DSM: Question Generation over Knowledge Base via Modeling Diverse Subgraphs with Meta-learner

Shasha Guo[1,2], Jing Zhang[1,2]*, Yanling Wang[1,2], Qianyi Zhang[1,2], Cuiping Li[1,2], Hong Chen[1,2]
[1]School of Information, Renmin University of China, Beijing, China
[2]Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education
{guoshashaxing, zhang-jing, wangyanling, qianyizhang, licuiping,chong}@ruc.edu.cn

Abstract

Existing methods on knowledge base question generation (KBQG) learn a one-size-fits-all model by training together all subgraphs without distinguishing the diverse semantics of subgraphs. In this work, we show that making use of the past experience on semantically similar subgraphs can reduce the learning difficulty and promote the performance of KBQG models. To achieve this, we propose a novel approach to model diverse subgraphs with meta-learner (DSM). Specifically, we devise a graph contrastive learning-based retriever to identify semantically similar subgraphs, so that we can construct the semantics-aware learning tasks for the meta-learner to learn semantics-specific and semantics-agnostic knowledge on and across these tasks. Extensive experiments on two widely-adopted benchmarks for KBQG show that DSM derives new state-of-the-art performance and benefits the question answering tasks as a means of data augmentation. Codes and datasets are available online[1].

1 Introduction

In recent years, knowledge base question generation (KBQG) has attracted substantial research interests as it shows great promise to improve the quality of question answering (QA). Specifically, KBQG can augment training data for QA systems (Chen et al., 2020; Indurthi et al., 2017), and it can also motivate the machines to actively ask questions in human-machine conversations (Sun et al., 2018b; Zeng and Nakano, 2020).

Concretely, KBQG generates natural language questions according to a set of facts extracted from KB, where each fact is typically specified as a triplet (e, r, e') meaning entity e has relation r with entity e'. Previous efforts on KBQG can be categorized into template-based models (Seyler et al., 2017) and neural network-based (NN-based) models (Bi et al., 2020; Kumar et al., 2019). The former ones heavily depend on hand-crafted templates, resulting in low scalability as these templates are limited to narrow domains. Alternatively, NN-based models address this issue via inputting the set of triplets about a certain answer into a Seq2Seq architecture to automatically generate the question.

In fact, for generating a question, triplets about a certain answer can naturally form a subgraph as illustrated in Figure 1. We observe that subgraphs differ in their semantics, which is especially shown in the relations that express the triplets[2] as well as the structural patterns such as chain, star, and triangle[3]. Existing efforts do not distinguish the semantics of different subgraphs but learn a one-size-fits-all model by training together all subgraphs, which increases the learning difficulty. Inspired by humans who solve a problem by searching the relevant problems that they have encountered in the past and adjusting the solution of these problems to the new one (Lancaster and Kolodner, 1987; Ross, 1984), we avoid directly learning a model on the entire data but try to leverage the past experience from similar KBQG cases to supervise generation from the current subgraph.

To achieve the goal, we propose a KBQG approach which models Diverse Subgraph with Meta-learner (DSM). DSM retrieves semantically similar subgraphs which share similar relations and structures to construct semantics-aware learning tasks, so that the model can carefully learn potential question generation (QG) patterns over each kind of subgraphs. With multiple learning tasks, we employ a Model-Agnostic Meta-Learning (Finn et al., 2017)-like (MAML-like) meta-learner to capture semantics-specific and semantics-agnostic knowl-

*Corresponding author.

[1]https://github.com/RUCKBReasoning/DSM

[2]The relations instead of the concrete entities in a subgraph is the decisive factor of the meaning.

[3]More complex examples can be viewed as the combination of chain, star, and triangles.

