Knowledge Graph Integrated Graph Neural Networks for Chinese Medical Text Classification

1st Ge Lan
College of Software
Nankai University
Tianjin, China
grettelan@mail.nankai.edu.cn

2nd Ye Li
College of Artificial Intelligence
Nankai University
Tianjin, China
2120200369@mail.nankai.edu.cn

3rd Mengting Hu*
College of Software
Nankai University
Tianjin, China
mthu@nankai.edu.cn

4th Yufei Sun
College of Software
Nankai University
Tianjin, China
yufei_sun@sina.com

5th Yuzhi Zhang
College of Software
Nankai University
Tianjin, China
zyz@nankai.edu.cn

2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-0126-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/BIBM52615.2021.9669286

***Abstract*—Text classification is a well-developed task in natural language processing. In the medical area, this task is still difficult since the discriminative ability requires domain-specific knowledge, such as high specialization, terminology, and structured relationships. In this work, we leverage a large Chinese medical knowledge graph (KG) to derive the above knowledge and propose graph neural networks (GNN) to make the best use of such knowledge. Particularly, for a medical text, we build two graphs: text-graph, which is based on the occurrences of contextualized words; text-specific knowledge graph, retrieved from KG in light of the common terms between the text and KG. The above two graphs are bridged by the common terms and then merged into a joint one. We propose GNN to learn this graph. As such, our model builds interactions between adjacent nodes, and meanwhile, the medical knowledge can be propagated from KG to text. To enhance the node representations and improve knowledge interaction, we introduce general prior knowledge to the text-graph and domain-specific prior knowledge to the text-specific knowledge graph. We conduct extensive experiments on three medical datasets. The experimental results show that our model outperforms strong baseline methods significantly.**

***Index Terms*—graph neural networks, knowledge graph, medical text classification**

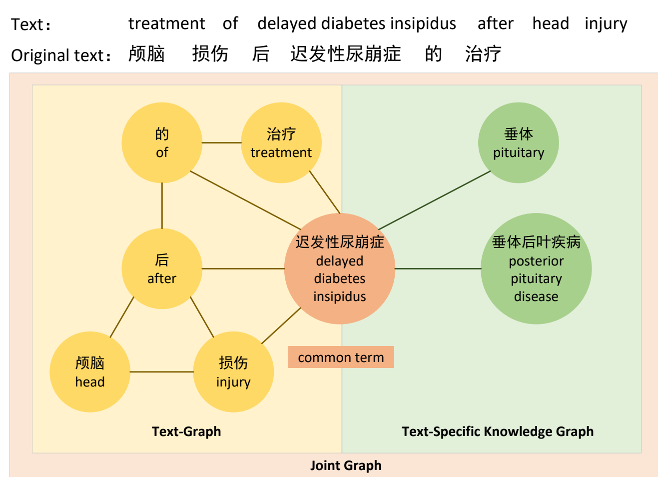

Fig. 1. An example of a joint graph, merged by a text-graph and a text-specific knowledge graph.

I. Introduction

Text classification is a fundamental task for many medical applications, such as disease analysis [1], biomedical topic categorization [2], and finding relationships among biomedical entities [3], [4]. Especially, with the result of the growth of medical research and online medical consultation, the amount of medical texts has expanded. For medical paper retrieval [5]–[7], medical question answering [8]–[10], and other applications, medical text classification has become a critical task. Traditional methods for text classification, including Naive-Bayes [11], Linear Discriminant Analysis (LDA) [12], and Support Vector Machines (SVM) [13], rely heavily on hand-crafted features. Deep learning models, like Convolutional Neural Networks (CNN) [14]–[16] and Recurrent Neural Networks (RNN) [17], [18], have lead promising development for this task. As CNN prioritizes locality and RNN prioritizes sequentiality, they can capture syntax and semantic information. Nevertheless, they may ignore non-consecutive structure in a text or corpus. More recently, GNN-based methods [19]–[21] are proposed to address such issues, which capture the structure information of a text. Defferrard et al. [22] firstly employed GNN in the text classification task. Zhang et al. [23] focused on words local structure within each document (TextING).

However, in medical text classification task, the lack of domain-specific expertise limits the effectiveness of GNN-based methods. For example, in the text "*treatment of delayed diabetes insipidus after head injury*", the model may have never seen the entity `delayed diabetes insipidus`

Mengting Hu* is the corresponding author.

and this makes the medical texts more challenging. To make up for the lack of medical expertise [24], we introduce a medical KG, which contains large scale specialized knowledge, terms and structured relationships to enhance the semantic and structured medical information.

