TextSGCN: Document-Level Graph Topology Refinement for Text Classification

Chen Li
Beihang University
lichen@act.buaa.edu.cn

Xutan Peng
The University of Sheffield
x.peng@shef.ac.uk

Hao Peng
Beihang University
penghao@buaa.edu.cn

Jianxin Li
Beihang University
lijx@act.buaa.edu.cn

Lihong Wang
CNCERT
wlh@isc.org.cn

Philip S. Yu
UIC
psyu@uic.edu

Abstract

Graph-based neural networks hold promises in encoding text. However, existing methods are limited to resort words to associate documents when constructing document-level graphs, which increases the parameter size while causing undesirable noise. In this paper, we propose a research plan to investigate the construction of a better document-level graph for enhanced free text representation learning. First, we consider the degree of similarity between documents from the perspective of semantic, syntactic, and sequential context to initialize a weighted graph. To promote smooth propagation of features/labels within the graph, we then attempt to refine the graph topology. On the one hand, we strengthen the features/labels propagation inside the local dense subgraph by generating supernodes automatically. On the other hand, we realize the global information exchange across clusters by introducing novel propagation highways. We will conduct extensive experiments on a variety of datasets, benchmarking the proposed method against traditional approaches which are based on document-word heterogeneous graphs. In addition, we also design various empirical studies to further discuss the validity and interpretability of our model.

1 Introduction

Text classification is one of the most fundamental tasks in natural language processing and text mining, and has been long attracting attention in the community. Its definition is straightforward, that is, assigning one or multiple labels are to a given text. In practice, due to its versatility, text classification is widely used in numerous tasks, such as sentiment classification [3], spam detection [1], and rumor/stance identification [29].

The first and essential step to achieve text classification is to encode the text. In recent years, deep learning models have shown advantages over traditional classification tasks in the way of reducing the burden of feature engineering. This is achieved as deep learning models learn mappings to represent input documents as low-dimensional vectors, then perform classification with neural networks. In order to better capture coherent semantic information from text sequences in the representation step, previous works most commonly utilize sequential models, e.g., convolutional neural network [12, 34, 4, 21], recurrent neural network [17, 33, 27], and the mixture of both [25]. Different from these sequential learning models, some graph-based models have very recently been proposed, which have attracted widespread attention and have been successfully applied to semi-supervised text classification tasks [13, 23].

Pre-registration workshop NeurIPS (2020), Vancouver, Canada.

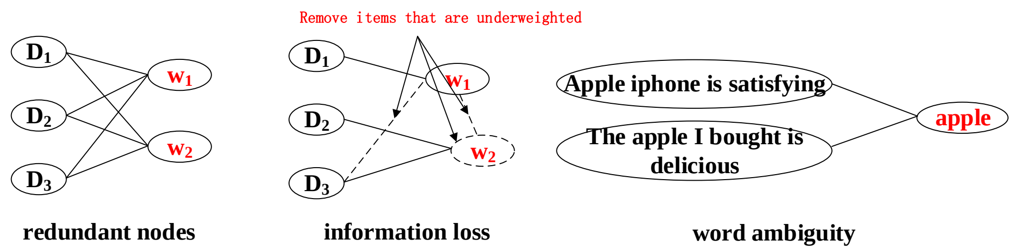

Figure 1: To handle the massive words in the dictionary, existing methods generally perform artificial constraint screening on word nodes when constructing document-word heterogeneous graphs. However, this operation cannot guarantee the quality of the remaining word nodes while inevitably cause information loss.

Compared with sequential learning models which handle inputs token-wise, graph-based methods directly process a large amount of text simultaneously by building graphs and utilizing Graph Neural Networks (GNNs). In this way, on the one hand, local and global features can be captured by modeling the relevance between documents. On the other hand, the scale of model parameters and the computing overhead can be effectively reduced. In practice, data from many real-world applications can be naturally cast into graphs, e.g., citation network and post propagation tree [30]. Nevertheless, a corpus composed of free text, which is the most common setting of text classification, cannot be converted into graphs directly. Existing workaround is to construct a heterogeneous graph by establishing and connecting the relationship between words/entities in the text. For example, [31] bootstrap a graph based on continuous contextual relationships and uses Graph Convolutional Network (GCN) [13] for text representation learning. [18] introduce semantic and syntactic information into the graph construction by including more text information and realizing intra-graph and inter-graph propagation. [32] further regard word and topic as the bridge for linking documents to diffuse information. However, using words or topics to bridge related text has limitations, as it can easily bring about noise and increase computational complexity. As shown in Figure 1, problems such as redundant nodes, information loss, and word ambiguity will affect the structure of graphs, and consequently, the propagation of labels/features.