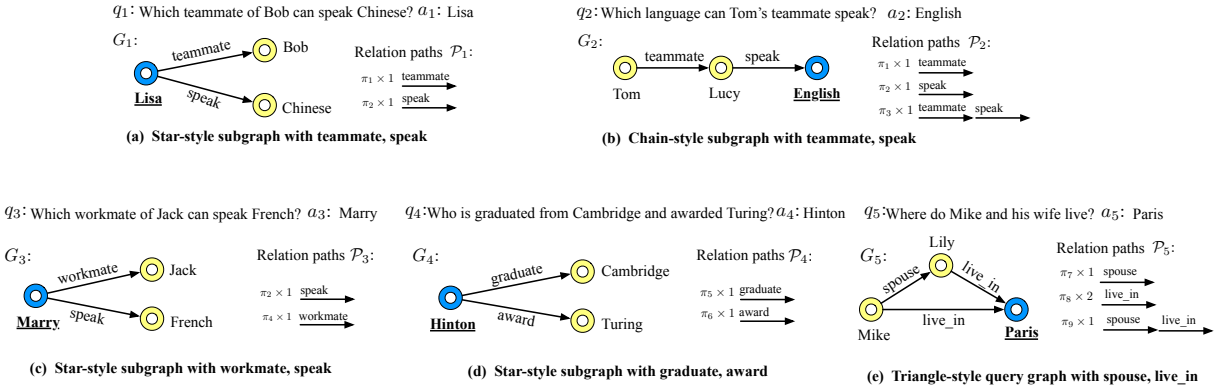


Figure 1: Illustration of diverse subgraphs in a knowledge graph, where blue nodes represent answer entities. Each subgraph is used to generate a question about the answer entity. Especially, question generation (QG) based on the subgraphs with different relations (e.g., teammate, speak, and workmate) and structures (e.g., chain, star, and triangle) could follow distinct potential rules, so we suggest addressing subgraphs with distinct semantics differently.

edge on and across different learning tasks.

To create the above learning tasks in DSM, retrieving similar subgraphs is crucial. Although classic graph matching algorithms (Li et al., 2019; Riba et al., 2018) can help do this, they only consider graph structural properties, regardless of the semantics of relations. Inspired by the great success of graph neural networks (GNNs) (Hamilton et al., 2017; Kipf and Welling, 2017; Velickovic et al., 2018), we turn to represent subgraphs in the embedding space by GNNs, as they can easily encode both relations and structures. By doing this, we can retrieve semantically similar subgraphs according to cosine similarity between their representations. Due to the lack of supervision, we perform graph contrastive learning (GCL) (Qiu et al., 2020; You et al., 2020a), which is one of the mainstream graph self-supervised learning methods. To enable GCL, we propose relation path-based similarity, a simple and effective metric, to retrieve similar subgraphs as positive samples of contrastive learning.

Contributions. (1) We design a KBQG approach that considers the diversity of subgraph semantics. Instead of training subgraphs of different semantics together, we construct semantics-specific learning tasks to reduce the learning difficulty. (2) We devise a GCL-based retriever to identify semantically similar subgraphs, so that we can construct semantics-aware learning tasks for the meta-learner to enable the meta-leaner learn semantics-specific and semantics-agnostic knowledge. (3) Our model shows the new state-of-the-art (SOTA) performance in BLEU and ROUGE, and benefits the QA tasks as a means of data augmentation. Human evaluation and case studies also show that our model can generate more relevant and fluent questions than other baselines.

2 Related Work

Knowledge Base Question Generation. Existing KBQG models can be divided into two categories — template-based models and neural network-based (NN-based) models. The former (Seyler et al., 2017) designs heuristic templates for question generation, which is simple but has low scalability. Driven by advances of deep neural networks (Shen et al., 2018; Vaswani et al., 2017), NN-based models (Bi et al., 2020; Elsahar et al., 2018; Indurthi et al., 2017; Liu et al., 2019) are applied to generate questions automatically. Generally, triplets in a subgraph are organized into a sequence to be the input of a Seq2Seq (Sutskever et al., 2014) neural network. Since the graph topology around each entity also contains useful semantics, recent studies (Chen et al., 2020; Kumar et al., 2019) utilize graph neural networks to encode the structural patterns. Nevertheless, previous studies overlook modeling the diversity of graph semantics, which could make the model easy to get over-fitting. On the other hand, since existing NN-based models are trained from scratch, their performance thereby heavily relies on the scale of training data. Pre-trained language models (PLMs) can help solve this problem as they have been trained on the large corpus to be empowered with rich semantic information. Currently, some attempts of question generation over unstructured textual data (Chan and Fan, 2019; Dong et al., 2019) have adopted PLMs. But rare studies discuss PLM-based question gen-

eration over structured KB. To this end, we target to design a PLM-based KBQG model which considers the diversity of subgraphs in KB.

Graph Self-supervised Learning. To construct semantics-aware learning tasks, we need to retrieve subgraphs with similar semantics. This work performs self-supervised learning (SSL) over subgraphs to learn their representations. By doing so, we can retrieve semantically similar subgraphs according to the similarity between their representations. Graph SSL schemes, which learn node-level or graph-level representations to retain the attributive and structural patterns of the graph data, have been widely studied in recent years. Generally, graph SSL (Liu et al., 2021; Wang et al., 2021) can be divided into four types, including generation-based (Kim and Oh, 2021; You et al., 2020b), auxiliary property-based (Peng et al., 2020a; Sun et al., 2020), contrastive-based (Hu et al., 2020a; Wang et al., 2022) and hybrid (Hu et al., 2020b; Peng et al., 2020b) methods. This paper adopts the graph contrastive learning method to learn subgraph representations via "graph-graph" contrast, which is more suitable to the target of subgraph retrieval.

