FAQ Retrieval using Question-Aware Graph Convolutional Network and Contextualized Language Model

Wan-Ting Tseng[1], Chin-Ying Wu[1], Yung-Chang Hsu[2] and Berlin Chen[1]

[1] National Taiwan Normal University, Taiwan
60847014s@gapps.ntnu.edu.tw, elain1224@gmail.com, berlin@csie.ntnu.edu.tw
[2] EZAI, Taiwan
mic@ez-ai.com.tw

***Abstract*—Frequently asked question (FAQ) retrieval, which seeks to provide the most relevant question, or question-answer (QA) pair, in response to a user's query, has found its applications in widespread use cases. More recently, methods based on bidirectional encoder representations from Transformers (BERT) and its variants, which typically take the word embeddings of a question in training time (or query in test time) as the input to predict relevant answers, have shown good promise for FAQ retrieval. However, these BERT-based methods do not pay enough attention to the global information specifically about an FAQ task. To cater for this, we in this paper put forward a question-aware graph convolutional network (QGCN) to induce vector embeddings of vocabulary words, thereby encapsulating the global question-question, question-word and word-word relations which can be used to augment the embeddings derived from BERT for better FAQ retrieval. Meanwhile, we also investigate leverage domain-specific knowledge graphs to enrich the question and query embeddings (denoted by K-BERT). Finally, we conduct extensive experiments to evaluate the utility of the proposed approaches on two publicly-available FAQ datasets (viz. TaipeiQA and StackFAQ), where the associated results confirm the promising efficacy of the proposed approach in comparison to some top-of-the-line methods.**

Keywords—Frequently Asked Question, Graph Convolutional Networks., knowledge graph, language model

I. INTRODUCTION

The ever-increasing volumes of text and multimedia information repositories on the Internet has accelerated the demand to design and develop effective frequently asked question (FAQ) retrieval [1], [2], [3]. FAQ has found its applications in a vast range of use cases, like customer care services, online forums and among others. It is common practice to facilitate FAQ retrieval by leveraging a collection of question-answer (denoted by *Q-A*) pairs that seems to recur frequently to search for the most relevant answer (or *Q-A* pair) with regard to a user's query (denoted by *q* for short). A critical intermediate step for FAQ retrieval is to construct suitable representations for both questions and queries. Many of the early attempts made use of hand-crafted features such as sparse lexical features pertaining to bag-of-word and *n*-grams statistics, named entities, and the like. Of late, there have been a bunch of efforts devoted to employing deep learning models, including but is not limited to convolutional neural networks (CNN) [4], recurrent neural network (RNN) [5], transformer [6] and their extensions, to derive context-aware question or query embeddings in a data-driven manner. Among them, bidirectional encoder representations from Transformers (BERT) has recently aroused much attention due to its excellent performance in capturing semantic interactions between two or more text units [7]. Nevertheless, the above models can only capture local word-level semantic and syntactic information within a question or query, largely ignoring the global information of an FAQ retrieval task, such as non-consecutive semantics relatedness between training questions, as well as words in different but similar questions.

Graph convolutional networks (GCN) have proven effective [8], [9], [10], [11] on tasks containing global language structures or dependency relationships, such as co-occurrence relations among words, importance measures between documents and words, and similarity relations among documents or sentences. As an illustration, by first constructing a graph composed of words, sentences or documents as nodes, as well as relations between these nodes as edges, we can perform convolution operations on nodes connected to one another in the graph, so as to obtain vector representations of nodes that naturally depend on those of their neighbors. As such GCN can facilitate capturing the global context of a domain-specific language usage to a certain extent [12], [13]. In view of the above, we in this paper propose to capitalize on a question-aware graph convolutional network to complement BERT (denoted by QGCN-BERT) for enhanced FAQ retrieval. By doing so, wet induce vector embeddings of vocabulary words, encapsulating the global question-question, question-word and word-word relations, which are considered beneficial to FAQ retrieval.

