8-2024

Nonfactoid question answering as query-focused summarization with graph-enhanced multihop inference

Yang DENG
Singapore Management University, ydeng@smu.edu.sg

Wenxuan ZHANG
Alibaba Group

Weiwen XU
Chinese University of Hong Kong

Ying SHEN
Sun Yat-sen University

Wai LAM
Chinese University of Hong Kong

Non-factoid Question Answering as Query-focused Summarization with Graph-enhanced Multi-hop Inference

Yang Deng, Wenxuan Zhang, Weiwen Xu, Ying Shen, and Wai Lam

***Abstract*—Non-factoid question answering (QA) is one of the most extensive yet challenging applications and research areas in natural language processing (NLP). Existing methods fall short of handling the long-distance and complex semantic relations among the question and the document sentences. In this work, we propose a novel query-focused summarization method, namely Graph-enhanced Multi-hop Query-focused Summarizer (GMQS), to tackle the non-factoid QA problem. Specifically, we leverage graph-enhanced reasoning techniques to elaborate the multi-hop inference process in non-factoid QA. Three types of graphs with different semantic relations, namely semantic relevance, topical coherence, and coreference linking, are constructed for explicitly capturing the question-document and sentence-sentence interrelationships. Relational Graph Attention Network (RGAT) is then developed to aggregate the multi-relational information accordingly. In addition, the proposed method can be adapted to both extractive and abstractive applications as well as be mutually enhanced by joint learning. Experimental results show that the proposed method consistently outperforms both existing extractive and abstractive methods on two non-factoid QA datasets, WikiHow and PubMedQA, and possesses the capability of performing explainable multi-hop reasoning.**

***Index Terms*—Non-factoid Question Answering, Query-focused Summarization, Graph Neural Network, Multi-hop Reasoning**

I. Introduction

NON-FACTOID Question Answering (QA) has received a significant amount of attention recently due to its board applications on a variety of real-world Community-based Question Answering (CQA) sites, such as Quora, StackOverFlow, and Amazon Q&A. Different from factoid QA [1], which can be simply answered by a short text span or a single sentence without detailed information, e.g., "Who is the author of Harry Potter?", the answers for non-factoid questions are supposed to be more informative, involving some detailed analysis, like opinions and explanations, to explain or justify the final answers, such as questions in community QA [2], [3] or explainable QA [4], [5]. Non-factoid QA contains a wider range of open-ended questions, including "How" or "Why" questions, yes-no questions. For example, "How to tube feed a puppy?" or "Are human coronaviruses uncommon in patients with gastrointestinal illness?" cannot be answered without the context from the document, as the example in Figure 1&6.

In practice, non-factoid QA requires the capability of merging multiple sparse and diverse information from different sentences across the whole supporting document or evidences together to form a concise and complete answer. Document summarization methods have been adopted as an effective way to summarize salient information, which can also be adopted to provide a concise answer for the given question in the context of non-factoid question answering [7], [8]. Essentially, the key to tackling the non-factoid QA problem is to measure the relevance degree between the question and candidate answer sentences [9], [10]. This leads to a variety of researches that elaborate the semantic interactions between the question and candidate answer sentences, from Siamese Neural Models [11], [12] to Compare-Aggregate Models [13], [14]. However, traditional document summarization methods, when being applied on non-factoid QA [8], [15], fall short of capturing the important semantic interactions between the question and the document sentence.

To achieve this, we investigate the non-factoid QA problem as a query-focused summarization problem, as they share a similar goal to produce a concise but informative summary, driven by a specific query. In the past studies, query-focused summarization was mainly explored by traditional information retrieval methods [2], [7], [16], which heavily rely on hand-crafted features or tedious multi-stage pipelines. Inspired by the promising performance of deep learning models on other NLP tasks, several efforts have been made on developing deep learning based models [3], [17]–[19] to summarize the source document with the guidance of specific queries. However, most of them focus on capturing the semantically relevant information with the query to produce the summary, while failing to provide informative and logical answers due to the overlook of two crucial characteristics in non-factoid QA:

- The long-distance interrelationships among the document sentences make it difficult to fetch all the necessary information for constructing the final answer.
- The complex semantic relations attach great importance to the reasoning procedure and the explainability of the answer.

As the example shown in Figure 1, given the specific question, there are several **highlighted** sentences required

The work described in this article is substantially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14200719), the National Natural Science Foundation of China (No.61602013), and the Shenzhen General Research Project (No. JCYJ20190808182805919). *(Corresponding author: Ying Shen)*

Yang Deng, Weiwen Xu, and Wai Lam are with the Department of System Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong 999077. (E-mail: {ydeng, wwxu, wlam}@se.cuhk.edu.hk)

Wenxuan Zhang is with with DAMO Academy, Alibaba Group. (E-mail: saike.zwx@ alibaba-inc.com)

Ying Shen is with the School of Intelligent Systems Engineering, Sun Yat-sen University, Gunagzhou 510275, China. (E-mail: sheny76@ mail.sysu.edu.cn)

Question: ***Are human coronaviruses uncommon in patients with gastrointestinal illness?***

Document: <S>Coronaviruses infect numerous animal species causing a variety of illnesses including respiratory, neurologic and enteric disease. <S>***Human coronaviruses (HCoV) are mainly associated with respiratory tract disease but have been implicated in enteric disease.*** <S>***To investigate the frequency of coronaviruses in stool samples from children and adults with gastrointestinal illness by RT-PCR.*** <S>Clinical samples submitted for infectious diarrhea testing were collected from December 2007 through March 2008. <S>**RNA extraction and RT-PCR was performed for stools negative for Clostridium difficile using primer sets against HCoV-229E, HCoV-OC43, HCoV-NL63, and HCoV-HKU1.** <S>Clinical data from samples positive for coronaviruses were reviewed and recorded. <S>Samples from 479 patients were collected including 151 pediatric (< or = 18 years), and 328 adults (>18 years). <S>Of these samples, 4 patients (1.3%, 2 adult; 2 pediatric) screened positive for the presence of a coronavirus. <S>**All detected coronaviruses were identified as HCoV-HKU1.** <S>**No stools screened positive for either HCoV-229E, HCoV-NL63 or HCoV-OC43.** <S>**All HCoV-HKU1 positive samples occurred between mid-January to mid-February.** <S>**Clinical manifestations from HCoV-HKU1 positive patients included diarrhea, emesis and respiratory complaints.** <S>Three (75%) patients were admitted to the hospital with a median length of stay of 6 days. <S>

Answer: ***Coronaviruses as a group are not commonly identified in stool samples of patients presenting with gastrointestinal illness.*** **HCoV-HKU1 can be identified in stool samples from children and adults with gastrointestinal disease, with most individuals having respiratory findings as well. No stool samples screened positive for HCoV-NL63, HCoV-229E, or HCoV-OC43.**

Fig. 1. An example from PubMedQA [6]. The **highlighted** sentences illustrate the inference process when humans answer the given question. ***Italic*** represents direct matching sentences from the question. **Underlined** and **wavy-underlined** represent sentences inferred by 2nd-hop and 3rd-hop reasoning, respectively, to justify the answer.

to be concentrated for conducting summarization so as to generate the answer. Besides, one-time inference sometimes is insufficient for collecting all the required information for producing a complete answer. It leads to the necessity of measuring the importance of each sentence, instead of regarding the source text as an undifferentiated whole. Inspired by recent advances in factoid QA studies [20], [21], one intuitive approach to address the long-distance interrelationship issue is to employ multi-hop reasoning, which enables to collect all the important justifications or evidences that contribute to the final answer. Recently, [22] develops a multi-hop inference module for non-factoid QA, based on the semantic relevance degree among the document sentences. Despite its effectiveness, the multi-hop reasoning patterns are *implicitly* obtained from a *single* relation, i.e., semantic relevance. There are two other semantic relations that have been identified to be useful in studying the interrelationship among the document sentences in summarization: topical coherence [23], [24] and coreference linking [25], [26]. On one hand, despite the content transition in the multi-hop inference process, the latent topic concerning the given question is supposed to be coherent. On the other hand, resolving coreference across the whole document can bridge the long-distance relationship between different sentences that are discussing the same object.

Fortunately, graph structures have the natural advantages of exploiting both structural and semantic information to reason over multi-hop relational paths. Existing graph-enhanced multi-hop reasoning techniques are basically proposed for factoid QA [26]–[29], which aims to construct entity graphs for linking the mentioned entities among sentences. Then, graph neural networks [30], such as GCN [31], [32], GAT [33]–[35], are employed to model the multi-hop information transition. However, in non-factoid QA, the semantic relationships among sentences are more complicated. Such *multiple* relations between textual units are expected to be fully utilized in a unified graph for detecting salient information and performing *explicit* reasoning.

In this work, we tackle the non-factoid QA problem by proposing a novel query-focused summarization method, namely Graph-enhanced Multi-hop Query-focused Summarizer (GMQS). In specific, we investigate graph-based reasoning techniques to conduct the multi-hop inference for collecting the key information from the document towards the given question. Three types of graphs with different semantic relations, namely **Semantic Relevance**, **Topical Coherence**, and **Coreference Linking**, are constructed for explicitly capturing the question-document and sentence-sentence interrelationships. Relational Graph Attention Network (RGAT) is then developed to aggregate the multi-relational information accordingly. In addition, the multi-hop relational information can then be utilized under either extractive or abstractive application to produce a summary as the answer to the given non-factoid question. We empirically show that the proposed method outperforms existing baselines on non-factoid QA with a promising capability of multi-hop reasoning.

A preliminary study was published as a conference paper [22]. We substantially enhance the method with three main improvements: 1) We propose a new graph-enhanced multi-hop reasoning model for non-factoid QA. 2) We develop an adaptive relational graph attention network with a multi-relational graph structure for modeling the complex sentence relations. 3) We unify the extractive and abstractive query-focused summarization into one Transformer-based architecture. In addition, we conduct extensive experiments to validate the proposed method from various aspects, such as automatic and human evaluation for both extractive and abstractive scenarios, the contribution of different components, and detailed analyses of the multi-hop reasoning process. Overall, the proposed GMQS method substantially improves MSG [22] with better performance, training efficiency, and explainability.

