Enhanced Representation Learning for Examination Papers with Hierarchical Document Structure

Yixiao Ma[1], Shiwei Tong[1], Ye Liu[1], Likang Wu[1], Qi Liu[1], Enhong Chen[1,*], Wei Tong[1], Zi Yan[2]
[1]Anhui Province Key Laboratory of Big Data Analysis and Application, School of Computer Science and Technology, University of Science and Technology of China;
[2]The National Education Examinations Authority of the People's Republic of China
{iiishawn,tongsw,liuyer,wulk,tongustc}@mail.ustc.edu.cn;{qiliuql,chenen}@ustc.edu.cn;yanz@mail.neea.edu.cn

ABSTRACT

Representation learning of examination papers is the cornerstone of the Examination Paper Analysis (EPA) in education area including Paper Difficulty Prediction (PDR) and Finding Similar Papers (FSP). Previous works mainly focus on the representation learning of each test item, but few works notice the hierarchical document structure in examination papers. To this end, in this paper, we propose a novel Examination Organization Encoder (EOE) to learn a robust representation of the examination paper with the hierarchical document structure. Specifically, we first propose a syntax parser to recover the hierarchical document structure and convert an examination paper to an Examination Organization Tree (EOT), where the test items are the leaf nodes and the internal nodes are summarization of their child nodes. Then, we applied a two-layer GRU-based module to obtain the representation of each leaf node. After that, we design a subtree encoder module to aggregate the representation of each leaf node, which is used to calculate an embedding for each layer in the EOT. Finally, we feed all the layer embedding into an output module, the process is over and we get the examination paper representation that can be used for downstream tasks. Extensive experiments on real-world data demonstrate the effectiveness and interpretability of our method.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**;
• **Information systems** → *Document structure.*

KEYWORDS

Document Embedding; Examination Paper Embedding; Structured Document Analysis

ACM Reference Format:
Yixiao Ma, Shiwei Tong, Ye Liu, Likang Wu, Qi Liu, Enhong Chen, Wei Tong and Zi Yan. 2021. Enhanced Representation Learning for Examination Papers with Hierarchical Document Structure. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463068

*Corresponding author.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463068

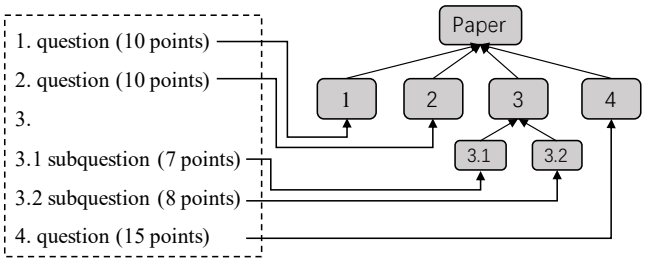

Figure 1: An example of EOT. The left part is an examination paper where the importance point is labeled along the side of each test item. The right part is the hierarchical document structure.

1 INTRODUCTION

Recent years have witnessed the booming of applications in education area, where Examination Paper Analysis (EPA) attracts more and more attention. EPA can help people well depict the features of the examination paper such as Paper Difficulty Prediction (PDR). Additionally, people can conduct Finding Similar Papers to perform duplication detection. The fundamental issue of these applications in EPA is the representation learning of the examination papers.

Some previous works focus on question representation, like[12]. However, as a pivot role in examination paper analysis, designing a specific representation learning method for examination paper is imminent but unexplored. Contemporary researches on representation learning for document mainly focus on document embedding and semantics structural analysis. The works on document embedding concentrate on utilizing deep learning methods to aggregate the global representation from each unit (e.g., TextGCN [11], TextING [13]) and some recent works manage to introduce document layout to embedding such as LayoutLM [9]. Meanwhile, some works conduct semantics analysis to enhance the document representation. For example, Ji et al. [3] apply a Rhetorical Structure Theory (RST) [6] parser to measure importance of different parts in a document. However, existing methods hardly notice the hierarchical document structure in examination papers, which not only reveal the relations between each test item but also indicate the locations of test items.