3 Preliminary

A **knowledge base** (KB) organizes the factual information as a set of triplets, *i.e.*, $KB = \{(e, r, e')|e, e' \in E, r \in R\}$, where E and R denote the entity set and the relation set respectively. From a KB, we can extract any subgraph $G_i \subset KB$. For convenience, we employ A_i to denote the adjacent matrix of G_i and use n_i to represent the number of nodes in G_i.

3.1 KBQG

Given a set of (subgraph, answer, question) tuples as the training data denoted by $\mathcal{D} = \{(G_i, a_i, q_i)\}_{i=1}^{N}$ with N as the total number of data samples, the objective of KBQG is to learn a mapping function f with parameter θ, *i.e.*,

$$f_\theta : (G_i, a_i) \rightarrow \hat{q}_i, \tag{1}$$

where $\hat{q}_i$ denotes the predicted natural language question consisting of a sequence of word tokens, and q_i is the ground truth. The goal is to optimize the model parameter θ to maximize the conditional likelihood $P_\theta\left(q_i|G_i, a_i\right)$.

3.2 PLM-based KBQG Model

Inspired by the great success of pretrained language models (PLMs), we first use BART (Lewis et al., 2020), a pre-trained Seq2Seq model, as a baseline to instantiate f_θ. Specifically, we linearize each subgraph G_i into a triplet-based sequence, where each triplet is separated by the special token "</s>", then we input the sequence into BART to generate a question about the answer a_i[4].

As reported in Table 2, directly fine-tuning the BART has already outperformed the state-of-the-art baseline G2S+AE+RL (Chen et al., 2020). However, this straightforward solution ignores the diverse semantics of subgraphs. Instead of learning a one-size-fits-all model, we model diverse semantics and learn over semantically similar subgraphs, aiming to reduce the learning difficulty.

4 DSM

We introduce our KBQG approach which models Diverse Subgraph with Meta-learner (DSM).

4.1 Model Overview

Figure 2 illustrates the overview of our approach. Overall, DSM contains two key components, a subgraph retriever and a MAML-like meta-learner. The subgraph retriever retrieves top-k similar subgraphs to a query subgraph to construct a learning task. Based on multiple learning tasks, a MAML-like meta-learner summarizes semantics-specific and semantics-agnostic knowledge on and across these learning tasks.

Specifically, for a given query subgraph G_i of an answer entity a_i, the subgraph retriever retrieves top-k similar subgraphs to G_i, which compose the support set $\mathcal{S}_i = \{(G_j, a_j, q_j)\}_{j=1}^{k}$ for the data sample $\mathcal{D}_i = (G_i, a_i, q_i)$. To flexibly retrieve the top-k subgraphs, we represent each subgraph by a GNN encoder and leverage graph contrastive learning (GCL) to learn parameters of the GNN encoder. A key of GCL is constructing positive sample pairs. To enable GCL to capture the semantic similarity between subgraphs, we devise a relation path-based similarity metric to guide the positive sample pair construction. After retrieving the support set for each learning task, the MAML-like meta-learner optimizes the parameter θ of the mapping function

[4]We have empirically proved that the generated question is insensitive to the order of the triplets.

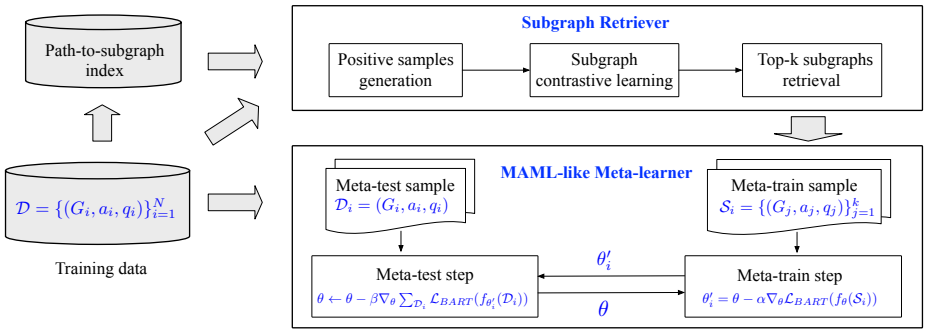


Figure 2: Illustration of the overview framework, which comprises a subgraph retriever and a MAML-like meta-learner. The subgraph retriever, implemented by a GCL-based method, retrieves top-k similar subgraphs to construct multiple learning tasks, such that the meta-learner can learn the semantics-specific knowledge on each task by the meta-train step and learn the semantics-agnostic knowledge across these tasks by the meta-test step.

f, which is instantiated by a pre-trained BART[5]. Precisely, meta-learner includes two optimization steps: meta-train step and meta-test step, where the former targets to learn semantics-specific knowledge, while the latter captures semantics-agnostic knowledge shared across different tasks.