The main contributions are summarized as follows:

- We propose a novel query-focused summarization method to tackle the non-factoid question answering problem, which leverages graph-enhanced reasoning techniques to elaborate the multi-hop inference for summarizing the key information to form the answer to the given non-factoid questions.
- We identify three types of semantic relations, namely semantic relevance, topical coherence, and coreference linking, for explicitly modeling the question-document and sentence-sentence relationships. Relational Graph Attention Network is developed to aggregate the multi-relational information.
- The proposed method unifies the extractive and abstractive query-focused summarization into one architecture, which can jointly improve the summarization performance of non-

factoid QA and be adaptively used for different applications.
- Experimental results on two non-factoid QA datasets, namely WikiHowQA and PubMedQA, show that the proposed method substantially and consistently outperforms several strong baselines.

II. Related Works

A. Non-factoid Question Answering

Different from factoid QA that can be tackled by extracting answer spans [1], [36], generating short sentences [37] or returning a Boolean answer [38], non-factoid QA aims at producing relatively informative and complete answers. Most non-factoid QA studies focus on information retrieval (IR) based methods, such as answer sentence selection [9] or answer ranking [10], [39], by measuring the semantic relevance degree between the question and candidate answers or answer sentences [12]–[14], including Siamese architecture [11], [12] and Compare-Aggregate framework [13]. In the Siamese architecture [11], [12], the same encoder is used to learn the vector representations for the input sentences (both questions and answers), individually. In order to enhance the interaction between the representational learning of the question and answer, various attention mechanisms [40]–[42] are proposed to attend the correlated and important information for a better relevance measurement. Furthermore, the Compare-Aggregate architecture [13], [14], [43] captures more interactions between two sentences, by aggregating comparison signals from low-level elements into high-level representations.

Inspired by the successful applications of text generation on other NLP tasks, some recent studies [3], [4], [44] adopt generation-based methods to generate natural sentences as the answer in non-factoid QA. In specific, several efforts have been made on tackling long-answer generative question answering over supporting documents, which targets on questions that require detailed explanations [4]. This kind of QA problem contains a large proportion of non-factoid questions, such as "how" or "why" type questions [3], [45]. Besides, some studies aim at generating a conclusion for the concerned question [5], [6]. [4] proposes a multi-task Seq2Seq model with the concatenation of the question and support documents to generate long-form answers. [46] and [5] incorporate some background knowledge into Seq2Seq model for generating natural answers to why questions and conclusion-centric questions.

However, existing studies on non-factoid QA typically focus on capturing the question-related content from the document. In this paper, we tackle the non-factoid QA as a query-focused summarization problem, which aims to further merge sparse and diverse information from different sentences across the whole document to form a concise but complete answer.

B. Query-focused Summarization

Early works on query-focused summarization mainly investigate the approach to extracting query-related sentences to construct the summary [19], [47], [48], which are later improved by exploiting sentence compression on the extracted sentences [23], [49]. Recently, some data-driven neural abstractive models are proposed to generate the natural form of summaries with respect to the given query [17], [18], [50]. However, current studies on query-focused abstractive summarization are restricted by the lack of large-scale datasets [18], [51]. To overcome this challenge, researchers explore the utilities of weak supervision [52] and domain adaptation [53] techniques by leveraging external resources from some related tasks, or unsupervised learning [16], [54].

In the light of both the capability and limitation of query-focused summarization studies, some researchers spark a new pave of query-focused summarization in non-factoid QA [2], [7], [55], which requires the ability of reasoning or inference in summarization, not merely relevance measurement, and also preserves remarkable testbeds of large-scale datasets. Similar to traditional summarization, according to the type of summary, query-focused summarization studies in non-factoid QA can also be categorized into extractive [2], [7], [15], [55] and abstractive summarization [3], [22], [56]. In this paper, we investigate the capability of multi-hop reasoning for adapting query-focused summarization methods into non-factoid QA.

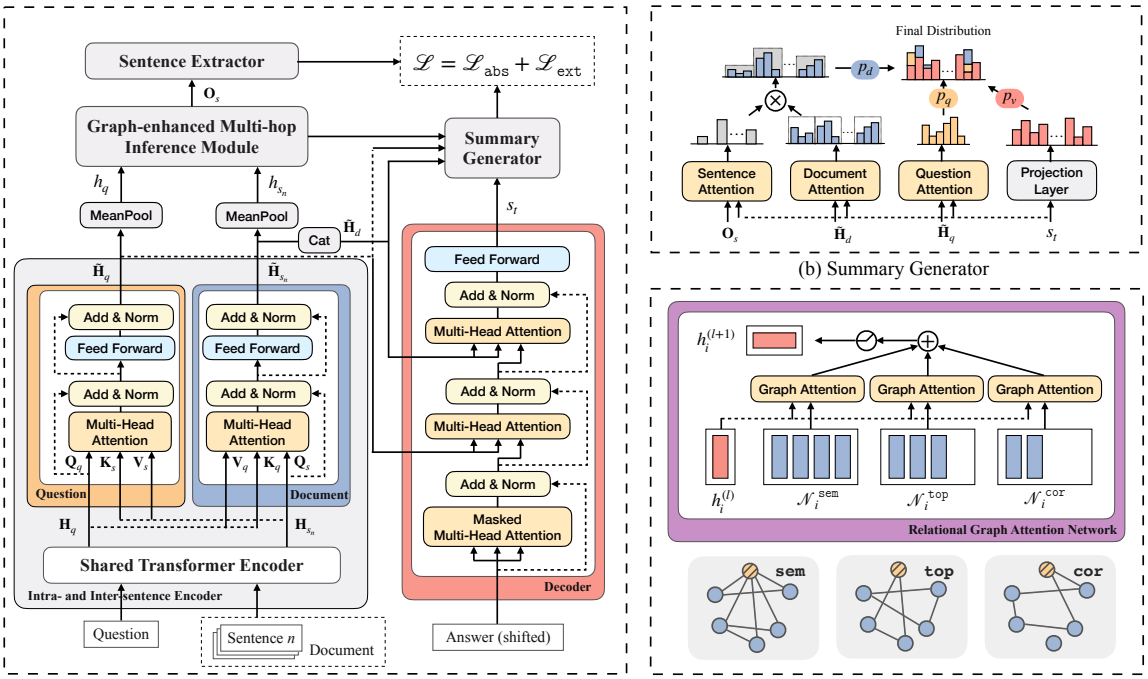
C. Multi-hop Reasoning in QA

One of the challenges for applying neural models on QA systems is that it is required to preserve the capability of reasoning for the aggregation of multiple evidence facts in order to answer complex natural language questions [28], [57]. Many attempts have been made on learning to provide evidence or justifications for a human-understandable explanation of the multi-hop inference process in factoid QA [20], [21], [58], [59], where the inferred evidences are only treated as the middle steps for finding the answer. However, in non-factoid QA, the intermediate output is also important to form a complete answer, which requires a bridge between the multi-hop inference and summarization [22].

Performing explicit multi-hop reasoning on graph structure has been demonstrated to be an effective approach for multi-hop factoid QA [26]–[29], [60] and some other text generation tasks [61], [62]. The multi-hop reasoning modules in these works mainly focus on linking entities among sentences. In this work, we investigate the utility of graph-enhanced multi-hop inference to capture three types of semantic relations in non-factoid QA systems.

D. Text Summarization

The methods for text summarization are generally categorized into extractive and abstractive approaches. The extractive methods [63], [64] produce a summary by extracting salient sentences from the source document, while the abstractive methods [65]–[67] generate a summary from the vocabulary based on the understanding of the document. In addition, researchers attempt to take advantages of both extractive and abstractive methods by using hybrid techniques, such as joint learning [68], [69], extract-then-abstract [70], [71]. On the other hand, many efforts have been made on exploiting the utilities of graph structures to capture relations between textual units for benefiting summarization [25], [72], [73].



(a) Overview of GMQS (b) Summary Generator (c) Graph-enhanced Multi-hop Inference Module

Fig. 2. Overview of GMQS.

Pretrained language models, such as BERT [74], BART [75], recently emerge for achieving impressive improvements in text summarization. In this work, we make the first attempt of jointly learning the extractive and abstractive query-focused summarization.

III. Problem Definition

The input of both extractive and abstractive query-focused summarization contains a sequence of words $\{w_1^q, w_2^q, ..., w_{m_q}^q, ...\}$ for the query q and a sequence of words $\{w_1^d, w_2^d, ..., w_{m_d}^d, ...\}$ for the document d, where m_q and m_d are the word indexes. The sequence of words in a document can also be represented as a sequence of sentences $\boldsymbol{s} = \{s_1, s_2, ..., s_n, ...\}$, where n is the sentence index. The goal of both extractive and abstractive query-focused summarization is to produce a summary y, based on the query q and the document d. Without the loss of generality, we refer the term "query" as "question" and the term "summary" as "answer" in the following description of non-factoid QA.

Non-factoid Question Answering as Extractive Query-focused Summarization: The output of extractive query-focused summarization is a sequence of predicted probability $\{\tilde{y}^s\}$ for each sentence in the document d, where $\tilde{y}_n^s$ represents the probability of the n-th sentence been extracted into the answer y. The goal is to learn a sentence-level sequence labeling model $f_e(\cdot)$ to determine which sentences should be included to form the final answer:

$$f_e(q, d) = \{\tilde{y}_1^s, \tilde{y}_2^s, ..., \tilde{y}_n^s, ...\}. \tag{1}$$

Non-factoid Question Answering as Abstractive Query-focused Summarization: The output of abstractive query-focused summarization is a sequence of predicted probability of vocabulary distribution P_t at each time-step t. The goal is to learn an auto-regressive sequence-to-sequence model $f_a(\cdot)$ to generate new sentences to form the final answer:

$$f_a(q, d, y_{<t}) = P_t. \tag{2}$$

IV. Method

We introduce the proposed method, namely Graph-enhanced Multi-hop Query-focused Summarizer (GMQS), for non-factoid question answering. Figure 2 depicts the overall architecture of GMQS, which contains four main components:

- **Intra- and Inter-sentence Encoder** reads the sentences of both the question and document by capturing semantic relationships from sentences themselves as well as interactions between the question and document.
- **Graph-enhanced Multi-hop Inference Module** elaborates a multi-relational graph structure to perform multi-hop reasoning over the whole document by taking into account three types of semantic relations.
- **Sentence Extractor** scores each sentence in the document according to the learned sentence representation.
- **Summary Generator** produces the abstractive summary as the answer to the given question.

A. Intra- and Inter-sentence Encoder

Unlike the encoder of traditional summarization models, which only needs to establish explicit representations for a single sentence, query-focused summarization is further required to capture the interaction between the question and the document. To achieve this, the encoder is designed to be capable of modeling intra- and inter-sentence interactions.