On a separate front, although the BERT-based supervised method determines the relevance between the query and an answer based on context-aware semantic embeddings, which model local dependency among words and get around the term-mismatch problem to some extent, either generic or domain-specific knowledge clues have not been put to good use in the FAQ process. For this reason, we also investigate to inject triplets of entity relations distilled from an open-domain knowledge base into BERT to expand and refine the representations of an input query for more accurate relevance estimation (denoted by K-BERT) [14], [20].

The main contributions of this paper can be summarized as follows:

(1) To our knowledge, we are the first one to construct a heterogeneous graph network for the FAQ retrieval task, so as to model the relations between questions and words, which involves not only word nodes but also question nodes.

(2) We compare two types of methods for model fusion, which combines the capability of BERT with a Question-aware Graph Convolutional Network (viz. QGCN-BERT).

(3) We explore to inject clues drawn from an open-domain (generic) knowledge graph to facilitate BERT to expand and refine the representations of an input query for more accurate relevance estimation (viz. K-BERT). Alternatively, we add the knowledge graph to QGCN-BERT.

II. RELATED WORK

A. *Frequently Asked Question Retrieval*

The task of FAQ retrieval is to rank a collection of question-answer (*Q-A*) pairs, $\{(Q_1, A_1), \ldots, (Q_n, A_n), \ldots, (Q_N, A_N)\}$, with respect to a user's query, and then return the answer of the topmost ranked one as the desired answer [16], [17], [18], [19], [20].

Recently, a common thread to FAQ retrieval has been to rank question-answer pairs by considering either the similarity between the query and a question (viz. the *q-Q* similarity measure), or the relevance between the query and the associated answer of a question (viz. the *q-A* relevance measure). For example, the *q-Q* similarity measure can be computed with unsupervised information retrieval (IR) models, such as the vector space method [22] and the Okapi BM25 method [21], to name just a few. Meanwhile, the *q-A* relevance can be determined with a simple supervised neural model stacked on top of a pre-trained contextual language model, which takes a query as the input and predicts the likelihoods of all answers given the query. Prevailing contextual language models, such as BERT [7], embeddings from language models (ELMo) [23], generative pre-trained transformer (GPT) [24], the generalized autoregressive pretraining method (XLNet) and many others, can serve this purpose to obtain context-aware query embeddings. Among them, BERT has recently aroused much attention due to its excellent performance on capturing semantic interactions between two or more text units. More specifically, BERT is an effective neural contextualized language model, which makes effective use of bi-directional self-attention (also called the Transformer) to capture both short and long span contextual interaction between the tokens in its input sequence [6]. In contrast to the traditional embedding methods, the advantage of BERT is that it can produce different question-aware representations for the same word at different locations by considering bi-directional dependence relations of words across consecutive sentences and it also allows for word order and other local information, which is of crucial importance in understanding the meaning of a sentence.

B. *Graph Convolutional Networks (GCN)*

Recently, there have been several attempts in the literature to extend neural networks to deal with arbitrarily structured graphs. One of the prevailing paradigms is the family of graph convolutional networks (GCN). GCN is instantiated with multilayer neural networks (usually consisting of 2 layers) that employ convolution operators on a target graph and iteratively aggregates the embeddings of the neighbors for every node on the graph to generate its own embedding [25], [27], [28]. A bit of terminology: consider a graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ that is encompasses a set of nodes $\mathrm{V} = \{v_1, v_2, \ldots, v_n\}$ and a set of edges $\mathrm{E} = \{e_{i,j}\}$, where any given pair of nodes v_i and v_j is connected by an edge $e_{i,j}$ if they have a neighborhood relationship (or share some properties). We can represent the graph with either an adjacency matrix or a derived vector space representation. Furthermore, the degree matrix D of the graph G is defined by $\mathrm{D}_{i,i} = \sum_j \mathrm{A}_{i,j}$. For GCN equipped merely with a single-layer structure, the updated feature matrix of all nodes on G is calculated as follows:

$$\mathrm{H}^{(1)} = ReLU(\widetilde{\mathrm{A}}\mathrm{X}\mathrm{W}_0) \tag{1}$$

where $\mathrm{X} \in \mathbb{R}^{n \times m}$ is the input matrix that contains the corresponding m-dimensional feature vectors of all nodes, $\mathrm{W} \in \mathbb{R}^{m \times k}$ is a weight matrix to be estimated, $\widetilde{\mathrm{A}} = \mathrm{D}^{-\frac{1}{2}}\mathrm{A}\mathrm{D}^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix. The normalization operation that converts A to $\widetilde{\mathrm{A}}$ is to avoid numerical instabilities and exploding (or vanishing) gradients when estimating W of the corresponding GCN model for G. Building on this procedure, we can extend to capture higher-order neighborhood information from G by stacking multiple GCN layers:

$$\mathrm{H}^{(i+1)} = ReLU(\widetilde{\mathrm{A}}\mathrm{H}^{(i)}\mathrm{W}_i) \tag{2}$$

where i denotes the layer number and $H^{(0)}=X$.

The authors of [28] first proposed semi-supervised learning of GCN for a node-level classification task. In addition, [12] regarded simultaneously documents and words of a text corpus as nodes to construct the corpus graph (a heterogeneous graph) and used GCN to learn embeddings of words and documents. It can capture global word co-occurrence information, given a limited number of labeled documents is provided.

III. PROPOSED APPROACH

This section first sheds light on the way to employ BERT to measure the *q*-*A* relevance, and then describes how the word-level embeddings of a query generated by GCN that was previously trained on a question-word heterogeneous graph constructed from the training questions can be used to augment the query embedding generated by BERT. After that, we introduce the ways that the triplets of entity-level or word-level semantic and pragmatic relations extracted from an open-domain knowledge base and we inject the knowledge graph to QGCN-BERT to make the model have domain knowledge[29], [30], [31], [32].

A. FAQ Retrieval with BERT

FAQ retrieval manages to search for the most relevant answer (or question-answer pair) from a dataset in response a user's query. Note here that since a given answer may be associated with different questions, which means that the total number of distinct answers may be smaller than or equal to *N* (the size of question-answer pairs compiled a priori). To fulfill FAQ retrieval, we can develop a supervised ranking model built on top of BERT. Specifically, the model encompasses a single layer neural network stacked on top of a pre-trained BERT model to estimate the *q*-*A* relevance measure for ranking the collection of *Q*-*A* pairs. In the test phase, the model will accept an arbitrary query q as the input and its output layer will predict the posterior probability $P(A_n|q), n=1,\dots,N$, of any answer A_n (denoted also by $BERT(q,A_n)$). The answer A_n that has the highest $P(A_n|q)$ value will be regarded as the desired answer that is anticipated to be the most relevant to q. On the other side, in the training phase, since the test queries are not given in advance, we can instead capitalize on the corresponding relations of existing *Q*-*A* pairs for model training. More concretely, a one-layer feedforward neural network (FFNN) is trained (and meanwhile the parameters of BERT are fine-tuned) by maximizing the $P(A_n|Q_n)$ for all the *Q*-*A* pairs in the collection [20].

B. Heterogeneous Graph Embeddings for FAQ Retrieval

Although BERT has been proven powerful in capturing the contextual information within a sentence or document., their ability of capturing the global information about the vocabulary and structure of a language is relatively limited. In this review, we explore the use of heterogeneous graph embeddings to augment the query embeddings generated from BERT. The heterogeneous graph is constructed with training questions and words in these questions as the nodes [33], [34]. The cooccurrence relationship among any pair of word nodes i and j is represented as an undirect edge with a weight that quantify their relatedness, which can be computed using a formula that is expressed by normalized point-wise mutual information (NPMI) [35]:

$$\mathrm{NPMI}(w_i,w_j)=-\frac{1}{log p(w_i,w_j)}log\frac{p(w_i,w_j)}{p(w_i)p(w_j)} \qquad (3)$$