In this paper, we propose KG integrated GNN for Chinese medical text classification (TextKGNN). To leverage the abundant prior knowledge, the common terms between the text and medical KG are first extracted, like `delayed diabetes insipidus` in Figure 1. Based on the common terms, two graphs can be obtained for a text. Firstly, the text is converted into a *text-graph* by a sliding window [23]. The nodes within a sliding window are connected to each other. Secondly, a *text-specific knowledge graph* is derived from KG, including the common terms and their adjacent nodes. Bridging through the common terms, the above two graphs can be merged into a *joint graph*. We propose GNN, learning upon the joint graph, to transfer the domain knowledge from KG to the text. With the interaction between nodes, the representations of the text are enhanced.

In addition, depending on the characteristics of the two graphs, different prior knowledge is introduced to further reinforce the node embedding. On the one hand, we employ word embedding pre-trained on the general corpus to introduce the general prior knowledge for the text. On the other hand, we pre-train graph embeddings on the whole KG to preserve the global structure and semantic information, which aims to enhance the domain prior knowledge for the text-specific knowledge graph.

In summary, our contributions are threefold:

- We propose KG integrated GNN for Chinese medical text classification. To the best of our knowledge, this is the first work that incorporates the KG to GNN for text classifying task.
- Prior knowledge with the special design is introduced to enhance node representations, including both the general and domain-specific structure and semantic knowledge.
- Extensive experiments on three Chinese medical datasets demonstrate that our approach outperforms strong baseline methods significantly.

II. Methodology

The architecture of TextKGNN mainly includes four modules, illustrated in Figure 2. Firstly, a graph embedding method, i.e. node2vec, is employed to pre-train the node representations of the medical KG. Secondly, with the common terms between text and KG, we retrieve a text-specific knowledge graph from KG and initialize its nodes with the pre-trained representations. Thirdly, each document is constructed to a text-graph and integrated with the relevant text-specific knowledge graph into a joint graph. Fourthly, upon the joint graph, nodes update their representations interactively via GNN and are used for inference.

A. Graph Embedding on KG

In recent years, knowledge graphs in many domains have been built with high-quality domain-specific knowledge [25]. OMAHA[1] is a large scale medical KG, including 980,000 ontologies, 1.24 million entities, and 2.92 million relationships. We utilize the prior knowledge in OMAHA, including global structure and semantic information, to enhance the expertise of the document.

Specifically, graph embedding aims to represent each node of the graph in vector space. Its basic idea is that the adjacent nodes are close to each other in the vector space. Upon the OMAHA medical KG, we choose node2vec, a random walk-based graph embedding method, to learn the representations for nodes. Starting from each node in the KG, node2vec utilizes biased-random walks that provide a trade-off between breadth-first and depth-first graph searches to obtain fixed-length walks. In the fixed-length walk, node2vec employs skip-gram [26], a self-supervised model, to learn the semantic knowledge for each node. For a T-length walk $v_1, v_2, ..., v_T$, each node v_i can obtain its context $c_i = \{v_{i+j} | -n \leq j \leq n, j \neq 0\}$, where n is the window size.

The input of the model is the one-hot vector of v_i, and the output of the model represents the probability of c_i nodes co-existing with v_i, the object function is defined as below.

$$\begin{aligned} \Gamma &= \sum_{\mathrm{i}=1}^{T} \sum_{-n\leq j\leq n, j\neq 0} \log P(v_{i+j}|v_i) \\ &= \sum_{\mathrm{i}=1}^{T} \sum_{-n\leq j\leq n, j\neq 0} \log soft\max(\mathbf{x}_{v_i}^T \mathbf{x'}_{v_{i+j}}) \end{aligned} \tag{1}$$

where $\mathbf{x'}_{v_{i+j}}$ is the output weights of v_{i+j}, and $\mathbf{x}_{v_i}$ represents the node embedding of node v_i. Through learning on a large number of fixed-length KG walks, node2vec can get domain-rich and informative representation of the KG node.

B. Knowledge Graph Retrieval

This module aims to retrieve KG nodes related to the document through the text common term and generate a text-specific knowledge graph with these nodes. We treat the co-occurring words between text and KG as the common terms. To build the text-specific knowledge graph based on the common term, we need to obtain all their adjacent terms from KG. However, in the OMAHA KG, the terms are not directly connected to each another. Instead, a term is linked to its ontology, and related ontologies are linked to each other.