We adopt multi-head self-attention module from Transformer [76] as the basic unit for encoding the raw text into semantic sentence representations. The multi-head attention unit

is denoted as $\textbf{MHAtt}(Q, K, V)$, where Q, K, V are query, key, and value, respectively. Each multi-head attention unit consists of three components: (i) The Scale Dot-Product Attention to apply attention weights upon the value vector with size of d_h; (ii) The feed-forward network with ReLU activation, which is defined as $\textbf{FFN}(\cdot)$; (iii) The layer normalization, which is defined as $\textbf{LayerNorm}(\cdot)$. Generally, a multi-head attention unit can be represented as:

$$V_{att} = \textbf{softmax}\left(QK^T/\sqrt{d_h}\right)V \tag{3}$$

$$\textbf{MHAtt}(Q, K, V) = \textbf{LayerNorm}(\textbf{FFN}(V_{att}) + V). \tag{4}$$

Given the question q and the document d that consists of a sequence of sentences $\boldsymbol{s} = \{s_1, s_2, ..., s_n\}$, we first use the self-attention to compute the representations of the question and each document sentence separately:

$$H_q = \textbf{MHAtt}(E(q), E(q), E(q)), \tag{5}$$

$$H_{s_n} = \textbf{MHAtt}(E(s_n), E(s_n), E(s_n)), \tag{6}$$

where $E(\cdot)$ is the embeddings of the input text, which is the concatenation of word and position embeddings. Such intra-sentence interaction attends the important information within the question and each individual document sentence.

After obtaining the encoded representations for all the input sequences, we perform the cross-attention to capture the semantically relevant information between the question and each document sentence:

$$\tilde{H}_q = \frac{1}{N}\sum\nolimits_{n=1}^{N} \textbf{MHAtt}(H_{s_n}, H_q, H_q), \tag{7}$$

$$\tilde{H}_{s_n} = \textbf{MHAtt}(H_q, H_{s_n}, H_{s_n}), \tag{8}$$

where $\tilde{H}_q$ and $\tilde{H}_{s_n}$ are the attentive representations for the word sequences of the question and each document sentence, respectively. Then, meaning pooling operation is applied to obtain the final encoded sentence representations:

$$h_q = \textbf{MeanPool}(\tilde{H}_q), \quad h_{s_n} = \textbf{MeanPool}(\tilde{H}_{s_n}). \tag{9}$$

B. Graph-enhanced Multi-hop Inference Module

Graph-enhanced Multi-hop Inference Module measures the degree of importance of each sentence in the document for producing the answer, through a multi-hop reasoning procedure, which is based on the graph structure and three types of semantic and linguistic relations, namely **Semantic Relevance**, **Topical Coherence**, and **Co-reference Linking**.

1) Multiple Semantic Relations: We first introduce the three types of semantic and linguistic relations as the backbone of the Graph-enhanced Multi-hop Inference Module:

(1) **Semantic Relevance**. There are two kinds of semantic relevance to be considered for the multi-hop inference in non-factoid QA. The first one is the relevance degree between the question and each sentence in the document, which is also the essential measurement in answer sentence selection studies [12], [13]. The other one is the information-consistency between the concerned sentence and those highly weighted sentences from the previous hops [22]. Therefore, motivated by Maximal Absolute Relevance (MAR) measurement in [22], we elaborate the relation of semantic relevance between: (i) the question and each sentence in the document, and (ii) the sentence and the most similar sentence in the document.

(2) **Topical Coherence**. Despite the content transition in the multi-hop inference process, the concerned latent topic is supposed to be coherent for collecting the information to answer the given question [23], [24]. To capture the relation of topical coherence, we leverage LDA topic model [77] to identify the latent topic of each sentence in the document. The sentences estimated with the same latent topic are taken into consideration for modeling the topical coherence.

(3) **Coreference Linking**. Resolving long-term coreference is of great importance in multi-hop question answering [26], since the question is often concerning about some certain objects. Instead of implicitly modeling the long-term coreference, we employ a state-of-the-art coreference resolution tool, *NeuralCoref*, to link the coreference objects among the question and all sentences in the document.

2) Multi-relational Graph Construction: To facilitate the reasoning process, it requires to model and aggregate the complex relations with multiple hops of refinement. To this end, we construct a multi-relational graph to represent the relational information obtained from different relational inference units. The multi-relational graph is denoted as $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathcal{R})$, with nodes $n_i \in \mathcal{N}$, labeled edges (i.e., relations) between node n_i and n_j as $(n_i, r, n_j) \in \mathcal{E}$, where $r \in \mathcal{R}$ is the relation type between two nodes. We treat the question q, each document sentence s_n as a node in $\mathcal{G}$, with the total number of nodes as $1 + |\boldsymbol{s}|$. We initialize each node with their corresponding encoded sentence representations h_* obtained from the encoder described in Section IV-A.

To represent the multi-relational information obtained from all the relational inference units, we employ different adjacency matrices for the graph $\mathcal{G}$. Specifically, the relation types between two nodes is denoted as $r \in \mathcal{R} = \{\texttt{sem}, \texttt{top}, \texttt{cor}\}$, representing the relations of **Semantic Relevance**, **Topical Coherence**, and **Coreference Linking**, respectively. Three adjacency matrices can thus be constructed for $\mathcal{G}$:

$$\boldsymbol{A}^{\texttt{sem}}_{i,j} = \begin{cases} 1, & \text{if } n_i = q, n_j \in \boldsymbol{s}, \\ 1, & \text{if } n_i \in \boldsymbol{s}, n_j = \underset{n_j \in \boldsymbol{s} \setminus n_i}{\arg\max}\, \textbf{Sim}(n_i, n_j), \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

$$\boldsymbol{A}^{\texttt{top}}_{i,j} = \begin{cases} 1, & \text{if } n_i, n_j \in \mathcal{N}, \textbf{LDA}(n_i) = \textbf{LDA}(n_j), \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

$$\boldsymbol{A}^{\texttt{cor}}_{i,j} = \begin{cases} 1, & \text{if } n_i, n_j \in \mathcal{N}, \textbf{Coref}_{\mathcal{N}}(n_i, n_j) \neq \emptyset, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where $\textbf{Sim}(\cdot)$ denotes the semantic similarity function, which is based on tf-idf cosine similarity between sentences to capture lexical similarity. $\textbf{LDA}(\cdot)$ denotes the predicted latent topic by the LDA topic model. $\textbf{Coref}_{\mathcal{N}}(\cdot)$ represents the shared coreference clusters between two sentences, which is resolved from all the sentences in $\mathcal{N}$.

3) Multi-hop Information Aggregation: In order to capture the information from multiple semantic relations with a multi-hop inference process, we investigate the utilities of two kinds of graph neural networks, namely Relational Graph Convo-

lutional Network (R-GCN) and Relational Graph Attention Network (R-GAT).

Relational Graph Convolutional Network. R-GCN [78] has the capability of aggregating multiple relations between entities in a knowledge graph for the link prediction task, which can also be extended to model the multiple semantic relations for the multi-hop information aggregation in non-factoid QA. For a node n_i in $\mathcal{G}$, the multi-relational information is aggregated from its neighboring nodes:

$$h_i^{(l+1)} = \sigma\left(\sum\nolimits_{r\in\mathcal{R}}\sum\nolimits_{j\in\mathcal{N}_i^r}\widehat{\boldsymbol{A}}_{i,j}^r\boldsymbol{W}_r^{(l)}h_j^{(l)}\right), \tag{13}$$

where $h_i^{(l)}$ is the hidden state of the node n_i at the l-th layer of the network, $\mathcal{N}_i^r$ denotes the neighboring indices of the node n_i under the relation r (including node n_i itself), $\boldsymbol{W}_r^{(l)} \in \mathbb{R}^{|\mathcal{N}|\times d_h}$ are trainable parameters representing the transformation from neighboring nodes and from the node n_i itself. $\sigma(\cdot)$ denotes the activation function, such as $\mathrm{ReLU}(x) = \max(0, x)$. $\widehat{\boldsymbol{A}}_{i,j}^r$ is a normalization constant, such as $\widehat{\boldsymbol{A}}_{i,j}^r = 1/|\mathcal{N}_i^r|$ in [78]. To avoid the scale changing of the feature representation, we apply a symmetric normalization transformation:

$$\widehat{\boldsymbol{A}}^r = \boldsymbol{D}_r^{-1/2}\boldsymbol{A}^r\boldsymbol{D}_r^{-1/2},\ r\in\{\texttt{sem},\texttt{top},\texttt{cor}\}, \tag{14}$$

where $\boldsymbol{A}^r$ is the adjacency matrix described in Section IV-B2 under the relation $r \in \mathcal{R}$, $\boldsymbol{D}_r$ is the corresponding degree matrix of $\boldsymbol{A}^r$ as $\boldsymbol{D}_{rii} = \sum_j \boldsymbol{A}_{i,j}^r$.

Relational Graph Attention Network. Despite the success of considering multi-relational information in the graph, R-GCN also inherits some limitations from the original GCN. As opposed to GCN, Graph Attention Network (GAT) [33] is proposed to assign different importance to neighbors of the node, instead of using the fixed or pre-defined edge weights. Motivated by the advantages of GAT and R-GCN, we further extend R-GCN to be Relational Graph Attention Network (R-GAT) for enhancing the multi-hop inference process.

Following the graph attention mechanism proposed in [33], the attention weight $\alpha_{i,j}$ indicates the importance of node j's features to node i. For each relation $r \in \mathcal{R}$, we compute the relation-specific attention weights $\alpha_{i,j}^r$ as:

$$\alpha_{i,j}^r = \frac{\exp\left(\mathrm{LeakyReLU}(\widehat{\boldsymbol{A}}_{i,j}^r\boldsymbol{\omega}_r^\top[\boldsymbol{W}_r h_i||\boldsymbol{W}_r h_j])\right)}{\sum\limits_{k\in\mathcal{N}_i^r}\exp\left(\mathrm{LeakyReLU}(\widehat{\boldsymbol{A}}_{i,k}^r\boldsymbol{\omega}_r^\top[\boldsymbol{W}_r h_i||\boldsymbol{W}_r h_k])\right)}, \tag{15}$$

where $\omega_r \in \mathbb{R}^{2d_h'}$ and $\boldsymbol{W}_r \in \mathbb{R}^{d_h'\times d_h}$ are parameters to be learnt for relation r. $||$ denotes the concatenation operation. The LeakyReLU activation function is applied for nonlinearity.