Examination Organization Tree (EOT), a tree-like representation of the examination paper, whose leaves are test items in the document, and internal nodes are summarization of their child nodes. Some previous works also use tree-like structure, such as [8]. In fact, EOT shows the hierarchical document structure of the examination paper and furthermore implicitly indicates the importance of different test items in the global representation of the examination

paper. As shown in Figure 1, this is a paper with EOT depth of 3, item *question 3.1* with 8 points is considered to be more important than item *question 3.2* with 7 points and item *question 4* with 15 points is considered as more important than item *question 1* with 10 points. We can easily find that the test items in the tail are considered to have higher importance. Thus, we have the conclusion that exploiting the hierarchical document structure can help us to determine the importance of each part of examination papers, and furthermore improve the representation of the paper.

However, there are three challenges in exploiting the EOT in the representation learning of examination papers. First, the hierarchical document structure is implicit which results in EOT can not be directly observed. Second, as there are two different dimension information, i.e., semantics information from the content of test items and structural information from the hierarchical document structure of the examination paper. How to combine these two-dimension information is challenging. Third, although EOT preserves the structure importance information, it is hard to quantize the importance of each part.

To this end, we propose a novel Examination Organization Encoder (EOE) to learn a robust representation of the examination paper with the hierarchical document structure. Specifically, because the hierarchical document structure is implicit, we first propose a syntax parser to recover the structure and convert an examination paper to an Examination Organization Tree (EOT), where the test items are the leaf nodes and the internal nodes are summarization of their child nodes. Then, we applied a two-layer GRU-based module to obtain the representation of each leaf node. After that, to combine the two-dimension information, we design a subtree encoder module to aggregate the representation of each leaf node, which is used to calculate an embedding for each layer in the EOT. Finally, we feed all the layer embedding into an output module to get the examination paper representation, where the importance of each part will be automatically learned. Because directly evaluate the quality of the learned representation of the examination papers is nontrivial, we adopt two classic EPA tasks (i.e., PDR and FSP) to demonstrate the effectiveness and interpretability of our method.

2 EXAMINATION ORGANIZATION ENCODER

The processing flow of EOE framework could be divided into four steps. We firstly extract the EOT from an examination paper by applying a syntax parser on it, based on which we can encode every part of the document, i.e., leaf nodes of an EOT. Then, we implement the subtree encoder to obtain representations for all internal nodes in a recursive way. All layers except for the root layer are subsequently encoded by the Layer Encoder module. Finally, we combine all layer embeddings to acquire the Document Embedding. The architecture of EOE is shown in Figure 2, where the four modules denote the corresponding steps we mentioned above.

2.1 Paragraph Encoder Module

The first stage of our method is constructing the input part. By applying a finite state machine based syntax parser we get the EOT structure of an examination paper, to encoding the leaf node in it, we propose the Paragraph Encoder module as shown in the blue panel in Figure 2. We aim to get the paragraph embedding as leaf nodes of the examination organization tree. In detail, all words in an examination paper are embedded into a matrix $\mathbf{x} \in R^{n_x \times d_x}$ by the Bag-of-Words algorithm, where each row of $\mathbf{x}$ denotes an embedding of a word, n_x is the number of words corpus, and d_x is the dimension of these embeddings. For paragraph i, its embedding vector $\mathbf{p}_i$ is obtained as follows:

$$\mathbf{s}_j = MaxPooling(BiGRU([\mathbf{x}_k])), k \in \Phi_j^s, \tag{1}$$

$$\mathbf{p}_i = MaxPooling(BiGRU([\mathbf{s}_j])), j \in \Phi_i^p, \tag{2}$$

where $\Phi_i^p(\Phi_j^s)$ indicates the set of id which contained in the paragraph $\mathbf{p}_i$(sentence $\mathbf{s}_j$). In a word, we compose two sub-networks into a hierarchical structure, each level consists of a two-layer bi-directional GRU and a MaxPooling layer. The lower-level sub-network generates sentence representations and the higher-level sub-network generates the paragraph representation, that is our leaf node embedding.