Inference. At the inference phase, given a subgraph G_i of an answer entity a_i as the query, we first create the support set $\mathcal{S}_i$ for it. Afterwards, we fine-tune f_θ on $\mathcal{S}_i$ to obtain f_{θ_i}, then we generate the natural language question from G_i using the fine-tuned model f_{θ_i}.

4.2 Subgraph Retriever by GCL

A core step of DSM is to retrieve subgraphs that are similar to the query subgraph. Accordingly, it is necessary to discuss what are semantically similar subgraphs, so that we can design an effective graph retriever. Specifically, we claim that the semantics of a subgraph is mainly determined by its relations and structures.

- If two subgraphs **share relations**, we can generate semantically similar questions. Examples in Figure 1(a) and Figure 1(b), and examples in Figure 1(a) and Figure 1(c), support this claim. Although subgraphs in Figure 1(a) and Figure 1(b) have distinct structures, it is worth noting that the shared relations help produce semantically similar questions.
- **If two subgraphs share relations and have similar structures**, their generated questions can be more similar, especially in the sentence patterns. Figure 1(a) and Figure 1(c) demonstrate the assumption.

[5]The backbone of our model is BART, but other PLMs can also be used, such as T5.

More concretely, the structure is useful but not the decisive factor in the semantics of a subgraph. If relation sets of two subgraphs do not intersect, no matter how similar their structures are, the generated questions differ a lot from each other. Besides, we find that entity names can be easily copied from the input subgraph to the generated question, so it is unnecessary to consider entity names when measuring the semantic similarity.

Motivated by the great success of graph neural networks (GNNs) (Kipf and Welling, 2017; Velickovic et al., 2018), we leverage GNNs to learn a low-dimensional real-valued embedding for each subgraph, so that we can retrieve according to the cosine similarity between subgraph representations. Since GNNs are able to encode both the semantics of relations and structures, the cosine similarity between subgraph embeddings can represent the above-demanded subgraph similarity. Due to the lack of supervision, we propose to perform graph contrastive learning (GCL) (Velickovic et al., 2019; Wang et al., 2022), one of the mainstream graph self-supervised learning (SSL) methods. To enable GCL, we define relation path-based similarity, a simple and effective metric, for finding similar subgraphs as the positive sample pairs of GCL.

Next, we explain how to design a **subgraph encoder** for encoding both relations and structures, and how to conduct **positive sample generation** for contrastive learning.

4.2.1 Relation-enhanced Subgraph Encoder

We propose a relation-enhanced graph encoder for representing a subgraph, as relation information is a crucial factor in the semantics of the generated questions. For example, in Figure 1, although the entities in Figure 1(c) are totally different from

those in Figure 1(a), they can be simply replaced as the placeholders, which do not influence the semantics of the generated questions. On the contrary, if two subgraphs such as (<Lisa, born_in, **France**>) and (<Lisa, favorite, **France**>) share the same entities but have different relations, the generated questions "*Where was Lisa born in?*" and "*Which is the favorite city of Lisa?*" are far from similar with each other. In light of this, we initialize the feature of an entity node e_j with the relations connected to it. More clearly, we input the sequence of relation names into the pre-trained BERT (Devlin et al., 2019). Then we average the relation embeddings to represent the entity's initial feature $\mathbf{h}_j^{(0)}$,

$$\mathbf{h}_j^{(0)} = \frac{1}{|R_j|} \sum_{r \in R_j} \mathbf{r}, \tag{2}$$

where $\mathbf{r}$ denotes the embedding of the relation r, and R_j is the set of the relations connected to e_j. Given a subgraph G_i with entities' initial features $\mathbf{H}_i^{(0)} = \{\mathbf{h}_j^{(0)}\}_{j=1}^{n_i}$ and the adjacency matrix A_i as input, the GNN encoder g_ϕ with L layers outputs the entity embeddings $\mathbf{H}_i^{(L)}$. Then we average all the outputted nodes' representations and apply a sigmoid activation function on the pooled result to represent the graph-level representation $\mathbf{z}_i$.