The graph attention mechanism can be extended to employ multi-head attention, similar to [76]. Specifically, K independent attention weights can be calculated based on Equation (15), resulting in the following output node representation for the next layer:

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i^r}\alpha_{i,j}^{r,k,(l)}\widehat{\boldsymbol{A}}_{i,j}^r\boldsymbol{W}_{r,k}^{(l)}h_j^{(l)}\right), \tag{16}$$

where $\alpha_{i,j}^{r,k,(l)}$ are normalized attention coefficients computed by the k-th head of attention for relation r, and $W_{r,k} \in \mathbb{R}^{d_h'\times d_h}$ is the corresponding linear transformation matrix to be learnt. In particular, we denote the output node representations in the last layer of the graph neural network as o_q and o_{s_n} for the question and each document sentence, respectively:

$$o_q = h_q^{(L_G)},\quad o_{s_n} = h_{s_n}^{(L_G)}, \tag{17}$$

where L_G is the number of graph layers. And the number of graph layers can be regarded as the number of reasoning hops, since each graph layer only consider the relation between two adjacent sentences in the graph, while multiple graph layers can collectively measure the interrelations among multi-hop connected sentences in the graph.

C. Sentence Extractor

After obtaining the sentence vectors from Graph-enhanced Multi-hop Inference Module, we build a summarization-specific classifier to extract summaries based on the multi-hop inference results. The classifier contains a linear transformation and the sigmoid function:

$$\tilde{y}_s = \sigma(W_e^\top o_s + b_e), \tag{18}$$

where $\sigma(\cdot)$ denotes the sigmoid function, $W_e \in \mathbb{R}^{d_h'\times 2}$ and $b_e \in \mathbb{R}^2$ are parameters to be learnt. The extractive query-based summarization is based on the ranked $\tilde{y}_s$ to extract sentences.

D. Summary Generator

We obtain the token-level representations $\tilde{H}_q$ and $\tilde{H}_{s_n}$ from the encoding phase, and the sentence-level document representation o_q and o_{s_n} via the graph-enhanced multi-hop inference module for the question and each document sentence, respectively.

Similar to the encoder, we adopt Transformer decoder layer for decoding. The difference is that the decoder takes into account two sources of information, including the question and the document. For each decoder layer:

$$X_a = \textbf{MHAtt}(E(a), E(a), E(a)), \tag{19}$$

$$X_c = \textbf{MHAtt}([\tilde{H}_q||\tilde{H}_d], X_a, X_a), \tag{20}$$

$$S_{\text{dec}} = \textbf{FFN}(X_c), \tag{21}$$

where $E(a)$ denotes the masked answer embedding, and S_{dec} is the hidden states produced by the Transformer decoder layer. We concatenate all the token-level document sentence representations to be the token-level document representations as $\tilde{H}_d = ||_n\tilde{H}_{s_n}$.

Let s_t denote the hidden state of the decoder at the t-th step. The attention for each word in the question and the document, α_t^q and α_t^d, are generated by:

$$e_t^{q_j} = \omega_t^{qT}\tanh(W_q\tilde{H}_{q_j} + W_{qs}s_t + b_q), \tag{22}$$

$$\alpha_t^q = \mathrm{softmax}(e_t^q), \tag{23}$$

$$e_t^{d_i} = \omega_t^{dT}\tanh(W_d\tilde{H}_{d_i} + W_{ds}s_t + b_d), \tag{24}$$

$$\alpha_t^d = \mathrm{softmax}(e_t^d), \tag{25}$$

where $W_q \in \mathbb{R}^{d'_h \times d_h}$, $W_{qs} \in \mathbb{R}^{d'_h \times d_h}$, $W_d \in \mathbb{R}^{d'_h \times d_h}$, $W_{ds} \in \mathbb{R}^{d'_h \times d_h}$, $\omega_t^q \in \mathbb{R}^{d'_h}$, $\omega_t^d \in \mathbb{R}^{d'_h}$, $b_q \in \mathbb{R}^{d'_h}$, $b_d \in \mathbb{R}^{d'_h}$ are parameters to be learned.

Then, we incorporate the multi-hop inference results $O_s = \{o_{s_1}, ..., o_{s_n}\}$ to compute the dynamic multi-hop reasoning gate β_t for each sentence in the document:

$$\beta_t = \sigma(\omega_t^{sT} \tanh(W_s O_s + W_{ss} s_t + b_s)), \tag{26}$$

where $W_s \in \mathbb{R}^{d'_h \times d_h}$, $W_{ss} \in \mathbb{R}^{d'_h \times d_h}$, $\omega_t^s \in \mathbb{R}^{d'_h}$, $b_s \in \mathbb{R}^{d'_h}$ are parameters to be learned. We re-weight the word-level document attention scores α^d with a soft multi-hop reasoning gate β to attend important justification sentences along with the decoding process:

$$\hat{\alpha}_t^{d_i} = \frac{\alpha_t^{d_i} \beta_{t, d_i \in s_k}}{\sum_i \alpha_t^{d_i} \beta_{t, d_i \in s_k}}. \tag{27}$$

Thus, the re-weighted word-level document attention $\hat{\alpha}^d$ naturally blends with the results from the multi-hop inference module to enhance the influence of those important justification sentences.

Finally, we extend the basic pointer-generator network [65] to be a multi-pointer architecture to generate answers with the dynamic multi-hop reasoning flow as well as handle the out-of-vocabulary (OOV) issue. Such approach enables GMQS to copy words from the question as well as be aware of the differential importance degree of different sentences in the document. The attention weights α_t^q and $\hat{\alpha}_t^d$ are used to compute context vectors c_t^q and c_t^d as the probability distribution over the source words:

$$c_t^q = \tilde{H}_q^T \alpha_t^q, \quad c_t^d = \tilde{H}_d^T \hat{\alpha}_t^d. \tag{28}$$

The context vector aggregates the information from the source text for the current step. We concatenate the context vector with the decoder state s_t and pass through a linear layer to generate the answer representation h_t^s:

$$h_t^s = W_1[s_t || c_t^q || c_t^d] + b_1, \tag{29}$$

where $W_1 \in \mathbb{R}^{d'_h \times 3d_h}$ and $b_1 \in \mathbb{R}^{d'_h}$ are parameters to be learned.

Then, the probability distribution P^v over the fixed vocabulary is obtained by passing the answer representation h_t^s through a softmax layer:

$$P^v(y_t) = \text{softmax}(W_2 h_t^s + b_2), \tag{30}$$

where $W_2 \in \mathbb{R}^{|V| \times d_h}$ and $b_2 \in \mathbb{R}^{|V|}$ are parameters to be learned, and $|V|$ denotes the vocabulary size.

The final probability distribution of y_t is obtained from three views of word distributions:

$$P^q(y_t) = \sum\nolimits_{i:w_i=w} \alpha_t^{q_i}, \quad P^d(y_t) = \sum\nolimits_{i:w_i=w} \hat{\alpha}_t^{d_i}, \tag{31}$$

$$P^{all}(y_t) = [P^v(y_t), P^q(y_t), P^d(y_t)], \tag{32}$$

$$\rho = \text{softmax}(W_\rho[s_t || c_t^q || c_t^d] + b_\rho), \tag{33}$$

$$P_t(y_t) = \rho \cdot P^{all}(y_t), \tag{34}$$

where $W_\rho \in \mathbb{R}^{3 \times d_h}$ and $b_\rho \in \mathbb{R}^3$ are parameters to be learned, ρ is the multi-pointer scalar to determine the weight of each view of the probability distribution.

TABLE I
STATISTICS OF DATASET.

Dataset (train/dev/test)	WikiHow	PubMedQA
#Samples	168K / 6K / 6K	169K / 21K / 21K
Avg QLen	7.00 / 7.02 / 7.01	16.3 / 16.4 / 16.3
Avg DLen	582 / 580 / 584	238 / 238 / 239
Avg ALen	62.2 / 62.2 / 62.2	41.0 / 41.0 / 40.9
Avg #Sents/Doc	20.7 / 20.7 / 20.6	9.32 / 9.31 / 9.33

E. Training Procedure

After obtaining $\tilde{y}_s$ from the sentence extractor, we use the cross entropy as the objective function for extractive query-focused summarization:

$$\mathcal{L}_{\text{ext}} = -\frac{1}{N} \sum\nolimits_{n=1}^{N} \left(y_n^s \log \tilde{y}_n^s + (1 - y_n^s) \log (1 - \tilde{y}_n^s)\right), \tag{35}$$

where y_n^s is the ground-truth label of the n-th sentence been extracted into the answer y.

With $P_t(y_t)$ from the summary generator, we train the abstractive query-focused summarization to minimize the negative log-likelihood:

$$\mathcal{L}_{\text{abs}} = -\frac{1}{T} \sum\nolimits_{t=1}^{T} \log P_t(y_t), \tag{36}$$

where y is the ground-truth answer.

In order to mutually enhance both extractive and abstractive summarization, the proposed model can be jointly trained by:

$$\mathcal{L} = \mathcal{L}_{abs} + \lambda \mathcal{L}_{ext}, \tag{37}$$

where $\lambda \geq 0$ is a hyper-parameter for balancing the ratio between two losses.

V. EXPERIMENTAL SETUP

A. Dataset & Evaluation Metrics

We evaluate the proposed method on two non-factoid QA datasets with abstractive answers, namely WikiHow [79] and PubMedQA [6]. WikiHow is an abstractive summarization dataset collected from a community-based QA website, *WikiHow*[1], in which each sample consists of a non-factoid question, a long article, and the abstractive summary as the answer to the given question. An actual sample is presented in Fig. 6. PubMedQA is a conclusion-based biomedical QA dataset collected from *PubMed*[2] abstracts, in which each instance is composed of a question, a context, and an abstractive answer which is the summarized conclusion of the context corresponding to the question. An actual sample is presented in Fig. 1. The statistics of the WikiHow and PubMedQA datasets are shown in Table I. We adopt ROUGE F1 (R1, R2, RL) for automatically evaluating the summarized answers.

[1]https://www.wikihow.com
[2]https://www.ncbi.nlm.nih.gov/pubmed/

B. Compared Methods

There are four results of our method, GMQS, as follows:

- **GMQS-ext** and **GMQS-abs** only use the single-task learning loss, *i.e.*, Eq. (35) or Eq. (36), to train an extractive or abstractive summarizer, respectively.
- **GMQS-ext-joint** and **GMQS-abs-joint** use the joint learning loss, *i.e.*, Eq. (37), to train the overall framework, but adopt the output from the sentence extractor or the summary generator, respectively.