Assume the term `headache` is a common term, terms set $\varepsilon_1 = \{$`headache`, `head pain`$\}$ share a ontology, and $\varepsilon_2 = \{$`head`, `head area`$\}$ belongs to another ontology. Since the above ontologies are connected to each other, we can obtain ε_1 and ε_2, and connect `headache` with all other terms from the two sets. To improve the diversity, we then randomly sample from ε_1 and ε_2 to obtain a united node set V_1. If there are several common terms in the text, we will repeat

[1] https://hita.omaha.org.cn/term/dataSource

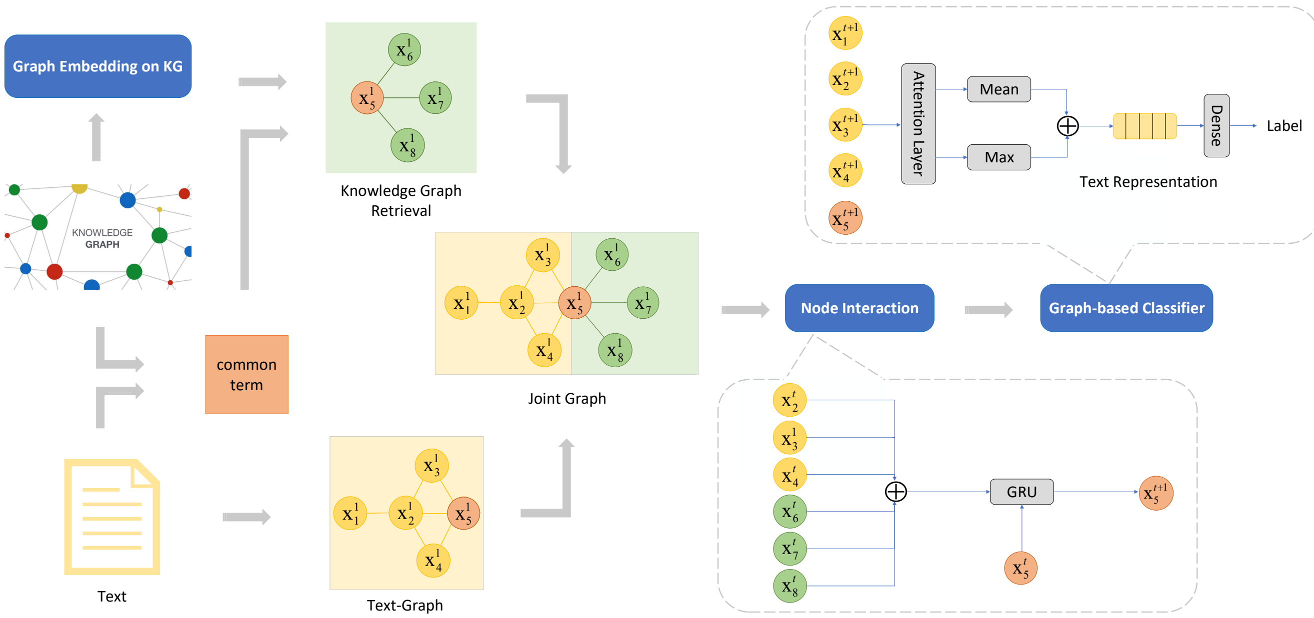

Fig. 2. The architecture of TextKGNN. Best view in color. In the node interaction module, we take $\mathbf{x}_5^t$ for example to show that each node is affected by all its neighbours.

the above steps for each common term to obtain V_1. The text-specific knowledge graph is constructed by V_1 and the ontologies relationships, concretely defined as below.

$$\begin{aligned} V_1 &= \{v_i | i \in [1, l]\} \\ E_1 &= \{e_{ij} | i \in [1, l]; j \in \beta_i\} \end{aligned} \tag{2}$$

where l is the number of nodes, E_1 is the edge set and β_i is the linked node set of the i_{th} node. Then, we initialize the nodes in V_1 with the pre-trained graph embeddings.

$$N_1 = \{\mathbf{x}_i | i \in [1, l]\} \tag{3}$$

where $\mathbf{x}_i$ is the representation of i_{th} node in V_1. We formulate the text-specific knowledge graph as $G_1 = \{N_1, E_1\}$.