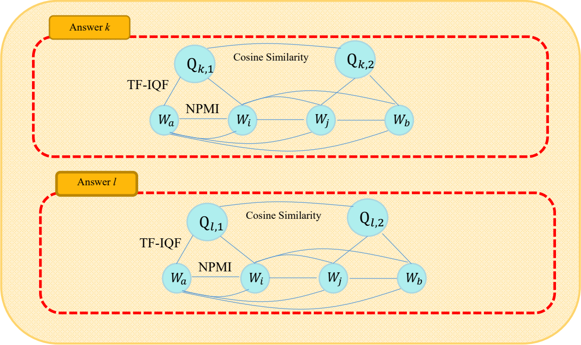

Figure 1: A schematic depiction of the construction of a heterogeneous graph for FAQ retrieval, where an edge is additionally created to connect any two questions that correspond to the same answer.

where *i* and *j* denote arbitrary two distinct words, $p(w_i)=\frac{\#W(i)}{\#W}$, $p(w_i,w_j)=\frac{\#W(i,j)}{\#W}$, $\#W(i)$ and $\#W(i,j)$ are the numbers of sliding windows respectively containing word i and words i and j, and $\#W$ is the total number of sliding windows. In this study we set the sliding window equal to a sentence. Note also that NPMI has its value ranging from -1 to 1: the higher the value the closer the semantic relation between two words, and vice versa. Meanwhile, the weight of the undirected edge between a question node and a word node is determined by the term frequency-inverse question frequency (TF-IQF) [36] score, which is expressed by

$$\text{TF-IQF}(Q_k,w_i)=\frac{n_{k,i}}{\sum_j n_{k,j}}log\frac{|\mathbf{Q}|}{1+|\{k':w_i\in Q_{k'}\}|} \qquad (4)$$

where $n_{k,i}$ is the number of times that word w_i occurs in question Q_k, $\sum_k n_{k,j}$ is the sum of the number of occurrence counts of all words in Q_k, $|\mathbf{Q}|$ is the total number of distinct questions in the collection of training question-answer pairs, and $|\{k':w_i\in Q_{k'}\}|$ is the number of questions where w_i appears [38], [39], [40], [41], [41]. Furthermore, since distinct questions may be associated with the same answer, we can additionally construct an edge between any two distinct questions that correspond to the same answer, while the weight of the edge is quantified based on the cosine similarity score between the two questions. We hereinafter term this extension as query-aware GCN (QGCN). After the construction of the heterogeneous graph, we can apply GCN (or QGCN) on the graph to obtain the corresponding GCN (or QGCN) embeddings of words. To this end, the model parameters of GCN (or QGCN) are trained with the cross-entropy objective function which aims to the discrepancy between the reference answer and the prediction output of a one-layer FFNN module that takes every training question as the input to predict its corresponding answer. Figure 1 shows a schematic depiction of the construction of a heterogeneous graph for FAQ retrieval, where an edge is additionally created to connect any two questions that correspond to the same answer.

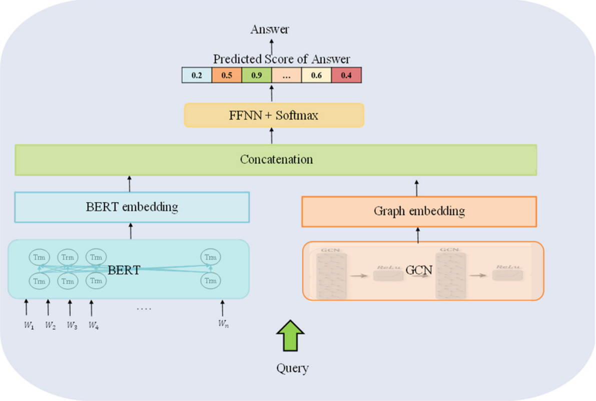

Figure 2: A schematic description of the integration of the GCN embedding of a query into the BERT-based model with the late-fusion strategy.