To make comprehensive comparisons, we compare our method to three different groups of state-of-the-art methods, including non-factoid question answering, traditional summarization, and query-focused summarization methods.

As for non-factoid QA methods, we adopt both retrieval- and generation-based methods for comparisons, where the retrieval-based methods perform a sentence-level classification task to determine whether the sentence should be selected:

- **Compare-Aggregate Model (CA)** [13] aggregates the comparison results in small units of two sentences;
- **COALA** [14] selects answers via the comparison of all question-answer aspects;
- **BERT** [80] adopts the pairwise fine-tuning to perform answer sentence selection;
- **HGN** [27] adopts a hierarchical graph to model different levels of granularity for multi-task learning in factoid QA. In our case, we apply it as an answer selection model;
- **MHPGM** [26] uses multiple hops of bidirectional attention and a pointer-generator decoder to read and reason within a long passage for generating the answer;
- **S2S-MT** [4] uses a multi-task Seq2Seq model with the concatenation of question and support document;
- **QPGN** [3] is a question-driven pointer-generator network with co-attention between the question and document.

As for traditional summarization methods, we also adopt both extractive and abstractive methods as well as hybrid methods for comparisons, where the question and the document are concatenated as the input for these methods:

- **NeuralSum** [63] performs extractive summarization as a sequence labeling task;
- **NeuSum** [64] jointly learns to score and select sentences for extractive summarization;
- **PGN** [65] copies words from the article via pointing, and produces novel words by the generator;
- **CopyTransformer** [66] incorporates the copy mechanism into the Transformer [76] for abstractive summarization;
- **UnifiedSum** [68] is a unified model combining sentence-level and word-level attentions to take advantage of both extractive and abstractive summarization approaches;
- **MGSum** [69] uses a multi-granularity interaction network to encode input documents and unifies extractive and abstractive summarization into one architecture.
- **BERTSum** [74] is a BERT-based general framework encompassing both extractive and abstractive summarization, namely BERTSumExt and BERTSumAbs.

Similarly, we compare the proposed method to both extractive and abstractive query-focused summarization methods:

- **MMR** [47] applies classical Maximal Marginal Relevance algorithm for query-based summarization;
- **AttSum** [19] applies the attention mechanism to simulate the human-like reading when a query is given;
- **HSCM** [55] integrates the hierarchical interaction information between the question and document into a sequential extractive summarization model;
- **QS** [17] utilizes the query information into the pointer-generation network;
- **SD$_2$** [18] combines a query-based attention model and a diversity-based attention model;
- **MSG** [22] incorporates multi-hop reasoning into question-driven summarization.

C. Implementation Details

Following the general settings [76], we apply a six-layer encoder and a two-layer decoder for all Transformer based models. The input embedding size and the hidden size are set to be 512. The word embeddings are randomly initialized. The size of the Transformer FFN inner representation size is set to be 2048, and ReLU is used as the activation function. The learning rate and the dropout rate are set to be 0.0001 and 0.1, respectively. During training, the batch size is set to be 32, while at the inference phase, we use beam search with a beam size of 10. For each model, we all train for 20 epochs. We adopt the NLTK package [81] for sentence and word tokenization. The maximum length of each sentence and the maximum number of sentences in each document are set to be 32 and 16, respectively. As for the extractive summarization setting, we follow previous studies [63], [64] to select top-3 scored sentences to construct the summary. As for the abstractive summarization setting, we also follow previous studies [22] to restrict the length of the generated summary within the range of 30 and 100. λ is set to 0.5, which is tuned on the validation set. For the graph construction, *GenSim*[3] is adopted to implement the Tf-idf and LDA models, while *NeuralCoref*[4] is adopted as the coreference resolution tool.

VI. Results & Analysis

A. Overall Performance on Extractive Methods

Table II presents the experimental results of extractive methods on WikiHow and PubMedQA datasets. Among the baseline methods, extractive summarization methods perform better than answer sentence selection methods on WikiHow. Even the heuristic unsupervised method, LEAD3, achieves a better performance than these sophisticated answer sentence selection methods on WikiHow. However, all kinds of baselines have a similar performance on PubMedQA. As known from the dataset statistics in Table I, the average question length in WikiHow is relatively short, where the inadequate information in the question restricts the interactive context modeling between question and answer sentences. Overall, the proposed method, GMQS, substantially and consistently

[3]https://radimrehurek.com/gensim/

[4]https://github.com/huggingface/neuralcoref

TABLE II
EXPERIMENTAL RESULTS ON EXTRACTIVE METHODS.

Model	WikiHow			PubMedQA		
	R1	R2	RL	R1	R2	RL
LEAD3	26.0	7.2	24.3	30.9	9.8	21.2
CA [13]	24.5	6.0	22.6	31.2	9.6	24.5
COALA [14]	26.1	6.2	23.7	31.6	9.8	25.6
BERT [80]	27.1	6.6	24.1	32.0	10.2	25.9
HGN [27]	26.3	6.3	23.9	31.5	9.8	25.5
NeuralSum [63]	26.7	6.4	24.0	30.9	9.7	22.4
NeuSum [64]	26.5	6.2	23.8	31.0	9.7	22.5
MGSum-ext [69]	27.4	7.1	24.4	32.0	10.5	26.1
BERTSumExt [74]	27.7	7.4	25.0	32.2	10.4	26.3
MMR [47]	26.8	6.1	23.6	30.1	9.0	24.4
AttSum [19]	26.4	6.3	24.0	31.2	9.8	25.3
HSCM [55]	27.2	7.0	24.7	32.3	10.1	26.0
GMQS-ext	28.6	7.9	26.1	33.2	11.8	27.6
GMQS-ext-joint	**29.0**	**8.1**	**26.4**	**33.5**	**11.9**	**27.7**

TABLE III
EXPERIMENTAL RESULTS ON ABSTRACTIVE METHODS.

Model	WikiHow			PubMedQA		
	R1	R2	RL	R1	R2	RL
LEAD3	26.0	7.2	24.3	30.9	9.8	21.2
MHPGM [26]	28.0	9.4	27.1	34.0	12.5	28.4
S2S-MT [4]	28.6	9.6	27.5	33.2	12.2	27.8
QPGN [3]	28.8	9.7	27.7	34.2	12.8	28.7
PGN [65]	28.5	9.2	26.5	32.9	11.5	28.1
CopyTransformer [66]	30.2	10.0	28.8	35.0	11.3	27.8
Unified [68]	30.0	9.9	28.7	35.7	12.1	29.0
MGSum-abs [69]	30.4	10.4	29.4	37.0	13.9	30.0
BERTSumAbs [74]	30.4	10.2	29.1	37.5	15.0	30.3
QS [17]	28.8	9.9	27.6	32.6	11.1	26.7
SD_2 [18]	27.7	7.9	25.8	32.3	10.5	26.0
MSG (3-Hop) [22]	30.5	10.5	29.3	37.2	14.8	30.2
GMQS-abs	31.5	11.2	30.7	38.1	15.3	31.0
GMQS-abs-joint	**32.2**	**11.6**	**31.2**	**38.8**	**15.7**	**31.6**

outperforms all the extractive methods, including answer sentence selection, traditional and query-focused summarization methods, by a noticeable margin on the two datasets. Even training from scratch, GMQS can achieve competitive performance with BERT-based methods, including BERT for answer sentence selection and extractive summarization. This result demonstrates the superiority of the proposed graph-enhanced multi-hop inference method on identifying the important sentences with salient as well as question-related information for extractive non-factoid QA. In addition, the joint learning with abstractive summarization further improves the extraction performance of GMQS.

B. Overall Performance on Abstractive Methods

Experimental results of abstractive methods are summarized in Table III. There are several notable observations as follows:

(1) Compared with extractive methods, all kinds of abstractive methods perform with more promising results, which indicates that answers for non-factoid questions include sparse and diverse information from different sentences across the whole supporting document or evidences. It is not enough to simply extract or select original sentences from the document.

(2) MSG and the proposed GMQS, which both consider the interrelationships among different document sentences by multi-hop reasoning, outperform other baseline methods with a substantial margin. This result shows that the multi-hop inference attaches great importance in non-factoid QA. GMQS further improves the performance over MSG by capturing more comprehensive semantic relationships during the multi-hop inference process.

(3) As for the performance boosting by the joint learning, the extractive learning makes more contribution to the abstractive learning than the reverse, since the learned importance degree of each sentence casts a direct impact on the generated sentences, according to Equation (27).

In both extractive and abstrative scenario, the proposed GMQS method substantially and consistently outperforms those strong baselines, which demonstrates not only the effectiveness of the graph-enhanced multi-hop inference on non-factoid QA, but also its promising applicability.

TABLE IV
HUMAN EVALUATION RESULTS. THE FLEISS' KAPPA OF THE ANNOTATIONS IS 0.42, WHICH INDICATES "MODERATE AGREEMENT".

Model	Info.	Conc.	Read.	Corr.
COALA	3.05	2.15	3.85	3.01
MGSum-ext	3.19	**2.21**	**4.01**	3.14
HSCM	3.33	2.09	3.87	3.32
GMQS-ext-joint	**3.41**	2.14	3.95	**3.56**
QPGN	3.53	3.45	3.61	3.30
MGSum-abs	3.98	**4.10**	4.12	3.48
MSG	4.07	3.75	3.80	3.72
GMQS-abs-joint	**4.21**	4.02	**4.14**	**3.89**

C. Human Evaluation

We conduct human evaluation to evaluate the generated answer from four aspects: (1) **Informativity**: how rich is the generated answer in information? (2) **Conciseness**: how concise the generated answer is? (3) **Readability**: how fluent and coherent the generated answer is? (4) **Correctness**: how well does the generated answer respond to the given question? We randomly sample 50 questions from two datasets and compare their answers produced by three extractive (COALA, MGSum-ext, HSCM) and three abstractive summarization methods (QPGN, MGSum-abs, and MSG). Three annotators are asked to score each generated answer with 1 to 5 (higher the better). Results are presented in Table IV. These annotators are all well-educated research assistants with a background of NLP and are all native speakers. The ground-truth answers are provided for evaluating the **Correctness** of the genereted answers. As for both extractive and abstractive methods, GMQS substantially outperforms existing methods on producing informative and correct answers, and preserving high-level conciseness and readability as well.