To alleviate these issues, we propose a novel framework, namely TextSGCN, which constructs a document-level non-heterogeneous graph and employs GNNs for text representation learning and classification. More concretely, based on the integration of semantic, syntactic, and sequence contextual information, it attempts to build a graph to learn higher-quality text representation by better satisfying an assumption of rgraph-based semi-supervised learning [16, 28, 15]:

Assumption $\mathcal{A}$: nodes within the same dense subgraph tend to share similar features/labels.

Specifically, by considering semantic similarity, dependency syntax tree similarity, and sequence contextual similarity, TextSGCN first obtains a graph with document nodes and weighted edges only. After that, we design and implement a structure-based clustering algorithm to generate a supernode for each cluster (i.e., dense subgraph) and connect it to all nodes in the cluster. To further improve the diffusion efficiency of features/labels, we introduce a trainable discriminant function to connect similar supernodes and create highways between supernodes/clusters. Finally, by feeding the generated document-level graph to the subsequent GNNs, text classification can be achieved.

The highlight of our research plan is three-fold. (1) We propose a novel free-text-oriented graph construction method wjich combines semantic, syntactic, and sequence contextual information. (2) We design a new text representation framework for document-level graphs. (3) We will conduct extensive experiments to analyze the performance of different construction strategies for free text, with comprehensive in-depth empirical studies.

2 Related Work

Recently, compared with conventional feature engineering techniques and depth models, graph-based methods have attracted more attention in text classification. With the emergence and improvement

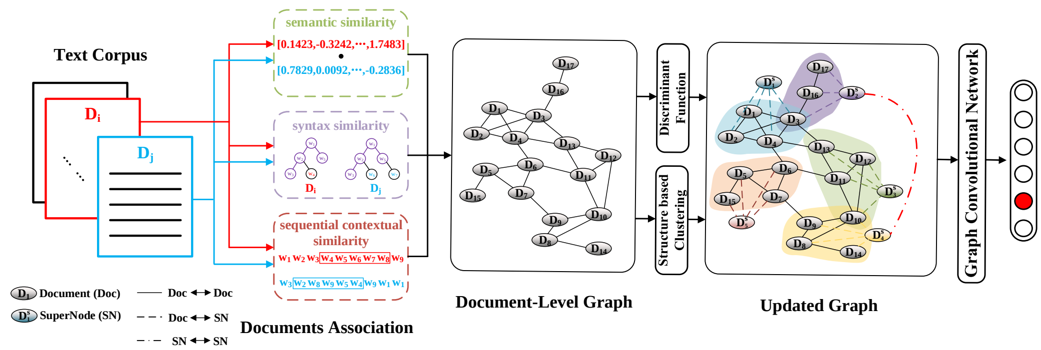

Figure 2: Overview of TextSGCN. The framework can be decoupled into three steps. The first step is to construct the initial document-level graph. Weights between document nodes are obtained based on pairwise calculation of the semantics, syntax, and sequential contextual similarity. The second step is to update the graph topology to make it more in line with Assumption $\mathcal{A}$. After applying a structure-based clustering algorithm on the initial graph, for each detected cluster, TextSGCN generates a supernode and associates all nodes within the cluster with it. On this basis, a trainable discriminant function is introduced to add new connections between supernodes. The final step is to feed the updated graph into the GNN for classification.

of GNNs, [2, 5, 13] have successively proposed semi-supervised text classification models by spreading features/labels smoothly along graph edges and achieve accurate text classification with limited samples. By introducing multi-head graph attention, [23] realize multi-channel learning with refined feature/label diffusion. More recently, by constructing document-word heterogeneous graphs, [31] extend the application of GNNs to text representation for free text. [18] further incorporate information such as semantics and syntax to fully capture local and global information through tensor learning. [32] attempt short text classification by using a heterogeneous graph consisting of documents, topics, and words. In addition, some methods [26, 9, 35] leverage GNNs to capture contextual information with text-level graphs, which also yield good results.