C. Joint Graph

The joint graph is merged with the text-graph and the text-specific knowledge graph. Before introducing the joint graph, we first illustrate how to construct the text-graph. In advance, the terms from KG are added into a Chinese word segmentation tool, i.e. pyltp [27]. We exploit pyltp[2] to obtain the node set for text-graph.

$$V_2 = \{v_i | i \in [1, m]\} \tag{4}$$

where m is the number of nodes in text-graph. The edges are defined by the occurrences between nodes within a fixed-size sliding window [23]. The text-graph $G_2 = \{N_2, E_2\}$ is defined as below.

$$\begin{aligned} N_2 &= \{\mathbf{x}_i | i \in [1, m]\} \\ E_2 &= \{e_{ij} | i \in [1, m]; j \in [i - w, i + w]\} \end{aligned} \tag{5}$$

[2]https://github.com/HIT-SCIR/pyltp

where N_2 is the node representation set, $\mathbf{x}_i$ is the representation of i_{th} node in V_2, E_2 denotes the edge set and w is the size of the sliding window. It is worth noting that the common terms in the text-graph are initialized by the pre-trained graph embeddings rather than general word embeddings. This aims to accelerate the transferring of domain-specific knowledge.

Since the text-graph and the text-specific knowledge graph have shared common terms, these terms are utilized to bridge two graphs into a joint one. We fuse G_1 and G_2 into a joint graph G.

$$\begin{aligned} N &= N_1 \cup N_2 \\ E &= E_1 \cup E_2 \end{aligned} \tag{6}$$

where N and E denote the representation set and the edge set of the joint graph, respectively.

D. Node Interaction and Classifier

Concretely, our model exploits the Gated Graph Neural Networks (GGNN) [28] to update the representations of nodes interactively. This indicates that a node is affected by all its neighbours. By updating multiple steps, we can derive better structured representations for text, with integrating both the general and domain-specific knowledge. At the t step, the node interaction is formulated as below.

$$\begin{aligned} \mathbf{a}^t &= \mathbf{A}\mathbf{x}^{t-1}\mathbf{W}_a \\ \mathbf{z}^t &= \sigma(\mathbf{W}_z\mathbf{a}^t + \mathbf{U}_z\mathbf{x}^{t-1} + \mathbf{b}_z) \\ \mathbf{r}^t &= \sigma(\mathbf{W}_r\mathbf{a}^t + \mathbf{U}_r\mathbf{x}^{t-1} + \mathbf{b}_r) \\ \widetilde{\mathbf{x}}^t &= \tanh(\mathbf{W}_h\mathbf{a}^t + \mathbf{U}_h(\mathbf{r}^t \odot \mathbf{x}^{t-1}) + \mathbf{b}_h) \\ \mathbf{x}^t &= \widetilde{\mathbf{x}}^t \odot \mathbf{z}^t + \mathbf{x}^{t-1} \odot (1 - \mathbf{z}^t) \end{aligned} \tag{7}$$

TABLE I
THE STATISTICS OF THE DATASETS. AVG. NODES MEANS THE AVERAGE NUMBER OF NODES IN THE DATASET. G IS THE JOINT GRAPH, G_1 IS THE TEXT-SPECIFIC KNOWLEDGE GRAPH AND G_2 IS THE TEXT-GRAPH.

Dataset	Docs	Training + Validation	Test	Classes	Avg.Nodes of G	Avg.Nodes of G_1	Avg.Nodes of G_2
DA	14,482	13,071	1,411	42	71.62	4.49	68.47
EM	20,099	18,070	2,029	36	73.14	4.58	70.18
RC	12,616	11,338	1,278	24	72.47	5.00	68.83

where matrix $\mathbf{A} \in \mathbb{R}^{|V| \times |V|}$ denotes the adjacency matrix of the joint graph. $\mathbf{x}^{t-1} \in \mathbb{R}^{|V| \times d}$ denotes the node representations at $(t-1)$ step and d is the dimension of representation. σ is the logistic sigmoid function, $\odot$ is the Hadamard product, $\mathbf{z}$ and $\mathbf{r}$ are the update gate and reset gate, all $\mathbf{W}$, $\mathbf{U}$ and $\mathbf{b}$ are parameter matrices and biases.