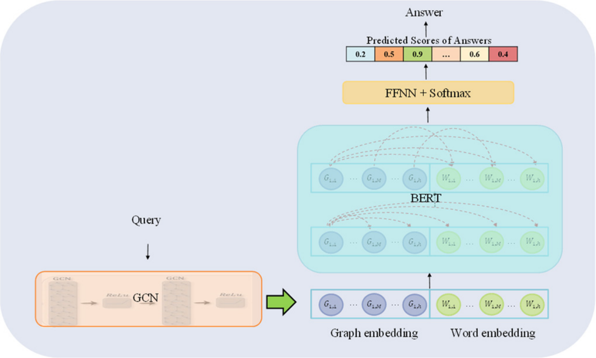
Answer
Predicted Scores of Answers
0.2
0.5
0.9
...
0.6
0.4
FFNN + Softmax
BERT
$G_{1,1}$ ... $G_{1,M}$ ... $G_{1,h}$ $W_{1,1}$ ... $W_{1,M}$ ... $W_{1,h}$
Query
GCN
ReLu
Graph embedding
Word embedding

Figure 3: A schematic description of the integration of the GCN embedding of a query into the BERT-based model with the early-fusion strategy.

In this paper, we adopt two disparate strategies to infuse the GCN embedding of a question (at training time) or a query (at test time) into the BERT-based model for enhanced FAQ retrieval. We will refer to the first as the late-fusion strategy (*cf.* Figure 1) while the second as the early-fusion strategy (*cf.* Figure 2). At training time, for the late-fusion strategy, the FFNN module is modified to takes input the concatenation of the GCN embedding and the BERT embedding of a training question. The retrieval model is trained to harness the synergistic strength of both local and global contextual information of the query for enhanced answer prediction. Note here that the GCN embedding of a training question is obtained by taking the average of the GCN embeddings for all words involved in it. One the other hand, for the early-fusion strategy, the BERT module take input the combination of the original word embeddings of all words involved in a question with the corresponding GCN embeddings of these words in succession.

Table 1: Summary statistics of two benchmark datasets for our experiments on FAQ retrieval.

	# train	#validation	# test	#Classes
TaipeiQA	5,821	1,665	1,035	149
StackExchange	750	250	250	125

This way, the wording embeddings and GCN embedding of all words in a question are tightly integrated through the multilayer self-attention mechanism of the BERT module. The output of the BERT module is anticipated to capture the order of the words in the question but also learn the domain-specific structure information obtained by GCN. In the same vein, the GCN embedding of a test question can be obtained for both the late-fusion and early-fusion strategies at test time. We can also use QGCN to replace GCN for the above attempts and procedures.

C. *Supervised Knowledge Injections for the q-A Relevance Measure*

On a separate front, the BERT-based method (either with or without the augmentation of GCN embedding for a test query) still might not perform well on knowledge-driven tasks like FAQ retrieval or question-matching, due to the incapability of modelling open-domain knowledge about deeper semantic and pragmatic interactions of words (entities) [20]. To ameliorate this problem, a surge of research has emerged recently to incorporate information distilled from an open-domain knowledge base [42], such as WordNet [43], HowNet [44], YAGO [45], or a domain-specific knowledge base, such as MedicalKG, into BERT-based models for different application tasks. Representative methods include, but is not limited to, the THU-ERNE [46] method and the Knowledge-enabled BERT (K-BERT) method. On the practical side, K-BERT seems more attractive than THU-ERNE, because it can easily inject a given open-domain knowledge base, in the form of a set of triplets $(w_i, \text{relation}_r, w_j)$ that describe disparate relations between words or entities, into a pretrained BERT-based model structure through the so-called soft-position and visible-matrix operations [14]. As such, in this paper, we also attempt to exploit K-BERT to incorporate open-domain knowledge clues for use in the *q-A* relevance measure [47], [48], [49], [50], [20].