However, these methods only consider words or manually defined nodes as the bridges across related documents, which greatly increases the scale of the graph while introducing a lot of noise. Moreover, it brings about unnecessary calculation and space overhead. TextSGCN, in contrast, can overcome above limitations by refining graph topology for better satisfying Assumption $\mathcal{A}$ and thus promoting smooth propagation of features/tags locally and globally. In particular, if the aforementioned supernode is considered as a word/topic, existing works boil down to instantiated versions of our model. To the best of our knowledge, this is the first attmpt in this research direction.

3 Methodology

3.1 Document-Level Graph Construction

To more reasonably correlate documents in the corpus, we select three common types of information to compute the inter-document similarity, i.e., semantic, syntax, and sequential context.

Semantic similarity. Motivated by the capacity of Long Short-Term Memory (LSTM) [8] to capture semantic information for word representation, we propose a LSTM-based method to measure the similarity between documents. With a pretrained LSTM, we obtain the representation for each document and calculate cosine similarity between documents D_i and D_j as

$$\mathrm{sim(sem)} = \cos(\mathrm{LSTM}(D_i), \mathrm{LSTM}(D_j)). \tag{1}$$

Syntax Similarity. Dependency parsing is considered to be useful for guiding the understanding of sentences, so we apply dependency parsing to obtain the syntax tree of the sentences in the document. By calculating the similarity of the syntactic tree across documents (the overlap ratio of the *word-relation-word* triples in the statistical tree structure), we obtain the syntax similarity as

$$\mathrm{sim(syn)} = |T(D_i) \cap T(D_j)|\ |T(D_i) \cup T(D_j)|, \text{with } T(D_k) = \{t_{k,1}, t_{k,2}, \cdots, t_{k,n}\}, \tag{2}$$

where $t_{k,n}$ is a word-relation-word triple in dependency tree of D_k ($k \in \{i, j\}$).

Sequential contextual similarity. The sequential context describes the language characteristics of local co-occurrence between words, which has been widely used in text representation learning. In this study, to evaluate the sequential context similarity between two documents, we design a sliding-window-based calculation method as

$$\text{sim(seq)} = \sum \log(p(w_m, w_n)/p(w_m)p(w_n)), \tag{3}$$

where w_m, w_n are the words which simultaneously appear in both D_i and D_j. $p(w_m, w_n)$ is the probability of the word pair (w_m, w_n) co-occurring in the same sliding window, which can be estimated as the fraction of the total number of sliding windows over D_i and D_j and the number of times that w_i and w_j co-occur in the same sliding window.

Finally we obtain the weighted adjacency matrix A of the document-level graph G by normalizing the weighted adjacency matrices of three aforesaid similarities ($A_{\text{sim(sem)}}$, $A_{\text{sim(seq)}}$, $A_{\text{sim(syn)}}$), as

$$A = \text{Normalize}(A_{\text{sim(sem)}}) + \text{Normalize}(A_{\text{sim(syn)}}) + \text{Normalize}(A_{\text{sim(seq)}}), \tag{4}$$

where $\text{Normalize}(\cdot)$ is a normalization function.

3.2 Refining Graph

Structure-based clustering. The community detection algorithm can efficiently find dense subgraphs based on the graph structure. In the case of the document-level graph, we design an iterative overlapping clustering algorithm. It divides communities through a modularity-based non-overlapping community detection algorithm and then implements overlapping clustering by setting duplicates for the same nodes in different communities.

Inserting supernodes. To strengthen the density within each community (more in line with Assumption $\mathcal{A}$) and reduce the farthest distance in the cluster (i.e., to solv the long-distance dependence problem), we add a supernode for each cluster. The feature and labels of the supernode can be directly collected from the nodes in the cluster. In addition, in order to facilitate the propagation of features/labels among disconnected graphs/distant clusters (i.e., highways for propagation), we introduce a discriminant function to judge whether there should be edges between supernodes.