Building on the relation-based initial node features, we adopt GIN (Xu et al., 2019), a SOTA GNN architecture, to instantiate the GNN encoder g_ϕ. In addition to the neighborhood homophily, GIN can encapsulate the structures of nodes, which is helpful for representing a subgraph.

Obviously, other heterogeneous GNN encoders such as RGCN (Schlichtkrull et al., 2018) are alternative encoders. However, most of them create a separate parameter for each relation, which ignores the relations' natural language semantics. We will demonstrate the superiority of our method through experiments (Cf. Table 3 for details).

Contrastive Loss Function. We adopt the normalized temperature-scaled contrastive loss as in (Sohn, 2016; You et al., 2020a). Formally, the NT-Xent for a mini-batch is formulated as:

$$\mathcal{L}_{GCL} = \sum_{i=1}^{n(m+1)} \sum_{j \in Pos(i)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_j^+ / \tau)}{\sum_{j'=1}^{n(m+1)} \mathbb{1}_{[j' \neq i]} \exp(\mathbf{z}_i \cdot \mathbf{z}_{j'} / \tau)}, \tag{3}$$

where n is the number of subgraphs in a mini-batch, m represents the number of positive samples for each subgraph, $Pos(i)$ represents the indices of positive samples of the i-th sample, and τ denotes a temperature parameter. The contrastive learning algorithm is illustrated in Algo. 1.

Algorithm 1: Contrastive Learning

Input: $\mathcal{G} = \{G_i\}_{i=1}^N$, m (positive sample size).
Output: ϕ of the GNN encoder.

Initialize the parameters ϕ for the GNN encoder;
for each epoch **do**
  Sample a mini-batch of n subgraphs $\mathcal{B} = \{G_i\}_{i=1}^n \subset \mathcal{G}$;
  for each subgraph $G_i \in \mathcal{B}$ **do**
    Generate m positive sample $\{G_j^+ | G_j^+ \in \mathcal{G}\}_{j=1}^m$ (Algo. 2);
  end for
  $\{\mathbf{z}_i\}_{i=1}^{n(m+1)} = g_\phi(\{G_i, \{G_j^+\}_{j=1}^m\}_{i=1}^n)$;
  $\phi \leftarrow \text{Adam}(\mathcal{L}_{GCL}(\{\mathbf{z}_i\}_{i=1}^{n(m+1)}))$;
end for

Algorithm 2: Positive Sample Generation

Input: Paths of graphs $\{\mathcal{P}_i\}_{i=1}^N$ and path-to-graph index $\{\mathcal{G}_p\}_{p=1}^M$.
Output: Top-m similar subgraphs about G_i.

for each path $p \in \mathcal{P}_i$ **do**
  for each graph $G_j \in \mathcal{G}_p$ **do**
    Add G_j into the sample candidate set $\mathcal{C}_i$;
  end for
end for
for each graph $G_j \in \mathcal{C}_i$ **do**
  Calculate $S_{RP}(G_i, G_j)$ by Eq. (4);
end for
Return top-m similar graphs according to $S_{RP}(G_i, G_j)$;

4.2.2 Positive Sample Generation

In this subsection, we explain how to construct positive sample pairs. The details are presented in Algo. 2. Without the ground truth, we define relation path-based similarity, a simple and effective metric for measuring the similarity between subgraphs, and propose an efficient retrieval method based on a path-to-graph index.

Intuitively, the relation path-based metric can be directly used by the subgraph retriever. However, this method fails to cover subgraphs whose relation paths are distinct but semantically similar such as (<Lisa, , **born_in**, France>) and (<Lisa, **place_of_birth**, France>).

Relation Path. Given a subgraph G_i, a relation path is denoted by $\pi = (r_1, \cdots, r_{|\pi|})$. Any relation in π is included in G_i and is traversed following the relation direction from the arrowhead to the arrowtail. The length $|\pi|$ of path π represents the maximal length of all possible paths in G_i. Figure 1 illustrates relation paths in a subgraph. For example, in Figure 1(e), we can enumerate four paths, including one $\pi_7 = \xrightarrow{spouse}$, two $\pi_8 = \xrightarrow{live\_in}$, and one $\pi_9 = \xrightarrow{spouse}\xrightarrow{live\_in}$. We use the classic DFS algorithm to obtain all paths of a subgraph.