IV. EXPERIMENTS

A. *Experimental Setup*

We assess the effectiveness of our proposed approach on two publicly-available FAQ retrieval datasets, viz. TaipeiQA and StackExchange. TaipeiQA is a publicly-available Chinese FAQ dataset crawled from the official website of the Taipei City Government, which consists of 8,521 *Q-A* pairs and is further divided into three parts: the training set (68%), the validation set (20%) and the test set (12%). Note here that the questions in the validation and test sets are taken as unseen queries, which are used to tune the model parameters and evaluate the performance of FAQ retrieval, respectively. On the other hand, StackExchange is a publicly-available English corpus. It was crawled from the StackExchange community-driven QA site, consisting of 125 distinct answers, each of which has 10 distinct questions corresponding to it. This leads to 1, 250 *Q-A* pairs in total. We divided the 1,250 *Q-A* pairs of StackExchange into three parts as well: the training set (60%), the validation set (20%), and the test set (20%).

B. Experimental Results

In the first set of experiments, we conduct a series of empirical evaluations on the efficacy of the various enhanced BERT-based models proposed in this paper, in ration to the vanilla BERT-based model and the GCN-based model (the latter directly uses the GCN embedding of a query as the input to the FFNN module for answer prediction; *cf.* Section III.B), for the TaipeiQA dataset. The corresponding results are shown in Table 2, from which we can make at least four observations. First, the BERT-based model achieves better performance than the GCN-based model, which reals that the local word-order and word-interaction information captured by BERT seems to outweigh the task-specific, global language structure captured by GCN for FAQ retrieval. Second, when infusing the GCN embedding into the BERT-based model, the early-fusion strategy (*cf.* Row 3) delivers better results than the late-fusion strategy (*cf.* Row 4), though the performance gap is moderate. Both these two strategies can considerably promote the retrieval effectiveness of the vanilla BERT-based model. Third, if QGCN is used instead of GCN to generate the embedding of a test query to be integrated into the BERT-based model (*cf.* Rows 5 and 6), slight and consist improvements can be further achieved for most evaluation metrics (expect for the precision metric); how to more effectively leverage the relationships between questions in the training set is still worthy of further investigation. Fourth, we turn to investigate the utility of injecting triplets $(w_i, \text{relation}_r, w_j)$, which were distilled from an open-domain knowledge base (viz. HowNet), to describe disparate relations between words for use in the vanilla BERT-based model and our best-performing model, respectively (*cf.* Section III.C). As can be seen from Rows 7 and 8 of Table 2, such injection of generic, open-domain knowledge cues seems to lead mixed results. At the time of writing of this paper, we are extensively exploring novel ways to incorporate domain-specific knowledge cues into variants of the BERT-based model in an unsupervised manner.

In the second set of experiments, we move on to the empirical evaluations on the above-mentioned models for the StackExchange dataset. The corresponding results are shown in Table 3, which reveals at least two noteworthy points. First, the performance gap between the GCN-based model and the vanilla BERT-based model becomes larger. One possible reason is that since the training *Q-A* pairs of StackExchange are much fewer than that of TaipeiQA, which inevitably makes the GCN-based model (which was trained from scratch) incur the problem of data-sparsity. This problem is less pronounced for the vanilla BERT-based model since it was pretrained with huge amounts of general corpora and then fine-tuned on the training set of StackExchange. Second, due similarly to the data-sparsity problem, the variants of our proposed modelling approach, which infuse the GCN embedding of a test query into the BERT-based model in different ways, do not lead to significant performance improvements, as compared to that achieved on TaipeiQA. It would be an important research direction to explore effective data augmentation mechanisms to enrich the training set of StackExchange.

Table 2: Experimental results on the TaipeiQA dataset.