D. Ablation Study

1) Comparisons on Multi-hop Inference Module: In order to validate the superiority of the proposed graph-enhanced

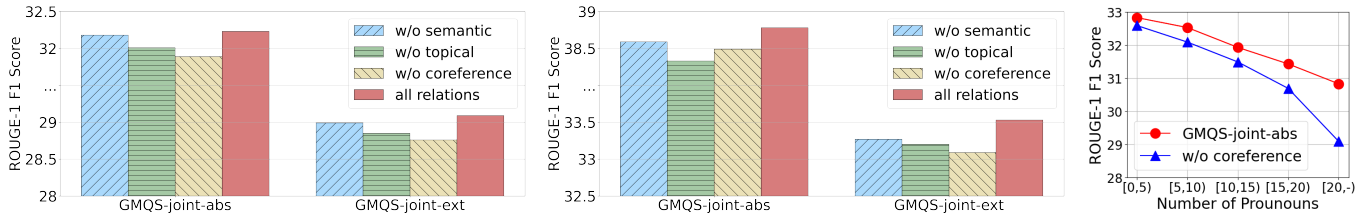


Fig. 3. Impact of different semantic relations.

TABLE V
COMPARISONS ON MULTI-HOP INFERENCE MODULES.

Model	WikiHow			PubMedQA		
	R1	R2	RL	R1	R2	RL
GMQS-ext-joint	**29.0**	**8.1**	**26.4**	**33.5**	**11.9**	**27.7**
- w/ RGCN	28.4	7.7	25.9	33.3	11.7	27.5
- w/ MHPGM	28.0	7.5	25.0	32.3	11.0	26.6
- w/ MSG	28.1	7.4	25.0	32.4	11.3	26.9
- w/ HGN	27.9	7.4	24.9	32.2	11.0	26.5
- w/o Multi-hop	27.7	7.3	24.7	32.0	10.8	26.3
GMQS-abs-joint	**32.2**	**11.6**	**31.2**	**38.8**	**15.7**	**31.6**
- w/ RGCN	31.6	11.1	30.7	38.6	15.4	31.4
- w/ MHPGM	31.3	10.8	30.3	37.5	14.4	30.4
- w/ MSG	31.5	10.8	30.3	38.2	14.8	30.8
- w/ HGN	31.0	10.6	29.9	37.6	14.4	30.5
- w/o Multi-hop	30.9	10.6	29.8	37.2	14.1	30.3

multi-hop inference module, we conduct comparisons with other alternative multi-hop inference components as follows:

- We first substitute RGAT with RGCN [78] for the aggregation of multi-relational information, i.e., w/ RGCN.
- Another way is to use the self-attention layer [76] to construct a fully-connected sentence graph for node representation learning, which is similar to the multi-hop reasoning module in MHPGM [26], i.e., w/ MHPGM.
- We alsp adopt the multi-hop inference module proposed in MSG [22], which elaborates the semantic relevance between the question and each document sentence as well as among all the document sentences, i.e., w/ MSG.
- The last one is to adapt the Hierarchical Graph Network (HGN) from [27] into non-factoid QA, which aggregates different granularity of information for multi-hop inference, i.e., w/ HGN.
- We also consider the situation when the multi-hop inference module is discarded, i.e., w/o Multi-hop.

The comparison results are presented in Table V. For all kinds of multi-hop inference modules, they contribute to better performance on both extractive and abstractive results more or less, showing the necessity of the multi-hop reasoning on non-factoid QA. The constructed multi-relational graph further enables the multi-hop inference module to capture diverse and complex interrelationships among sentences, leading to a higher performance of using RGCN and RGAT for graph representational learning. Overall, the proposed RGAT achieves the best performance among these alternative multi-hop inference modules.

2) Impact of Different Semantic Relations: To elaborate the multi-hop inference upon different reasoning paths, we model the multiple semantic relations between the question and the document sentences as well as among the document sentence. Thus, we examine the effect of each semantic relation during the multi-hop inference procedure in terms of discarding each one of these relational graphs. We present the ablation studies on both the extractive and abstractive results in Figure 3, where "w/o semantic", "w/o topical" and "w/o coreference" denote the GMQS-joint models without the semantic relevance, topical coherence, and coreference linking relation when constructing the multi-relational graph, respectively. Besides, "all relation" refers to the performance of the model with all three relations. We can see that all of the semantic relations contribute to the final performance and discarding any of them leads to a decrease of performance. This result illustrates the importance of explicitly modeling the complex relations among the question and the document sentences for non-factoid QA. The topical coherence and coreference linking relations attach more importance to the final performance, while the semantic relevance relation affects the performance the least as the intra-/inter-sentence encoder may capture such information to a certain extent. In addition, we observe that the coreference linking relation is more effective in the WikiHow dataset. Since there are more pronouns in the WikiHow dataset, the multi-hop reasoning relies more on coreference resolution to link the relation among different sentences in the document. However, as for the PubMedQA dataset with professional medical documents, the mentioned entities are clearly stated without using pronouns in the source document, so that the coreference relation might be less effective. To better verify this observation, we statistically present the performance in terms of the number of pronouns in the source document in Fig. VI-C. It can be observed that the it is harder to achieve a high performance in cases with a larger number of pronouns, while the coreference relation is more effectively in these cases.

E. Analysis of Multi-hop Reasoning

1) Impact of the Number of Hops: In the proposed graph-enhanced multi-hop inference module, the number of RGAT layers corresponds to the number of reasoning hops. To investigate the impact of the number of hops on the model performance, the experimental results on varying the number of RGAT layers are shown in Figure 4. We can see that, as expected, the performance of the model begins with growth when increasing the number of hops for reasoning. In particular, even using one hop of inference can make a noticeable contribution to both the extractive and abstractive

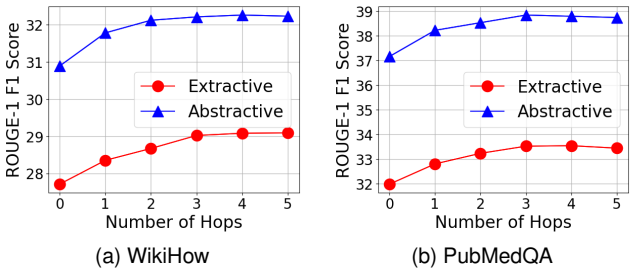


Fig. 4. Impact of different number of hops.

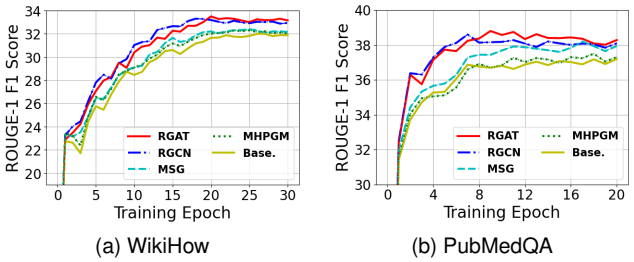


Fig. 5. Training efficiency analysis.

performance, which indicates the importance of considering the complex interrelationships among the document sentences. However, the performance merely changes on WikiHow and even slightly decreases on PubMedQA, when we further increase the number of RGAT layers. The possible reason is that the number of parameters also increases when we adapt more reasoning hops, leading to the over-fitting issue. This is a common phenomenon in GNN applications, which has also been observed from other NLP tasks that require the capability of multi-hop reasoning, such as knowledge graph completion [34], multi-choice QA [**?**], etc.

2) Training Efficiency Analysis: To better understand the training process of the graph-enhanced multi-hop inference module, we illustrate the testing performance curves of GMQS with different multi-hop inference modules as well as without the multi-hop inference module. Figure 5 shows the learning curves of the ROUGE-1 F1 score during the training process on both WikiHow and PubMedQA datasets, respectively.

As for the PubMedQA dataset, the proposed GMQS and the RGCN-variant quickly converge to the optimal value after about 12 epochs, and the RGCN-variant is even slightly faster than the proposed GMQS. However, the other multi-hop variants, e.g., MSG, need to take almost 20 epochs to converge to the optimal value, and the non-multi-hop model is the slowest one. This is because the multi-relational graph structure can be served as some prior knowledge for assisting in the multi-hop reasoning, which accelerates the learning process. Besides, there are more parameters to be trained for the RGAT than the RGCN, which may cause a slight speed reduction, but it also provides better performance. In addition, this result also shows that the multi-hop inference module enables the model to capture the important and salient information in the document more quickly. As for the WikiHow dataset, we can also make a similar conclusion.

3) Case Study: We present a case study in Figure 6 with generated answers from the proposed method and some baseline methods, including MSG, MGSum, and QPGN, to intuitively compare these methods. As for marks for the question and document, *Italic*, underlined, and wavy-underlined sentences represent those highly weighted sentences in 1st-hop, 2nd-hop, and 3rd-hop inference by GMQS, respectively. While the **highlighted** sentences represent those sentences that are supposed to be involved in the final answer. As for the reference answer and the answers produced by different methods, *Italic*, underlined, and wavy-underlined sentences represent those sentences that are related to the sentences in *1st-hop*, 2nd-hop, and 3rd-hop from the document, respectively. While the **highlighted** sentences represent those sentences that precisely answer the given question, i.e., similar to the reference answer. In other words, those regular sentences are incorrect or irrelevant to the given question.

We observe that it probably requires more than 3 hops of reasoning to infer the answers in this case, since there are multiple steps to answer the given question. We can still evaluate how the proposed GMQS handles such a case from the perspective of 3-hop inference. Compared to the reference answer, GMQS can capture most of the useful information to generate a good summary for answering the question, using either extractive or abstractive methods. Due to the length limitation in the experimental setup, the extractive result (GMQS-ext-joint) only fetches a certain number of sentences with the most important information from different hops of inference. The abstractive result (GMQS-abs-joint) successfully incorporates the key information to form the final answer. However, MGSum and QPGN introduce some unnecessary or incorrect information into the summarized answers.

Compared with MSG (3-Hop), which is also capable of multi-hop inference, the answer generated by GMQS covers more required information from the source document. This result indicates that only modeling the semantic relevance is inadequate for producing a comprehensive answer to the given non-factoid question. The proposed graph-enhanced multi-hop inference method enables to explicitly explain the inferred reasoning paths for producing the final answer. In this case, we visualized the multi-relational graph concerning the **highlighted** sentences during the multi-hop inference process in Figure 7. It can be observed that *Sentence 13* is not computed to be semantic relevant to other highlighted sentences. However, it is computed to be topically coherent to the question as well as linked to *Sentence 9* by the coreference of "milk replacer".