Since every nodes have different importance, a node-level soft attention mechanism is proposed to highlight the major nodes for classification, as f_{att} formulated. We then combine these attention weights with the non-linear projected representations f_{emb}.

$$\begin{aligned} f_{att}(\mathbf{x}^t) &= \sigma(\mathbf{W}_{att}\mathbf{x}^t + \mathbf{b}_{att}) \\ f_{emb}(\mathbf{x}^t) &= \tanh(\mathbf{W}_{emb}\mathbf{x}^t + \mathbf{b}_{emb}) \\ \mathbf{x} &= f_{att}(\mathbf{x}^t) \odot f_{emb}(\mathbf{x}^t) \end{aligned} \tag{8}$$

We utilize all joint graph nodes to obtain the graph-level representation of the document $\mathbf{x}_G$, as the formulation below.

$$\mathbf{x}_G = \mathrm{Mean}(\mathbf{x}) + \mathrm{Max}(\mathbf{x}) \tag{9}$$

Our model averages weighted word features to gather each word information. The max-pooling function is applied to focus on the nodes which have the most significant feature value for the $\mathbf{x}_G$. In the end, we feed the document feature $\mathbf{x}_G$ into a classification layer consisting of a fully connected layer and a softmax layer to get the one-hot label. Our model is trained by minimizing the cross-entropy loss using stochastic gradient descent [29].

III. EXPERIMENTS

A. Datasets

To evaluate the proposed method, we build three Chinese datasets for medical text classification. Concretely, the datasets are collected from Chinese medical journal articles published on HowNet[3]. These articles have been labeled by professional annotators based on the Chinese Library Taxonomy[4]. We exploit the title, keywords, and abstract for classifying. Three datasets have a sufficient amount of articles. The statistics of datasets are displayed in Table I. The details of the datasets are as follows.

- **Digestive system and abdominal diseases (DA)** contains 14,482 articles which are classified into 42 classes, including 9 stomach diseases, 11 intestinal diseases, and 2 esophageal diseases, etc.
- **Endocrine gland and metabolic diseases (EM)** contains 20,099 articles which are classified into 36 classes, including 5 thyroid diseases, 4 pituitary system diseases, and 4 adrenal gland diseases, etc.
- **Respiratory and chest diseases (RC)** contains 12,616 articles which are classified into 24 classes, including 3 trachea diseases, 4 bronchial diseases, and 10 lung diseases, etc.

Compared with public datasets, these medical datasets contain more professional terms and are finer-grained. Different classes may have similar contents, making it difficult for classification. For example, *superficial gastritis* , *atrophic gastritis* and *hypertrophic gastritis* are three different classes on DA dataset while entity `gastritis` is a high-frequency word in all these classes.

B. Compared Methods

Baseline Methods: We compare our proposed model TextKGNN with the following methods:

- **TextRNN** [17]: It utilizes RNN to calculate the sequential representations of the text.
- **TextCNN** [14]: It is a classic CNN-based classifier based on pre-trained word embeddings.
- **TextRCNN** [30]: It jointly learns the sequential information by RNN and the local contextual information by CNN for text classification.
- **fastText** [31]: It is a simple and efficient text classification method via averaging words or n-gram embeddings as the text embedding.
- **TextGCN** [32]: It is a graph-based method for text classification, which constructs one global graph for the whole corpus.
- **TextING** [23]: It captures the local structure of the text by constructing a graph with the sliding window. Then it uses GGNN to compute the text representation.

C. Experimental Settings

We randomly split each dataset into the training, validation, and test set by the ratio of 8:1:1. The hyper-parameter tuning is conducted on the validation set. We choose Adam [29] as the optimizer. The learning rate is set to 0.005. The dropout rate is 0.5, and the sliding window size is 3. We set the batch size as 2,048, the training epoch as 200, and the joint graph

[3]https://www.cnki.net/
[4]http://www.ztflh.com/?c=17417

TABLE II
EVALUATION RESULTS IN TERMS OF ACCURACY.

Model	DA	EM	RC
TextRNN	0.7125 $\pm$ 0.0038	0.7018 $\pm$ 0.0018	0.7107 $\pm$ 0.0042
TextRCNN	0.7325 $\pm$ 0.0087	0.7314 $\pm$ 0.0103	0.7291 $\pm$ 0.0048
fastText	0.7033 $\pm$ 0.0021	0.7036 $\pm$ 0.0032	0.7041 $\pm$ 0.0034
TextGCN	0.6779 $\pm$ 0.0018	0.6686 $\pm$ 0.0006	0.6740 $\pm$ 0.0069
TextCNN	0.7386 $\pm$ 0.0044	0.7437 $\pm$ 0.0007	0.7557 $\pm$ 0.0039
TextING	0.7378 $\pm$ 0.0057	0.7570 $\pm$ 0.0032	0.7528 $\pm$ 0.0059
TextKGNN	**0.7484†‡ $\pm$ 0.0029**	**0.7657†‡ $\pm$0.0022**	**0.7587 $\pm$ 0.0026**

TABLE III
THE RESULTS OF ABLATION STUDIES ON DA AND EM DATASETS.