2.2 Sub-tree Encoder Module

After the first step, the representation $\mathbf{p}_i$ for each leaf node i of our Examination Organization Tree is prepared. We introduce the sub-tree encoder next. As mentioned in the introduction section, all leaf nodes would make up the children of corresponding internal nodes according to the test paper's structure. Note that the root does not belong to internal nodes and there is leaf node that constructs internal node by only itself such as $\mathbf{p}_1$ and $\mathbf{p}_5$ in Figure 2(the aggregating process in the origin panel of Figure 2 just denotes a kind of possible situation). To get representations of internal nodes, we should run a recursive process that starts from layer $d = depth(EOT)$ to the first layer. Representation for an internal node can be obtained by applying the following equation:

$$\mathbf{h}_i = MaxPooling(BiGRU([\mathbf{h}_j])), j \in Children(i), \tag{3}$$

where the updated state $\mathbf{h}_i$ expresses the representation of node i, and for every leaf node j, its state $\mathbf{h}_j = \mathbf{p}_j$. With the state updated to the root node, all the required presentations are obtained in this process. We can easily find that an internal node's representation is only related to its children, so this encoder module models the importance of a node via a sub-tree level.

2.3 Layer Encoder Module

In general, the problems at an equal structural level may reveal similar importance. For instance, the scores of proving problems would be almost the same in a test paper. On that note, our model considers that the layer embedding is a crucial and necessary point since the layer denotes the structural level in our EOE model. The node representations that come from the sub-tree encoder are regarded as the inputs of our designed layer encoder module. Here we firstly utilize BiGRU to encoder the order information into the node representations in the same layer as follows:

$$\hat{\mathbf{h}}_j^i = BiGRU(\mathbf{h}_j) \begin{cases} if\ \mathbf{h}_j \in LeafNodes\ and\ depth(\mathbf{h}_j) \le i \\ \qquad\qquad or \\ if\ \mathbf{h}_j \in Internal\ Nodes\ and\ depth(\mathbf{h}_j) = i, \end{cases} \tag{4}$$

where $\hat{\mathbf{h}}_j^i$ denotes the processed state of the j-th node in the i-th layer. Due to the various hard levels or importances of problems in the same tree layer, we design an attention layer to evaluate the contribution of each node, which is inspired by [10]. So the output

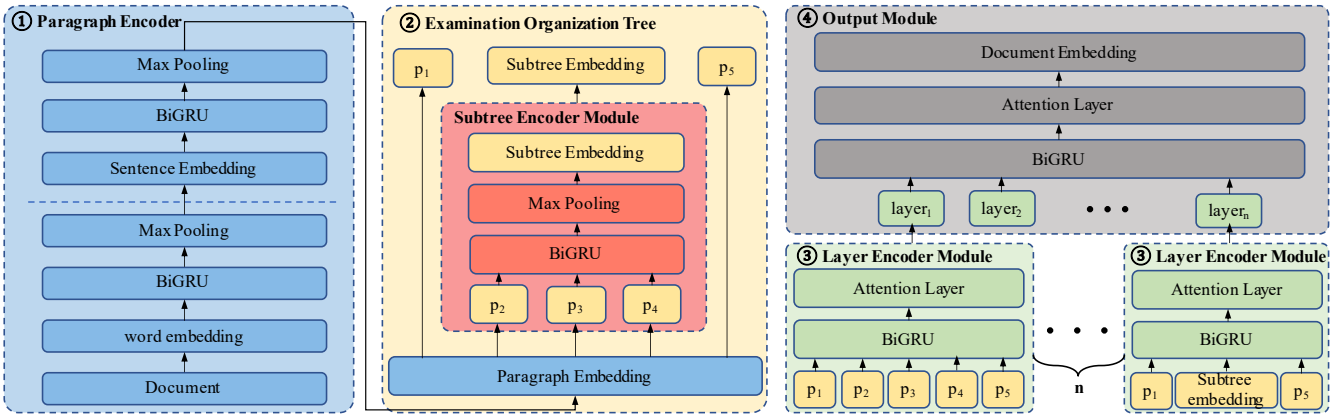

Figure 2: The architecture of EOE

representation $\mathbf{l}_i$ of our layer encoder module can be defined as:

$$\mathbf{u}_{ij} = tanh(W_a \hat{\mathbf{h}}_j^i + b_a), \alpha_{ij} = \frac{\exp\left(\mathbf{u}_{ij}^\top u_w\right)}{\sum_j \exp\left(\mathbf{u}_{ij}^\top u_w\right)}, \mathbf{l}_i = \sum_j \alpha_{ij} \hat{\mathbf{h}}_j^i. \quad (5)$$

In the above equation, α_i indicates the weight distribution of each node in layer i, $u_w$is a context vector that can be seen as a high-level representation of a fixed query "what is the informative word" over the words. u_w is initialed randomly and can be learned automatically during the backward procedure.

2.4 Output Module

In our output module, we aim to aggregate all layer representations to acquire the final document embedding. Similar to the last step, the order information is also necessary to embed in the layer representations by BiGRU model, that is $\hat{\mathbf{l}}_i = BiGRU(l_i), 2 \le i \le depth(EOT)$ for every layer except for the root. The focused degrees of different layers are computed by the attention algorithm, since different types of problems mean different weights in an exam. In detail, we get the final document embedding $\mathbf{D}$ as follow:

$$\mathbf{u_i} = tanh(W_d \hat{\mathbf{l}}_i + b_d), \alpha_i = \frac{\exp\left(\mathbf{u}_i^\top u_d\right)}{\sum_i \exp\left(\mathbf{u}_i^\top u_d\right)}, \mathbf{D} = \sum_i \alpha_i \hat{\mathbf{l}}_i. \quad (6)$$

That means our representation contains more reasonable and crucial structural information, we would evaluate the efficiency of our representation by using it to support downstream tasks. Our EOE model is flexible for different downstream tasks with their own loss functions (e.g., classification loss or ranking loss). For instance, we could choose a state-of-the-art Finding Similar Papers (FSP) model, and replace its paper representation part with EOE. However, in this paper, the comparison between complicated task models is not our focus. We adopt the basic and widely-used models in evaluation tasks, which will better compare the representation ability of different representation models.

3 EXPERIMENTS

In this section, we conduct extensive experiments with EOE on two typical tasks on examination paper data to demonstrate the effectiveness of our proposed method.

Table 1: The statistics of the dataset

Num. examination paper	EOT Depth	Avg. EOT Depth	Avg. Questions per paper
10000	3~4	3.12	18.84

3.1 Experimental Setup

3.1.1 Dataset. The dataset is collected from a widely-used online learning system, which contains mathematical examination papers of high school. We firstly apply a finite state machine based syntax parser to extract the EOT, than we select papers with EOT depths of 3 and 4 because only about 2.8% of papers have an EOT depth greater than 5. After the data pre-processing process, we retained 10000 examination papers. Each paper is labeled with ten categories of difficulty and there are 10,000 pairs of papers with similarity label. Some important statistics are listed in Table 1. We randomly partition the data into training/validating/testing sets with the ratio as 60%/20%/20%.

3.1.2 Evaluation Tasks. We pick two typical downstream tasks related with examination paper representation, namely: Finding Similar Papers (FSP) [7] and Paper Difficulty Prediction (PDP) [2].

The main objective for FSP is to find similar examination papers in large-scale online education systems. In the training process, we choose cosine similarity function to measure the similarity of two arbitrary examination papers, and then adopt the Hinge [1] as loss function. Since this task can be seen as a ranking problem, we adopt NDCG@N (Normalized Discounted Cumulative Gain), the most widely-used metric, for model evaluation.