F. Error Analysis

We conduct error analysis on the generated answers selected for human evaluation (Section VI-C). Table VI summarizes the four most frequent error types and their error rates. In general, missing information and redundant information are the most common errors in the generated answers by both extractive and abstractive GMQS methods. Compared with GMQS-ext, GMQS-abs can greatly avoid errors regarding incoherence in the generated answers. However, due to the hallucination issue, which is the typical flaw of generation methods, GMQS-abs suffers more from the incorrect information.

Question: How to tube feed a puppy?
Document: 1. You will need a 12 cc syringe, a soft rubber feeding tube, and a 16-inch urethral catheter with a diameter of 5 French (for small dogs) and 8 French (for large dogs). 2. These are the items you will use to create your feeding tube device. 3. You will also need puppy milk replacer that contains goats milk, like ESBILAC®. 4. You can also buy an already assembled feeding tube from your local veterinary office or pet store. 5. ***You will need to determine the puppy's weight so that you know how much milk replacer to give him.*** 6. ***Place him on a scale to determine his weight.*** 7. ***For every ounce of the puppy's weight, give him 1 cc or ml of the milk replacer.*** 8. Add one extra cc to be careful. 9. **You will want to heat the milk replacer up so that it is easier on the puppy's stomach.** 10. **Place the milk into the microwave for three to five seconds so that it reaches a lukewarm temperature.** 11. Draw the milk up until you have the measured amount of milk, plus one extra cc. 12. The extra cc will be used to ensure that puppy doesn't get any air bubbles, which could cause bloating or gas pain. 13. **Once the syringe has drawn up all of the milk replacer, press down gently until a tiny drop comes out of the syringe.** 14. Doing this will ensure that the syringe is working properly. 15. **You will need to attach the end of the rubber feeding tube to the end of the syringe.** 16. To do this, place the tip of the rubber tube up against the side of the puppy's bottom, or last, rib, and run the tube from there to the tip of the pup's nose. 17. **Pinch the tube where it touches the puppy's nose and make a mark there with a permanent marker.**
Reference Answer: Gather your supplies. *Weigh the puppy.* Measure out the correct amount of milk into a microwaveable bowl. Use the syringe to suck up the milk replacer. Attach the feeding tube to the syringe. Measure out the length of the tube you will insert into the puppy's mouth.
GMQS-ext-joint: ***For every ounce of the puppy's weight, give him 1 cc or ml of the milk replacer.*** **You will want to heat the milk replacer up so that it is easier on the puppy's stomach. Once the syringe has drawn up all of the milk replacer, press down gently until a tiny drop comes out of the syringe. You will need to attach the end of the rubber feeding tube to the end of the syringe.**
GMQS-abs-joint: Gather your supplies. ***Measure the puppy's weight.*** **Place the milk in the microwave. Fill the syringe with milk replacer. Attach the syringe to the rubber tube. Insert the syringe into the puppy's mouth.**
MSG (3-Hop): Gather your materials. ***Measure your puppy's weight.*** **Heat the milk replacer. Attach the rubber feeding tube to the end of the syringe.** Insert the end of the milk replacer into the milk replacer. **Insert the syringe into the puppy's mouth.**
MGSum-abs: Gather your supplies. ***Measure the puppy's weight.*** Add the milk replacer to the puppy's weight. Place the syringe in the microwave. Remove the syringe from the syringe.
QPGN: Gather your supplies. ***Measure the puppy's weight.*** Place the milk replacer on the puppy's stomach. Place the milk replacer on the puppy's stomach. Press the milk replacer into the milk replacer.

Fig. 6. Case study from WikiHow.

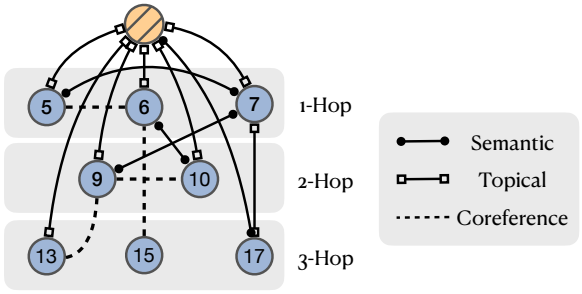


Fig. 7. Visualization of the multi-relational graph.

TABLE VI
ERROR ANALYSIS.

Error Type	GMQS-ext-joint	GMQS-abs-joint
Missing Info.	74%	64%
Redundant Info.	86%	62%
Incorrect Info.	12%	52%
Incoherence	44%	16%

VII. CONCLUSIONS AND FUTURE WORK

In this work, we study the non-factoid QA problem by proposing a novel query-focused summarization method, namely Graph-enhanced Multi-hop Query-focused Summarizer (GMQS). Specifically, we investigate graph-based reasoning techniques to perform multi-hop reasoning for collecting key information from documents to answer the given question. Three types of graphs with different semantic relationships are constructed, namely semantic relevance, topic coherence, and coreference linking, to explicitly capture the relationship between the question and each document sentence as well as among the document sentences. Then, the Relation Graph Attention Network (RGAT) is developed to aggregate the multi-relational information accordingly. In addition, the proposed method can be applied to both extractive and abstractive applications. Extensive experimental results show that the proposed method outperforms the existing baseline on non-factoid QA and has promising multi-hop reasoning capabilities.

It is noteworthy that the performance of the proposed framework depends on the construction of the semantic graphs to a great extent. In the future, we would like to explore other more informative graph representations such as knowledge graph, AMR graph, and leverage them to further improve the performance. By doing so, the finer-grained relations, such as word-level or entity-level interaction, can be investigated for improving the multi-hop inference. In addition, it is also worth exploring the deeper connections between multi-hop reasoning and the graph structure and studying more sophisticated graph neural network structures for the representational learning of multi-relational graphs.

REFERENCES

[1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100, 000+ questions for machine comprehension of text," in *EMNLP*, 2016, pp. 2383–2392.
[2] H. Song, Z. Ren, S. Liang, P. Li, J. Ma, and M. de Rijke, "Summarizing answers in non-factoid community question-answering," in *WSDM*, 2017, pp. 405–414.
[3] Y. Deng, W. Lam, Y. Xie, D. Chen, Y. Li, M. Yang, and Y. Shen, "Joint learning of answer selection and answer summary generation in community question answering," in *AAAI*, 2020, pp. 7651–7658.
[4] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, "ELI5: long form question answering," in *ACL*, 2019, pp. 3558–3567.
[5] M. Nakatsuji and S. Okui, "Conclusion-supplement answer generation for non-factoid questions," in *AAAI*, 2020, pp. 8520–8527.

[6] Q. Jin, B. Dhingra, Z. Liu, W. W. Cohen, and X. Lu, "Pubmedqa: A dataset for biomedical research question answering," in *EMNLP-IJCNLP*, 2019, pp. 2567–2577.

[7] E. Yulianti, R. Chen, F. Scholer, W. B. Croft, and M. Sanderson, "Document summarization for answering non-factoid queries," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 1, pp. 15–28, 2018.

[8] M. Keikha, J. H. Park, and W. B. Croft, "Evaluating answer passages using summarization measures," in *SIGIR*, 2014, pp. 963–966.

[9] Y. Yang, W. Yih, and C. Meek, "Wikiqa: A challenge dataset for open-domain question answering," in *EMNLP*, 2015, pp. 2013–2018.

[10] P. Nakov, L. Màrquez, W. Magdy, A. Moschitti, J. R. Glass, and B. Randeree, "Semeval-2015 task 3: Answer selection in community question answering," in *SemEval@NAACL-HLT*, 2015, pp. 269–281.

[11] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *SIGIR*, 2015, pp. 373–382.

[12] M. Tan, C. N. dos Santos, B. Xiang, and B. Zhou, "Improved representation learning for question answer matching," in *ACL*, 2016.

[13] S. Wang and J. Jiang, "A compare-aggregate model for matching text sequences," in *ICLR*, 2017.

[14] A. Rücklé, N. S. Moosavi, and I. Gurevych, "COALA: A neural coverage-based approach for long answer selection with small data," in *AAAI*, 2019, pp. 6932–6939.

[15] L. Wang, H. Raghavan, C. Cardie, and V. Castelli, "Query-focused opinion summarization for user-generated content," in *COLING*, 2014, pp. 1660–1669.

[16] G. Feigenblat, H. Roitman, O. Boni, and D. Konopnicki, "Unsupervised query-focused multi-document summarization using the cross entropy method," in *SIGIR*, 2017, pp. 961–964.

[17] J. Hasselqvist, N. Helmertz, and M. Kågebäck, "Query-based abstractive summarization using neural networks," *CoRR*, vol. abs/1712.06100, 2017.

[18] P. Nema, M. M. Khapra, A. Laha, and B. Ravindran, "Diversity driven attention model for query-based abstractive summarization," in *ACL*, 2017, pp. 1063–1072.

[19] Z. Cao, W. Li, S. Li, F. Wei, and Y. Li, "Attsum: Joint learning of focusing and summarization with neural attention," in *COLING*, 2016, pp. 547–556.

[20] V. Yadav, S. Bethard, and M. Surdeanu, "Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering," in *EMNLP-IJCNLP*, 2019, pp. 2578–2589.

[21] ——, "Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering," in *ACL*, 2020.

[22] Y. Deng, W. Zhang, and W. Lam, "Multi-hop inference for question-driven summarization," in *EMNLP*, 2020, pp. 6734–6744.

[23] Y. Li and S. Li, "Query-focused multi-document summarization: Combining a topic model with graph-based semi-supervised learning," in *COLING*, 2014, pp. 1197–1207.

[24] Y. Gao, Y. Xu, H. Huang, Q. Liu, L. Wei, and L. Liu, "Jointly learning topics in sentence embedding for document summarization," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 688–699, 2020.

[25] W. Li, X. Xiao, J. Liu, H. Wu, H. Wang, and J. Du, "Leveraging graph to improve abstractive multi-document summarization," in *ACL*, 2020, pp. 6232–6243.

[26] L. Bauer, Y. Wang, and M. Bansal, "Commonsense for generative multi-hop question answering tasks," in *EMNLP*, 2018, pp. 4220–4230.

[27] Y. Fang, S. Sun, Z. Gan, R. Pillai, S. Wang, and J. Liu, "Hierarchical graph network for multi-hop question answering," in *EMNLP*, 2020, pp. 8823–8838.

