**	0.6982	0.4892

our method is only suitable when discrete topic exists in the documents and the classification is for those topics.

The advantage of focal loss is clearly visible for 20NG and R52 dataset where there was the problem of class imbalance. The hybrid model with focal loss gave accuracy as high as 0.8921 on 20NG. The hybrid model without focal loss showed relatively low accuracy of 0.8642 on same dataset. The performance of our model is accredited to the features generated using LDA [5] and Text GCN [16].Our technique is not only more robust than other methods but also yields more interpretable results due to the involvement of topic modeling.

VII. Conclusion

In this work, we design a novel text classification model engined on Text Graph Convolution Networks (Text GCNs) and Latent Dirichlet allocation (LDA). We construct a structured heterogeneous text corpus graph to transform text classification into a node classification problem. Our model has exhibited promising results for the text classification tasks by capturing word-document and word-word dependencies using weighted scores (PMI and TF-IDF). The classification model employed for the task is also a novel architecture and has the potential to harvest the features to yield robust results.

In the future work, various variants of GCN can be employed that have the capability to outperform vanilla GCN. On top of this, better techniques to combine the generated features can be employed which can boost the performance of the classification model.

Acknowledgment

The authors would like to thank Anubhav Sachan for providing assistance in setting up the experimental environment in PyTorch framework. The authors would also express their gratitude to TEQIP-III cell for providing support for the work.

References

[1] D. Servan-Schreiber, A. Cleeremans, and J. L. McClelland, "Learning sequential structure in simple recurrent networks," in *Advances in neural information processing systems*, 1989, pp. 643–652.

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

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

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

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[6] G. Salton, E. A. Fox, and H. Wu, "Extended boolean information retrieval," *Communications of the ACM*, vol. 26, no. 11, pp. 1022–1036, 1983.

[7] G. Bouma, "Normalized (pointwise) mutual information in collocation extraction," *Proceedings of GSCL*, pp. 31–40, 2009.

[8] H. Kim, P. Howland, and H. Park, "Dimension reduction in text classification with support vector machines," *Journal of Machine Learning Research*, vol. 6, no. Jan, pp. 37–53, 2005.

[9] S. Zhou, K. Li, and Y. Liu, "Text categorization based on topic model," *International Journal of Computational Intelligence Systems*, vol. 2, no. 4, pp. 398–409, 2009.

[10] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.

[11] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[12] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, and K. Sima'an, "Graph convolutional encoders for syntax-aware neural machine translation," *arXiv preprint arXiv:1704.04675*, 2017.

[13] Y. Li, R. Jin, and Y. Luo, "Classifying relations in clinical narratives using segment graph convolutional and recurrent neural networks (seg-gcrns)," *Journal of the American Medical Informatics Association*, vol. 26, no. 3, pp. 262–268, 2019.

[14] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data. arxiv (2015)," *arXiv preprint arXiv:1506.05163*, 2015.

[15] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, "Large-scale hierarchical text classification with recursively regularized deep graph-cnn," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1063–1072.

[16] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 7370–7377.

[17] F. Wu, T. Zhang, A. H. d. Souza Jr, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," *arXiv preprint arXiv:1902.07153*, 2019.

[18] H. M. Wallach, "Topic modeling: beyond bag-of-words," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 977–984.

[19] J. Tang, R. Jin, and J. Zhang, "A topic modeling approach and its integration into the random walk framework for academic search," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 1055–1060.

[20] W. Sriurai, "Improving text categorization by using a topic model," *Advanced Computing*, vol. 2, no. 6, p. 21, 2011.

[21] A. K. Uysal and S. Gunal, "A novel probabilistic feature selection method for text classification," *Knowledge-Based Systems*, vol. 36, pp. 226–235, 2012.

[22] W. Hersh, C. Buckley, T. Leone, and D. Hickam, "Ohsumed: an interactive retrieval evaluation and new large test collection for research," in *SIGIR'94*. Springer, 1994, pp. 192–201.

[23] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.