The second task, namely Paper Difficulty Prediction (PDP), aims to estimate the difficulty of an examination paper. In the actual implementation process, according to the difficulty of examination papers, we manually divide all exam papers into ten categories and formulate this problem as a multi-class classification task. In detail, we connect the representation output of EOE with a fully-connected layer and adopt the cross entropy as loss function. After training, we evaluate the performance using four widely-used metrics including Accuracy (ACC), Precision, Recall, and F-1 score.

Table 2: Performance of comparison methods on different tasks.

Method	Paper Difficulty Prediction				Finding Similar Papers		
	ACC	Precision	Recall	F-1 score	NDCG@5	NDCG@10	NDCG@15
Doc2Vec	0.5515	0.3832	0.3266	0.3526	0.2558	0.2053	0.1790
HiAttention	0.7850	0.4903	0.4263	0.4561	0.3352	0.3010	0.2782
TextGCN	0.7920	0.7525	0.3891	0.5130	–	–	–
RST-based	0.7892	0.6887	0.4320	0.5310	0.3438	0.3122	0.2923
EOE-LEM	0.7360	0.6258	0.3367	0.4367	0.3108	0.2882	0.2638
EOE-ST	0.7993	0.7928	0.4222	0.5509	0.3562	0.3179	0.2892
EOE	**0.8034**	**0.7939**	**0.4228**	**0.5518**	**0.3722**	**0.3330**	**0.3165**

3.1.3 Compared methods. We compare EOE with several document representation methods. All these methods are able to generate examination paper representation, and then be applied to the two evaluation tasks mentioned above. Specifically, these methods are:

- **Doc2Vec**[5] refers to the traditional document representation method, which do not consider the structural information in the document.
- **HiAttention**[10] adopts a 2-level, hierarchical attention mechanism to measure importance of different parts in the document and further get the representation.
- **TextGCN**[11] treats document as a node in the graph, and turns the text classification problem to a node classification one. Note that since we use the code provided by [11], we only test TextGCN on the PDP task.
- **RST-based recursive neural network**[3] implements an rst parser on a document to obtain the discourse structure, which they used to improve document representation.

Then, to further validate the performance of each component in our model, we also design some simplified variants, including:

- **EOE-LEM** removes the Layer Encoder Module (LEM) from EOE, which ignores the role of leaf nodes in the whole document representation learning.
- **EOE-ST** randomly removes sub-trees in EOT structure, which to some extent erases the hierarchical structure.

3.1.4 Implementation Details. In EOE model, the embedding modules all output vectors of size 300. The size of the hidden layer is set to 256. We use Adam optimizer [4] with an initial learning rate of 0.001 and the batch size of our model is set to 32. For baseline models, we use default parameter settings as in their original papers. All experiments are conducted on a cluster of Linux servers with Tesla V100 GPUs.

3.2 Experimental Results

Overall Performance. The comparison results on both two tasks with four different models including EOE are shown in Table 2. There are several observations. First, EOE consistently achieve the best performance on almost all metrics, which demonstrates EOE can effectively capture the EOT structure and apply it to examination paper representation. Second, we can see EOE outperforms its two variants, i.e., EOE-LEM and EOE-ST, which proves the effect of leaf nodes and the whole EOT structure in examination paper representation respectively.

Visualization Analysis. As is mentioned in Section 1, in an examination paper, the test items in the tail, usually the finale question in an examination paper, are considered to have higher importance. In EOE architecture, we utilize attention mechanism in Layer Encoder Module to calculate the weights of leaf nodes or sub-trees, which reflects the influence extent on the whole examination paper representation. Figure 3 shows the visualization of the attention scores in Layer Encoder Module, and it is clear that more attention has been paid to the tail of this module. This finding is consistent with the research conclusion in education field, which demonstrate the effectiveness and rationality of our proposed method.