[28] W. Xu, Y. Deng, H. Zhang, D. Cai, and W. Lam, "Exploiting reasoning chains for multi-hop science question answering," in *Findings of ACL: EMNLP*, M. Moens, X. Huang, L. Specia, and S. W. Yih, Eds., 2021, pp. 1143–1156.

[29] Q. Lang, X. Liu, and W. Jia, "Afs graph: Multidimensional axiomatic fuzzy set knowledge graph for open-domain question answering," *IEEE Trans. Neural Networks Learn. Syst.*, 2022.

[30] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021.

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

[32] S. Jiang, Q. Chen, X. Liu, B. Hu, and L. Zhang, "Multi-hop graph convolutional network with high-order chebyshev approximation for text reasoning," in *ACL/IJCNLP*, 2021, pp. 6563–6573.

[33] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[34] G. Wang, R. Ying, J. Huang, and J. Leskovec, "Multi-hop attention graph neural networks," in *IJCAI*, 2021, pp. 3089–3096.

[35] J. Ma, J. Liu, Y. Wang, J. Li, and T. Liu, "Relation-aware fine-grained reasoning network for textbook question answering," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 34, no. 1, pp. 15–27, 2023.

[36] Q. Liu, X. Geng, H. Huang, T. Qin, J. Lu, and D. Jiang, "Mgrc: An end-to-end multigranularity reading comprehension model for question answering," *IEEE Trans. Neural Networks Learn. Syst.*, 2021.

[37] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "MS MARCO: A human generated machine reading comprehension dataset," in *NeurIPS*, 2016.

[38] Q. Liu, X. Geng, Y. Wang, E. Cambria, and D. Jiang, "Disentangled retrieval and reasoning for implicit question answering," *IEEE Trans. Neural Networks Learn. Syst.*, 2022.

[39] W. Zhang, Y. Deng, and W. Lam, "Answer ranking for product-related questions via multiple semantic relations modeling," in *SIGIR*, 2020, pp. 569–578.

[40] Y. Deng, Y. Xie, Y. Li, M. Yang, W. Lam, and Y. Shen, "Contextualized knowledge-aware attentive neural network: Enhancing answer selection with knowledge," *ACM Trans. Inf. Syst.*, vol. 40, no. 1, pp. 2:1–2:33, 2022.

[41] Y. Deng, Y. Shen, M. Yang, Y. Li, N. Du, W. Fan, and K. Lei, "Knowledge as A bridge: Improving cross-domain answer selection with external knowledge," in *COLING*, 2018, pp. 3295–3305.

[42] Y. Shen, Y. Deng, M. Yang, Y. Li, N. Du, W. Fan, and K. Lei, "Knowledge-aware attentive neural network for ranking question answer pairs," in *SIGIR*, 2018, pp. 901–904.

[43] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," in *IJCAI*, 2017, pp. 4144–4150.

[44] Y. Deng, Y. Li, W. Zhang, B. Ding, and W. Lam, "Toward personalized answer generation in e-commerce via multi-perspective preference modeling," *ACM Trans. Inf. Syst.*, vol. 40, no. 4, pp. 87:1–87:28, 2022.

[45] R. Ishida, K. Torisawa, J. Oh, R. Iida, C. Kruengkrai, and J. Kloetzer, "Semi-distantly supervised neural model for generating compact answers to open-domain why questions," in *AAAI*, 2018, pp. 5803–5811.

[46] R. Iida, C. Kruengkrai, R. Ishida, K. Torisawa, J. Oh, and J. Kloetzer, "Exploiting background knowledge in compact answer generation for why-questions," in *AAAI*, 2019, pp. 142–151.

[47] J. J. Lin, N. Madnani, and B. J. Dorr, "Putting the user in the loop: Interactive maximal marginal relevance for query-focused summarization," in *HLT-NAACL*, 2010, pp. 305–308.

[48] C. Shen and T. Li, "Learning to rank for query-focused multi-document summarization," in *ICDM*, 2011, pp. 626–634.

[49] L. Wang, H. Raghavan, V. Castelli, R. Florian, and C. Cardie, "A sentence compression based framework to query-focused multi-document summarization," in *ACL*, 2013, pp. 1384–1394.

[50] T. Ishigaki, H. Huang, H. Takamura, H. Chen, and M. Okumura, "Neural query-biased abstractive summarization using copying mechanism," in *ECIR*, 2020, pp. 174–181.

[51] T. Baumel, R. Cohen, and M. Elhadad, "Topic concentration in query focused summarization datasets," in *AAAI*, 2016, pp. 2573–2579.

[52] M. T. R. Laskar, E. Hoque, and J. X. Huang, "WSL-DS: weakly supervised learning with distant supervision for query focused multi-document abstractive summarization," in *COLING*, 2020, pp. 5647–5654.

[53] Y. Xu and M. Lapata, "Coarse-to-fine query focused multi-document summarization," in *EMNLP*, 2020, pp. 3632–3645.

[54] M. Singh, A. Mishra, Y. Oualil, K. Berberich, and D. Klakow, "Long-span language models for query-focused unsupervised extractive text summarization," in *ECIR*, 2018, pp. 657–664.

[55] Y. Deng, W. Zhang, Y. Li, M. Yang, W. Lam, and Y. Shen, "Bridging hierarchical and sequential context modeling for question-driven extractive answer summarization," in *SIGIR*, 2020, pp. 1693–1696.

[56] N. Zhang, S. Deng, J. Li, X. Chen, W. Zhang, and H. Chen, "Summarizing chinese medical answer with graph convolution networks and question-focused dual attention," in *Findings of ACL: EMNLP*, 2020, pp. 15–24.

[57] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," in *EMNLP*, 2018, pp. 2369–2380.

[58] Y. Feldman and R. El-Yaniv, "Multi-hop paragraph retrieval for open-domain question answering," in *ACL*, 2019, pp. 2296–2309.

[59] K. Nishida, K. Nishida, M. Nagata, A. Otsuka, I. Saito, H. Asano, and J. Tomita, "Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction," in *ACL*, 2019, pp. 2335–2345.

[60] L. Qiu, Y. Xiao, Y. Qu, H. Zhou, L. Li, W. Zhang, and Y. Yu, "Dynamically fused graph network for multi-hop reasoning," in *ACL*, 2019, pp. 6140–6150.

[61] S. Moon, P. Shah, A. Kumar, and R. Subba, "Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs," in *ACL*, 2019, pp. 845–854.
[62] H. Ji, P. Ke, S. Huang, F. Wei, X. Zhu, and M. Huang, "Language generation with multi-hop reasoning on commonsense knowledge graph," in *EMNLP*, 2020, pp. 725–736.
[63] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," in *ACL*, 2016.
[64] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "Neural document summarization by jointly learning to score and select sentences," in *ACL*, 2018, pp. 654–663.
[65] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *ACL*, 2017, pp. 1073–1083.
[66] S. Gehrmann, Y. Deng, and A. M. Rush, "Bottom-up abstractive summarization," in *EMNLP*, 2018, pp. 4098–4109.
[67] M. Yang, C. Li, Y. Shen, Q. Wu, Z. Zhao, and X. Chen, "Hierarchical human-like deep neural networks for abstractive text summarization," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 6, pp. 2744–2757, 2021.
[68] W. T. Hsu, C. Lin, M. Lee, K. Min, J. Tang, and M. Sun, "A unified model for extractive and abstractive summarization using inconsistency loss," in *ACL*, 2018, pp. 132–141.
[69] H. Jin, T. Wang, and X. Wan, "Multi-granularity interaction network for extractive and abstractive multi-document summarization," in *ACL*, 2020, pp. 6244–6254.
[70] Y. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," in *ACL*, 2018, pp. 675–686.
[71] J. Pilault, R. Li, S. Subramanian, and C. Pal, "On extractive and abstractive neural document summarization with transformer language models," in *EMNLP*, 2020, pp. 9308–9319.
[72] D. Wang, P. Liu, Y. Zheng, X. Qiu, and X. Huang, "Heterogeneous graph neural networks for extractive document summarization," in *ACL*, 2020, pp. 6209–6219.
[73] H. Zhang, C. Wang, Z. Wang, Z. Duan, B. Chen, M. Zhou, R. Henao, and L. Carin, "Learning hierarchical document graphs from multilevel sentence relations," *IEEE Trans. Neural Networks Learn. Syst.*, 2021.
[74] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *EMNLP-IJCNLP*, 2019, pp. 3728–3738.
[75] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *ACL*, 2020, pp. 7871–7880.
[76] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.
[77] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
[78] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *ESWC*, 2018, pp. 593–607.
[79] M. Koupaee and W. Y. Wang, "Wikihow: A large scale text summarization dataset," *CoRR*, vol. abs/1810.09305, 2018.
[80] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.
[81] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

Yang Deng is now working toward the PhD degree in the Department of System Engineering and Engineering Management, The Chinese University of Hong Kong. He received the BS degree from Beijing University of Posts and Telecommunications and the MS degree from Peking University. His research interests include Natural Language Processing, Information Retrieval, and Deep Learning.

Wenxuan Zhang is currently a research scientist at Alibaba DAMO Academy. He received the PhD degree from The Chinese University of Hong Kong. His research interests include Natural Language Processing and Deep Learning. He has published several papers in top-tier conferences in these areas. He has also been serving on the program committee of several international conferences and journals, including ACL, EMNLP, AAAI, SIGKDD, WSDM etc.

Weiwen Xu is now working toward the PhD degree in the Department of System Engineering and Engineering Management, The Chinese University of Hong Kong. He received the BS degree from University of Electronic Science and Technology of China. His research interests include Natural Language Processing, Information Retrieval, and Deep Learning.

Ying Shen is now an Associate Professor in School of Intelligent Systems Engineering, Sun Yat-Sen University. She received her Ph.D. degree from the University of Paris Ouest Nanterre La Défense (France), specialized in Computer Science. She received her Erasmus Mundus Master degree in Natural Language Processing from the University of Franche-Comté (France) and University of Wolverhampton (England). Her research interests include Natural Language Processing and deep learning.

Wai Lam received a Ph.D. in Computer Science from the University of Waterloo. He obtained his BSc. and M.Phil. degrees from The Chinese University of Hong Kong. After completing his Ph.D. degree, he conducted research at Indiana University Purdue University Indianapolis (IUPUI) and the University of Iowa. He joined The Chinese University of Hong Kong, where he is currently a professor. His research interests include text mining, natural language processing, and intelligent information retrieval. He has published extensively in top-tier conferences and journals in these areas.