Relation Path Similarity. Given two subgraphs G_i and G_j, we respectively enumerate relation paths in them to construct sets $\mathcal{P}_i$ and $\mathcal{P}_j$. Then relation path-based similarity S_{RP} between G_i and G_j is defined as:

$$S_{RP}(G_i, G_j) = \frac{|\mathcal{P}_i \cap \mathcal{P}_j|}{|\mathcal{P}_i \cup \mathcal{P}_j|}, \tag{4}$$

Take Figure 1(a) as an example, given the subgraph G_1, $S_{RP}(G_1, G_2) = \frac{2}{3}$, $S_{RP}(G_1, G_3) = \frac{1}{3}$, and $S_{RP}(G_1, G_4) = S_{RP}(G_1, G_5) = 0$. Such measurement results meet our intuitive expectation that relations play the most important role followed by the structural properties.

Path-to-Graph Index. To improve the efficiency of calculating relation path-based similarity, we build a path-to-graph index with the path identifier p as the key and the set of subgraphs including the path π_p, *i.e.*, $\mathcal{G}_p$, as the value. The whole index is denoted by $\{\mathcal{G}_p\}_{p=1}^{M}$ with M as the number of all the distinct paths in the training data. Using the index, we can retrieve all subgraphs that contain a specific path with a complexity of $O(1)$. Then Eq. (4) can be calculated with a complexity of $O(\bar{N}\bar{T})$, where $\bar{N}$ is the average number of subgraphs with a path, and $\bar{T}$ is the average number of paths in a subgraph.

4.3 MAML-like Meta-learner

The subgraph retriever retrieves top-k similar subgraphs $\{G_j\}_{j=1}^{k}$ for a given G_i in training data, which are used to construct the support set $\mathcal{S}_i = \{(G_j, a_j, q_j)\}_{j=1}^{k}$ for a learning task about the query sample $\mathcal{D}_i = (G_i, a_i, q_i)$.

For the meta-learner, the learning process consists of a meta-train step and a meta-test step. For each learning task corresponding to a sample $\mathcal{D}_i$, the meta-train step learns a task-specific learner θ_i' based on the support set $\mathcal{S}_i$:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{BART}(f_\theta(\mathcal{S}_i)), \tag{5}$$

Algorithm 3: The DSM algorithm

Input: $\mathcal{D} = \{(G_i, a_i, q_i)\}_{i=1}^{N}$, step size α and β, k (support set size).
Output: θ of BART-base.

```
Perform GCL on D (Algo. 1) to learn φ of the GNN encoder;
Encode all the subgraphs by GIN to get the embeddings {z_i}_{i=1}^N;
for each G_i ∈ D do
    Retrieve {G_j}_{j=1}^k = Top-k(cosine(z_i, {z_j}_{j=1}^N));
    Create the support set S_i = {(G_j, a_j, q_j)}_{j=1}^k;
end for
Fine-tune θ of BART-base on D;
for each epoch do
    Sample a mini-batch of n samples B = {(G_i, a_i, q_i)}_{i=1}^n ⊂ D;
    for each (G_i, a_i, q_i) ∈ B do
        Update θ_i' based on S_i via Eq. (5);
    end for
    Update θ based on D via Eq. (6);
end for
```

where $\mathcal{L}_{BART}(f_\theta(\mathcal{S}_i))$ is the learner's loss function, and α is the update learning rate.

To connect multiple learning tasks, the meta-test step learns the task-agnostic learner θ by the loss computed using task-specific learner θ_i':

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{D}_i \in \mathcal{D}} \mathcal{L}_{BART}(f_{\theta_i'}(\mathcal{D}_i)), \tag{6}$$

where $\sum_{\mathcal{D}_i} \mathcal{L}_{BART}(f_{\theta_i'}(\mathcal{D}_i))$ is the meta-objective, and β is the meta learning rate. We summarize the whole procedure of the proposed DSM in Algo. 3.

5 Experimental Evaluation

We conduct extensive experiments to mainly answer the four questions: (1) Does DSM take effect in improving the KBQG performance? (2) Can the GCL-based subgraph retriever result in more effective support set for the meta-learner? (3) How do the positive sample size for contrastive learning and the support set size for the meta-learner affect DSM? (4) Can DSM benefit the QA tasks as a means of data augmentation?

5.1 Experimental Protocol

Datasets. We adopt two benchmarks, WebQuestions (WQ) and PathQuestions (PQ) (Zhou et al.,

Table 1: Data statistics. #Instances denotes the number of instances. #Entities and #Relations are the total number of entities and relations included in the dataset. #Triples represents the min/max/avg number of triplets in each instance.