	Precision	Recall	F1	Accuracy
BERT	0.688	0.675	0.681	0.697
GCN	0.644	0.606	0.624	0.606
BERT+GCN (late-fusion)	0.759	0.719	0.738	0.719
BERT+GCN (early-fusion)	0.764	0.725	0.744	0.725
BERT+QGCN (late-fusion)	0.759	0.723	0.740	0.722
BERT+QGCN (early-fusion)	0.761	0.731	0.745	0.731
K-BERT	0.705	0.685	0.694	0.706
K-BERT+QGCN (early-fusion)	0.764	0.730	0.746	0.730

Table 3: Experimental results on the StackExchange dataset.

	Precision	Recall	F1	Accuracy
BERT	0.941	0.937	0.938	0.937
GCN	0.527	0.510	0.518	0.510
BERT+GCN (late-fusion)	0.936	0.934	0.934	0.934
BERT+GCN (early-fusion)	0.965	0.955	0.959	0.955
BERT+QGCN (late-fusion)	0.940	0.941	0.940	0.941
BERT+QGCN (early-fusion)	0.931	0.934	0.932	0.934

V. Conclusion and Future Work

In this paper, we have proposed a novel modeling approach to integrate the corresponding GCN and QGCN embeddings of training questions (or test queries) into the BERT-based model for enhanced FAQ retrieval. Extensive experiments conducted on the TaipeiQA and StackExchange datasets seems to confirm the effectiveness and viability of our approach. As to future work, we plan to explore more sophisticated ways to obtain GCN and QGCN embeddings of training questions and test queries, as well as to investigate different syntactic and semantic cues for use in the construction of the heterogeneous graph for better representing questions and their constituent words.

VI. REFERENCES

[1] Mladen Karan and Jan Šnajder. 2018. Paraphrase-focused learning to rank for domain-specific frequently asked questions retrieval. *Expert Systems with Applications*, 91: 418-433.

[2] Mahmoud Abo Khamis, Hung Q. Ngo, et al. 2016. FAQ: questions asked frequently. In P*roceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems* pages. 13-28.

[3] Wataru Sakata, Tomohide Shibata, et al. 2019. FAQ retrieval using query-question similarity and BERT-based query-answer relevance. In *Proceedings of the International ACM SIGIR Conference on Research and Devel-opment in Information Retrieval*, pages 1113–1116.

[4] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. 2017. Understanding of a convolutional neural network. In *Proceedings of the International Conference on Engineering and Technology* (ICET). IEEE.

[5] Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenomena 404: 132306.

[6] Ashish Vaswani, Noam Shazeer, et al. 2017. Attention is all you need. In *Proceedings of Conference on Neural Information Processing Systems*, pages 5998–6008.

[7] Jacob Devlin, Ming-Wei Chang, et al. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* pages 4171–4186.

[8] Franco Scarselli et al. The graph neural network model. 2008. *IEEE Transactions on Neural Networks* 20.1: 61-80.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. 2016. *Advances in neural information processing systems* 29: 3844-3852.

[10] Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*.

[11] Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering* 30.9: 1616-1637.

[12] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33.

[13] Jasmijn Bastings et al. 2017. Graph convolutional encoders for syntax-aware neural machine translation. *arXiv preprint arXiv:1704.04675*.

[14] Weijie Liu, Peng Zhou, et al. 2020. K-BERT: enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence AAAI,* pages 2901–2908.

[15] Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In Proceedings of the Conference on Uncertain-ty in Artificial Intelligence, pages 289–296.

[16] Yosi Mass et al. 2020. Unsupervised FAQ retrieval with question generation and BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[17] Sparsh Gupta and Vitor R. Carvalho. 2019. FAQ retrieval using attentive matching. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 929–932, New York, NY, USA. ACM.

[18] Mladen Karan and Jan Snajder. 2016. FAQIR: A frequently asked questions retrieval test collection. In *Proceedings of* Text, *Speech, and Dialogue*, volume 9924. Springer.