In the last step, the updated graph G' can be fed into subsequent GNN (in this work, we use GCN).

4 Experiment

In this section, we explain our plan for two groups of experiments. The first group is centered around text classification, which mainly focuses on the performance of TextSGCN under different settings and comparison against baseline methods. The second group is auxiliary experiments, focusing on some of the characteristics and module function in TextSGCN.

4.1 Datasets and Baselines

We plan to utilize widely used benchmark datasets to perform experiments and analysis, following previous studies [31, 18]. The benchmark corpora contain five text classification datasets: 20-Newsgroups dataset, Ohsumed dataset, R52 Reuters dataset, R8 Reuters dataset, and Movie Review dataset. These datasets involve many life genres, e.g., movie reviews, medical literature, news documents, etc. In addition, to investigate the gap between the graphs generated by TextSGCN and real-world graphs, we also select two popular citation datasets: Cora and Citeseer [30].

We divide our baseline methods into four categories. (1) Traditional feature engineering method, i.e., TF-IDF+LR. (2) Word embedding based models, such as PV-DBOW [14], PV-DM [14], fasttext [11], SWEM [22], and LEAM [24]. (3) Successful deep sequence models, such as CNN [12], LSTM [17], and Bi-LSTM [17]. (4) Graph-based methods (document-level), such as Graph-CNN-C [5], Graph-CNN-S [2], Graph-CNN-F [7], TextGCN [31], and TensorGCN [18].

4.2 Implementation

In our experiments, LSTM [10] and CoreNLP [20] will be used to perform semantic coding and syntactic parsing, respectively. For the realization of non-overlapping clustering, we select EgoSplitting [6] for its robustness. The hyperparameter settings in all methods are selected in the validation set and applied to the test set uniformly. To facilitate fair and comprehensive comparisons, we will report the variance and confidence interval of all the results.

4.3 Text Classification

From the perspective of information sources used to construct the graph, our framework can be divided into four variants: TextSGCN(sem), TextSGCN(syn), TextSGCN(seq), and TextSGCN(mixed). By comparing their individual classification performance on five datasets, we can obtain detailed ablation results and decide the best performing graph topology.

4.4 Graph Construction

Since graph topology is a direct constraint that guides GNN learning through label/feature propagation, it is necessary to specially discuss its role in classification.

First, we need to evaluate whether the constructed graph meets Assumption $\mathcal{A}$ as expected. That is, whether the dense subgraphs in the original document-level graph share similar features/labels and whether they are more similar in the updated graph (after inserting supernodes). We will calculate the similarity of the features/labels of the dense subgraph before and after the graph is updated (i.e., whether to insert supernodes). If the similarity of the updated graph is higher, it means that TextSGCN can make the graph more in line with Assumption $\mathcal{A}$.

Next, we will conduct a parallel experiment on a real-world graph (e.g., a citation network, where edges are reference relationships, and node features are bag-of-words based on scientific papers) to evaluate our model. The result interpetion will focus on comparing the gap between classification performance of GCN, TextGCN, TensorGCN, and TextSGCN. If TextSGCN outperforms or is on par with its counterparts with lower computational overhead, then we can claim that TextSFCN effectively associates documents and learns efficient text representation.

4.5 Discussion on Supernodes

Supernodes are an important bridge for TextSGCN to strengthen the relationship between nodes within a graph. We will perform T-SNE dimensionality reduction on the original nodes and the supernodes respectively, then compare the distribution of different types of nodes. Meanwhile, we will test the class separability of the original nodes and the supernodes (L1, L2) [19]. If the latter are more closely distributed than the former, the margin between different categories is larger, and the class separability is better, it is highly likely that the supernodes generated by TextSGCN can provide more stable labels/features during the propagation process.