Setting	DA	EM
Original	**0.7484 $\pm$ 0.0029**	**0.7657 $\pm$ 0.0022**
(1)Remove KG nodes	0.6010 $\pm$ 0.0092	0.6885 $\pm$ 0.0047
(2)Remove Graph Emb.	0.6896 $\pm$ 0.0026	0.7335 $\pm$ 0.0012
(3)Random Text Word Emb.	0.7108 $\pm$ 0.0076	0.7477 $\pm$ 0.0103

word interaction step t as 2. The dimension of embeddings is set to 300. Text words are initialized by word embeddings pre-trained on Sogou News corpus[5] via skip-gram. Text nodes not in pre-trained vocabulary are randomly initialized. For KG nodes, we employ node2vec on the KG to pre-train node embeddings. Node2vec walk length is set to 50 and its window size is 5. It is worth noting that all reported results are the average score and the standard deviation of 5 runs.

D. Experiment Results

The evaluation results of our model and all baselines are reported in Table II. Firstly, among all baseline methods, TextCNN achieves the best performance on DA and RC dataset, and TextING achieves the best performance on EM dataset. The proposed method TextKGNN obtains the best result. Especially compared with the above two strong baselines, TextKGNN outperforms them significantly, where the marker † refers to p-values$<$0.05 when comparing with TextCNN and the marker ‡ refers to p-values$<$0.05 when comparing with TextING. The results suggest that our model successfully integrates the KG to enrich the text knowledge and structure information. The motivation of TextKGNN is closer to the learning process of human beings, which has the intrinsic ability to transfer abundant knowledge to the unfamiliar scenario.

[5]http://www.sogou.com/labs/resource/list news.php

E. Ablation Study

To analyze the effects of individual components in our model, the ablation study is conducted. The results are shown in Table III.

In (1), the KG nodes are only exploited to enrich the word segmentation dictionary but not merged into a joint graph. This means that only the text-graph is fed to the node interaction module. Removing KG nodes causes that the accuracy significantly drops on both DA and EM datasets, showing the effectiveness of KG information. This indicates that text-specific knowledge graph nodes provide relevant information for the text and really enhance the performance of text classification.

In (2), KG nodes are initialized by general word embeddings rather than our pre-trained graph embeddings. Compared with the full model TextKGNN, its performances significantly drop. On the one hand, KG nodes with general word embeddings will have similar representations to the text words, making the KG nodes lose important KG information. On the other hand, KG nodes are usually uncommon, which appear in the general corpus, like the Sogou News corpus, with low-frequency or even nonexistent. Therefore, graph embedding is an essential module in the knowledge fusion process of our model. It can learn global structured and semantic knowledge from KG. Therefore, our model brings enhancement for the document when training our model.

In (3), we initialize the text words with random embeddings rather than the pre-trained word representations. We can also see slight performance drops on both datasets. This shows that the pre-trained word embeddings contain the general knowledge, which is beneficial for the proposed model.

IV. CONCLUSION

In this paper, we proposed KG integrated GNN for Chinese medical text classification (TextKGNN). Experimental results demonstrated that our model outperforms strong baseline methods significantly. It enhanced the text with medical expertise from a large medical KG. For each node in the joint graph, GNN enabled the node interaction and knowledge transfer.

Furthermore, well-designed prior knowledge enriched the local and global, the general and domain-specific, the structure and semantic knowledge. These all showed the effectiveness of the proposed model. In the future, we will explore the effects of incorporating other information from the knowledge graph, such as different types of relationships. In addition, to further improve the efficiency of KG for text classification, we will try to reduce the influence of KG noise on text.

ACKNOWLEDGMENT

This work is supported by National Key Research and Development Program (2018YFB0204304).

REFERENCES

[1] J. Sastre, A. H. Vahid, C. McDonagh, and P. Walsh, "A text mining approach to discovering covid-19 relevant factors," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 486–490.