[19] Barun Patra. 2017. A survey of community question answering. CoRR, abs/1705.04009.

[20] Wan-Ting Tseng et al., "Effective FAQ retrieval and question matching tasks with unsupervised knowledge injection," The 24th International Conference on Text, Speech and Dialogue (TSD 2021), Olomouc, Czech Republic, September 6-9, 2021.

[21] Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond, *Foundations and Trends in Information Retrieval,* 3(4): 333–389.

[22] Gerard Salton, Andrew Wong, and Chungshu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM, 18(11),* pages 613–620.

[23] Matthew Peters, Mark Neumann, et al. 2018. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Lan-guage Technologies,* pages 2227–2237.

[24] Zhilin Yang, Zihang Dai, et al. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Proceedings of Conference on Neural Information Processing Systems,* pages 5753–5763.

[25] Zonghan Wu et al. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

[26] Fenxiao Chen et al. 2020. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing* 9.

[27] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2020. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

[28] Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

[29] Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering* 30.9: 1616-1637.

[30] Quan Wang et al. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29.12: 2724-2743.

[31] Zhen Wang et al. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 28. No. 1.

[32] Yankai Lin et al. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. No. 1.

[33] Danqing Wang et al. 2020. Heterogeneous Graph Neural Networks for Extractive Document Summarization. *arXiv preprint arXiv:2004.12393*.

[34] Linmei Hu et al. 2019. Heterogeneous graph attention networks for semi-supervised short 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)*.

[35] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of GSCL*: 31-40.

[36] Jalilifard Amir et al. 2020. Semantic Sensitive TF-IDF to Determine Word Relevance in Documents. *arXiv preprint arXiv:2001.09896*.

[37] Stephen Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation*.

[38] Kuo Zhang et al. 2006. Keyword extraction using support vector machine. In *international conference on web-age information management*. Springer, Berlin, Heidelberg.

[39] Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*. Vol. 242.

[40] Juanzi Li, Qi'na Fan and Kuo Zhang. 2007. Keyword extraction based on tf/idf for Chinese news document. *Wuhan University Journal of Natural Sciences* 12.5: 917-921.

[41] Ho Chung Wu et al. 2008. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)* 26.3: 1-37.

[42] Wanyun Cuix, Yanghua Xiao, et al. 2017. KBQA: learning question answering over QA corpora and knowledge bases. *Proceedings of the VLDB Endowment*, 10(5): 656–676.

[43] George A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, *38*(11): 39–41.

[44] Zhendong Dong, Qiang Dong, and Changling Hao. 2010. HowNet and Its Computation of Meaning. In *Proceed-ings of the International Conference on Computational Linguistics,* pages 53–56.

[45] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: a core of semantic knowledge. In *Proceedings of the international conference on World Wide Web,* pages 697–706.

[46] Zhengyan Zhang, Xu Han, et al. 2019. ERNIE: enhanced language representation with informative entities, In *Proceedings of the Annual Meeting of the Association for Computational Linguistics,* pages 1441–1451.

[47] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. arXiv preprint arXiv:1909.03193.

[48] Quan Wang, Mao Zhendong, et al. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, *29*(12), 2724-2743.

[49] Guoliang Ji, Shizhu He, et al. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pages 687-696.

[50] Guoliang Ji, Kang Liu, et al. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *Thirtieth AAAI conference on artificial intelligence*.

[51] Svetlana Kiritchenko, Matwin Stan, et al. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Proceedings of Conference of the Canadian Society for Computational Studies of Intelligence,* pages 395-406.

[52] Zhen Wang, Jianwen Zhang, et al. 2014. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1591-1601.

[53] Matthew E Peters, Neumann Mark, et al. 2018. Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

[54] KARAN, Mladen; ŠNAJDER, Jan. Paraphrase-focused learning to rank for domain-specific frequently asked questions retrieval. *Expert Systems with Applications*, 2018, 91: 418-433.