5 Conclusion

In this proposal, we describe a novel document-level graph construction method and subsequent GNN model for text classification. The framework is designed to capture semantic, syntactic, and sequence contextual information while alleviating problems in document-word heterogeneous graphs, contributing to label/feature propagation, and reducing time and space complexity. We will conduct extensive experiments to verify the performance of different graph structures under the proposed framework, as well to obtain the correlation between different text information (i.e., semantics, syntax, context, and mixed) and text classification performance. In addition, our empirical studies will cover the interpretability of supernodes of the proposed TextSGCN.

Acknowledgement

This work is supported by the NSFC, China program (No. 61772151), and National Key R&D Program of China (No. 2018YFC0830804). We thank anonymous reviewers for their valuable help in improving this proposal.

References

[1] Seyed Ali Alhosseini, Raad Bin Tareaf, Pejman Najafi, and Christoph Meinel. Detect me if you can: Spam bot detection using inductive representation learning. In *Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 148–153, 2019.

[2] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[3] Yu Cao and Hua Xu. Satnet: Symmetric adversarial transfer network based on two-level alignment strategy towards cross-domain sentiment classification (student abstract). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13763–13764, 2020.

[4] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann LeCun. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 1107–1116, 2017.

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3837–3845, 2016.

[6] Alessandro Epasto, Silvio Lattanzi, and Renato Paes Leme. Ego-splitting framework: from non-overlapping to overlapping clusters. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 145–154. ACM, 2017.

[7] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[9] Lianzhe Huang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. Text level graph neural network for 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3442–3448, 2019.

[10] Ignacio Iacobacci and Roberto Navigli. Lstmembed: Learning word and sense representations from a large semantically annotated corpus with long short-term memories. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1685–1695, 2019.

[11] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 427–431, 2017.

[12] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751, 2014.

[13] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[14] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196, 2014.

[15] Chen Li, Xutan Peng, Hao Peng, Jianxin Li, Lihong Wang, and Philip S. Yu. Forming an electoral college for a graph: a heuristic semi-supervised learning framework. *CoRR*, abs/2006.06469, 2020.

[16] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3538–3545, 2018.

[17] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2873–2879, 2016.

[18] Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. Tensor graph convolutional networks for text classification. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8409–8416, 2020.

[19] Ana Carolina Lorena, Luís Paulo F. Garcia, Jens Lehmann, Marcílio Carlos Pereira de Souto, and Tin Kam Ho. How complex is your classification problem?: A survey on measuring classification complexity. *ACM Comput. Surv.*, 52(5):107:1–107:34, 2019.

[20] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60. The Association for Computer Linguistics, 2014.

[21] Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1063–1072. ACM, 2018.

[22] Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 440–450, 2018.

[23] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[24] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2321–2331, 2018.

[25] Ruishuang Wang, Zhao Li, Jian Cao, Tong Chen, and Lei Wang. Convolutional recurrent neural networks for text classification. In *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*, pages 1–6, 2019.

[26] Man Wu, Shirui Pan, Xingquan Zhu, Chuan Zhou, and Lei Pan. Domain-adversarial graph neural networks for text classification. In *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, pages 648–657, 2019.

[27] Jiacheng Xu, Danlu Chen, Xipeng Qiu, and Xuanjing Huang. Cached long short-term memory neural networks for document-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1660–1669, 2016.

[28] Liang Yang, Zesheng Kang, Xiaochun Cao, Di Jin, Bo Yang, and Yuanfang Guo. Topology optimization based graph convolutional network. In *IJCAI*, pages 4054–4061, 2019.

[29] Xiaoyu Yang, Yuefei Lyu, Tian Tian, Yifei Liu, Yudong Liu, and Xi Zhang. Rumor detection on social media with graph structured adversarial learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 1417–1423, 2020.

[30] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, pages 40–48, 2016.

[31] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7370–7377, 2019.

[32] Zhihao Ye, Gongyao Jiang, Ye Liu, Zhiyong Li, and Jin Yuan. Document and word representations generated by graph convolutional network and BERT for short text classification. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, pages 2275–2281, 2020.

[33] Shuhan Yuan, Xintao Wu, and Yang Xiang. Incorporating pre-training in long short-term memory networks for tweets classification. In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, pages 1329–1334, 2016.

[34] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657, 2015.

[35] Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. Every document owns its structure: Inductive text classification via graph neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 334–339, 2020.