Dataset	#Instances	#Entities	#Relations	#Triples
WQ	22,989	25,703	672	2/99/5.8
PQ	9,731	7,250	378	2/3/2.7

2018), for evaluating the proposed DSM. To be specific, WQ combines instances from WebQuestionsSP (Yih et al., 2016) and ComplexWebQuestions (Talmor and Berant, 2018). Table 1 shows statistics of the two datasets.

Evaluation Metrics. We adopt BLEU-n ($n = 1$-4) and ROUGE-L as automatic evaluation metrics. BLEU-n (Papineni et al., 2002) and ROUGE-L (Lin and Och, 2004) compute the ratios of the common n-grams between the generated question and the ground truth question, where the former can be viewed as precision and the latter focuses on recall. For QA performance, we use Hits@1 to evaluate the accuracy of the top-1 predicted answer. Since some questions have multiple answers, we also evaluate the F1 score. We also hire three people to evaluate the relevance and fluency of the generated questions.

Baselines. We compare with five baselines. Among them, **MHQG+AE** (Kumar et al., 2019) adopts a transformer (Vaswani et al., 2017) to encode the input subgraph and decode the question. **G2S+AE** (Chen et al., 2020) employs a bidirectional gated GNN to encode the input directed subgraph and decodes the question by a LSTM model. **G2S+AE+RL** (Chen et al., 2020) is a variant of G2S+AE that adds an additional reinforcement loss to incorporate the reward from BLEU-4 and ROUGE-L metrics. We also compare with **BART-base** and **BART-large** (Lewis et al., 2020) for KBQG. The details are explained in Section 3.

5.2 Overall Evaluation

Table 2 shows the overall evaluation results of all comparison models. By the results, we summarize the following conclusions: **(1) PLMs can contribute to KBQG.** BART-base and BART-large show better performance than the existing three baselines, because the BART models are pretrained on the large corpus so that they are empowered with rich knowledge, while existing baselines are all trained from scratch. **(2) DSM significantly outperforms BART-base and BART-large, which reflects the effectiveness of modeling the diversity of subgraphs.** The baselines train subgraphs of different semantics together, which increases the learning difficulty. Alternately, we learn on and across semantics-specific tasks to capture semantics-specific and semantics-agnostic knowledge. **(3) DSM shows more promising performance on the more diverse dataset WQ.** We observe that WQ has more diverse subgraphs including chain, star, and triangle structures, while PQ only has the subgraphs of chain and star structures. DSM obtains only 5.03% BLEU-4 gain and 1.52% ROUGE-L gain over the best results of baselines on PQ but significantly derives 6.81% BLEU-4 gain and 7.74% ROUGE-L gain over the best results of baselines on WQ. This shows that DSM can better address the dataset with more diverse subgraphs.

5.3 Evaluation of GCL-based Retriever

We evaluate whether the proposed GCL-based retriever can result in a support set of high quality. We keep the meta-learner in DSM, and vary the retriever as 1-RP (RP is the abbreviation of Relation Path) retriever, 2-RP retriever, All-RP retriever, GED-based retriever, DGI-based retriever, and RGCN-based retriever. The former three follow the same relation path-based similarity for generating positive samples in contrastive learning. Specifically, they enumerate the relation paths on the subgraphs and calculate the relation path similarity in Eq. (4) to retrieve top-k similar subgraphs. The differences lie in that 1-RP retriever and 2-RP retriever restrict the path length to 1 and 2 respectively, while All-RP considers all the possible paths. GED retriever retrieves top-k similar subgraphs according to the classic graph edit distance (Bunke, 1983). DGI-based retriever replaces contrastive loss in Eq. (3) with DGI loss (Velickovic et al., 2019), a "local-global contrast". RGCN-based retriever replaces the relation-enhanced subgraph encoder with the RGCN encoder (Schlichtkrull et al., 2018), which considers the features of entities and relations simultaneously.

Table 3 presents the evaluation results of DSM with various retrievers, which show that: **(1) DSM with variously devised retrievers can outperform the vanilla BART models.** The results indicate the effectiveness of the meta-learner coupled with a retriever in the KBQG task. **(2) All-RP shows more promising performance than 1-RP**

Table 2: Overall evaluation on WQ and PQ (%).