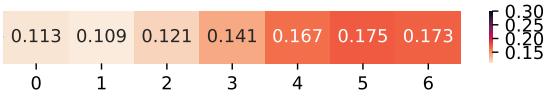

Figure 3: Visualizations of the attention scores in Layer Encoder Module

Discussion. There are some future directions. For one thing, although EOE can effectively capture the structural and semantic relations in text, it omits the heterogeneous data in examination paper data, e.g., images, which also plays an import role in mathematical examination papers. To tackle this issue, we may need a unified modeling architecture to learn heterogeneous representation. For another, we will consider relating EOE model with external knowledge graphs in education field, which may help enhance the representation ability of our method. We hope this work could lead to more future studies.

4 CONCLUSION

In this paper, we propose a novel Examination Organization Encoder (EOE) to learn a robust representation of the examination paper with the hierarchical document structure. Specifically, we first propose a syntax parser to convert an examination paper to an Examination Organization Tree (EOT), where the test items are the leaf nodes and the internal nodes are summarization of their child nodes. Then, through the attention layer, we catch the contribution distribution of various internal nodes and layers to get the examination paper representation with sufficient structural information. With extensive experiments on two typical downstream tasks in education, i.e., Finding Similar Papers and Paper Difficulty Prediction, we proved the strong representation ability of our method.

ACKNOWLEDGMENTS

This research was partially supported by grants from the National Natural Science Foundation of China (Grants No. U20A20229 and 61922073).

REFERENCES

[1] Claudio Gentile and Manfred K. K Warmuth. 1999. Linear Hinge Loss and Average Margin. In *Advances in Neural Information Processing Systems*, M. Kearns, S. Solla, and D. Cohn (Eds.), Vol. 11. MIT Press. https://proceedings.neurips.cc/paper/1998/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf

[2] Zhenya Huang, Qi Liu, Enhong Chen, Hongke Zhao, Mingyong Gao, Si Wei, Yu Su, and Guoping Hu. 2017. Question Difficulty Prediction for READING Problems in Standard Tests. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[3] Yangfeng Ji and Noah A. Smith. 2017. Neural Discourse Structure for Text Categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, 996–1005. https://doi.org/10.18653/v1/P17-1092

[4] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[5] Quoc V. Le and Tomás Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014 (JMLR Workshop and Conference Proceedings, Vol. 32)*. JMLR.org, 1188–1196. http://proceedings.mlr.press/v32/le14.html

[6] WILLIAM MANN and Sandra Thompson. 1988. Rethorical Structure Theory: Toward a functional theory of text organization. *Text* 8 (01 1988), 243–281. https://doi.org/10.1515/text.1.1988.8.3.243

[7] Wei Tong, Shiwei Tong, Wei Huang, Liyang He, Jianhui Ma, Qi Liu, and Enhong Chen. 2020. Exploiting Knowledge Hierarchy for Finding Similar Exercises in Online Education Systems. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1298–1303.

[8] Likang Wu, Zhi Li, Hongke Zhao, Zhen Pan, Qi Liu, and Enhong Chen. 2020. Estimating Early Fundraising Performance of Innovations via Graph-Based Market Environment Model. 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*. AAAI Press, 6396–6403. https://aaai.org/ojs/index.php/AAAI/article/view/6110

[9] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 1192–1200. https://dl.acm.org/doi/10.1145/3394486.3403172

[10] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, Kevin Knight, Ani Nenkova, and Owen Rambow (Eds.). The Association for Computational Linguistics, 1480–1489. https://doi.org/10.18653/v1/n16-1174

[11] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. 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*. AAAI Press, 7370–7377. https://doi.org/10.1609/aaai.v33i01.33017370

[12] Yu Yin, Qi Liu, Zhenya Huang, Enhong Chen, Wei Tong, Shijin Wang, and Yu Su. 2019. QuesNet: A Unified Representation for Heterogeneous Test Questions. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 1328–1336. https://doi.org/10.1145/3292500.3330900

[13] Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020. 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*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 334–339. https://doi.org/10.18653/v1/2020.acl-main.31