[2] X. Chen, C. Ruan, Y. Zhang, and H. Chen, "Heterogeneous information network based clustering for categorizations of traditional chinese medicine formula," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2018, pp. 839–846.

[3] E. Wang, F. Wang, Z. Yang, L. Wang, Y. Zhang, H. Lin, and J. Wang, "Extracting biomedical relations via a multi-head attention based graph convolutional network," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 793–798.

[4] Y. Xu, Z. Yang, Y. Song, Z. Li, L. Wang, Y. Zhang, H. Lin, and J. Wang, "Star-bilstm-lan for document-level mutation-disease relation extraction from biomedical literature," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 357–362.

[5] A. Mourão, F. Martins, and J. Magalhaes, "Multimodal medical information retrieval with unsupervised rank fusion," *Computerized Medical Imaging and Graphics*, vol. 39, pp. 35–45, 2015.

[6] H. Wang, Q. Zhang, and J. Yuan, "Semantically enhanced medical information retrieval system: a tensor factorization based approach," *Ieee Access*, vol. 5, pp. 7584–7593, 2017.

[7] L. Goeuriot, G. J. Jones, L. Kelly, H. Müller, and J. Zobel, "Medical information retrieval: introduction to the special issue," *Information Retrieval Journal*, vol. 19, no. 1-2, pp. 1–5, 2016.

[8] T. R. Goodwin and S. M. Harabagiu, "Medical question answering for clinical decision support," in *Proceedings of the 25th ACM international on conference on information and knowledge management*, 2016, pp. 297–306.

[9] A. B. Abacha and P. Zweigenbaum, "Means: A medical question-answering system combining nlp techniques and semantic web technologies," *Information processing & management*, vol. 51, no. 5, pp. 570–594, 2015.

[10] A. B. Abacha and D. Demner-Fushman, "Recognizing question entailment for medical question answering," in *AMIA Annual Symposium Proceedings*, vol. 2016. American Medical Informatics Association, 2016, p. 310.

[11] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos, "An evaluation of naive bayesian anti-spam filtering," in *Proceedings of the workshop on Machine Learning in the New Information Age*, 2000.

[12] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," *ACM*, p. 375, 2009.

[13] G. Forman, "Bns feature scaling: an improved representation over tf-idf for svm text classification," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 263–270.

[14] Y. 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*. ACL, 2014, pp. 1746–1751.

[15] Z. Zhang, X. Xu, W. Gong, Y. Chen, and H. Gao, "Efficient federated convolutional neural network with information fusion for rolling bearing fault diagnosis," *Control Engineering Practice*, vol. 116, p. 104913, 2021.

[16] Z. Zhang, C. Guan, H. Chen, X. Yang, W. Gong, and A. Yang, "Adaptive privacy preserving federated learning for fault diagnosis in internet of ships," *IEEE Internet of Things Journal*, 2021.

[17] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[18] B. Zuo, J. Cheng, and Z. Zhang, "Degradation prediction model for proton exchange membrane fuel cells based on long short-term memory neural network and savitzky-golay filter," *International Journal of Hydrogen Energy*, vol. 46, no. 29, pp. 15 928–15 937, 2021.

[19] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, and K. Simaan, "Graph convolutional encoders for syntax-aware neural machine translation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[20] J. Chen, T. Ma, and C. Xiao, "Fastgcn: Fast learning with graph convolutional networks via importance sampling," in *ICLR*, 2018.

[21] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[22] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, pp. 3844–3852, 2016.

[23] Y. Zhang, X. Yu, Z. Cui, S. Wu, Z. Wen, and L. 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*, 2020, pp. 334–339.

[24] L. LiuQiao, L. DuanHong *et al.*, "Knowledge graph construction techniques," *Journal of computer research and development*, vol. 53, no. 3, p. 582, 2016.

[25] L. Li, P. Wang, J. Yan, Y. Wang, and Y. Liu, "Real-world data medical knowledge graph: construction and applications," *Artificial Intelligence in Medicine*, vol. 103, no. 19, p. 101817, 2020.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Computer Science*, 2013.

[27] H. Yan, B. Deng, X. Li, and X. Qiu, "Tener: Adapting transformer encoder for named entity recognition," 2019.

[28] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *Computer Science*, 2015.

[29] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.

[30] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[31] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 01 2017, pp. 427–431.

[32] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 7370–7377.