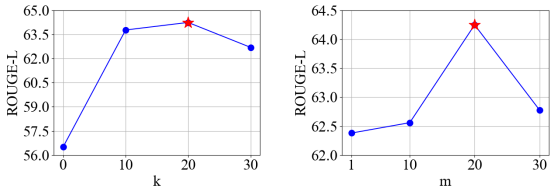
Model	WQ					PQ				
	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
MHQG+AE	42.35	29.32	18.43	9.63	35.72	45.02	35.86	28.73	17.86	63.45
G2S+AE	53.48	38.67	27.35	20.54	55.61	78.21	69.62	63.35	54.21	82.32
G2S+AE+RL	54.69	39.77	27.35	20.80	55.73	76.05	67.75	61.64	52.19	81.94
BART-base	56.39	41.05	29.59	21.46	56.51	79.59	70.63	64.30	55.73	84.54
BART-large	56.89	41.29	30.11	21.81	56.38	79.30	70.64	64.54	56.00	84.22
DSM(ours)	**62.94**	**48.20**	**37.50**	**28.62**	**64.25**	**82.44**	**74.20**	**68.60**	**61.03**	**86.06**
Performance Gain	6.05	6.91	7.39	6.81	7.74	2.85	3.56	4.06	5.03	1.52

Table 3: Evaluation of the GCL-based subgraph retriever and other retrievers (%).

Retriever	WQ					PQ				
	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
1-RP retriever	60.30	45.66	34.84	26.23	60.22	81.56	73.44	67.94	60.47	85.25
2-RP retriever	60.36	45.76	35.34	26.41	60.53	81.90	73.75	68.23	60.57	85.50
All-RP retriever	61.40	46.88	36.38	27.53	61.78	81.80	73.44	67.60	59.78	85.50
GED retriever	61.03	45.87	35.62	26.45	60.45	80.50	71.94	66.06	57.97	84.25
DGI-based retriever	61.28	46.57	36.23	26.58	62.13	82.10	73.80	68.25	60.66	85.80
RGCN-based retriever	60.87	45.72	34.57	25.03	59.24	80.40	71.56	65.44	56.84	84.40
GCL-based retriever	**62.94**	**48.20**	**37.50**	**28.62**	**64.25**	**82.44**	**74.20**	**68.60**	**61.03**	**86.06**

The bold format represents the best results over all the methods and the underline format represents the best results of baselines.

and 2-RP on the more diverse dataset WQ, while 2-RP outperforms the other two on PQ. Since PQ mostly contains the chain-style subgraphs with length 2, relation paths of length 2 could be more distinguished than other paths. On the contrary, WQ is more diverse, demanding relation paths with various lengths to express the potential structures. Thus All-RP performs better than the other two on WQ. **(3) The proposed GCL-based retriever outperforms the three RP retrievers and GED retriever**, because the three RP retrievers cannot find subgraphs without common relations, and the GED retriever only captures the structural knowledge but overlooks the relation semantics. **(4) GCL loss function outperforms the DGI loss function.** DGI aims to encode the global features of a whole subgraph into each node via the "local-global" contrast. On the contrary, GCL performs "global-global" contrast to directly compare two subgraphs, which is more suitable to the objective of the retriever. **(5) RGCN encoder is worse than the relation-enhanced GNN encoder.** In addition to the relation semantics, RGCN also encodes the entity information, which shows no obvious effect on determining the question semantics. Meanwhile, RGCN creates a separate parameter for each relation without considering the semantics presented by relation names, which also weakens its effect.



(a) Support set size (b) Positive sample size

Figure 3: Sensitivity study.

5.4 Sensitivity Study

We investigate how the support set size k in the meta-learner and the positive sample size m of GCL affect DSM. Figure 3(a) presents ROUGE-L of DSM over different support set sizes on WQ. We observe that the performance rises first then falls and reaches the top at 20. Similarly, Figure 3(b) presents ROUGE-L of DSM over different positive sample sizes for GCL on WQ. We observe the same optimal value of 20. The results indicate that more similar subgraphs might introduce additional noises. Similar trends are observed on PQ.

5.5 Positive Impacts on QA Tasks

We study whether the proposed DSM can benefit QA tasks as a means of data augmentation. We evaluate two classical KBQA models named GRAFT-Net (Sun et al., 2018a) and NSM (He et al., 2021)

Table 4: QA performance of GRAFT-Net and NSM.

Model	GRAFT-Net		NSM	
	F1	Hits@1	F1	Hits@1
Real	**0.622**	**0.681**	**0.666**	**0.727**
-o	0.493	0.575	0.524	0.594
+G2S	0.573	0.639	0.647	0.700
+BART	0.588	0.648	0.649	0.714
+DSM(ours)	0.604	0.664	0.663	0.721

Table 5: Human evaluation results on WQ.

Model	Fluency	Relevance
Ground truth	**4.86**	**4.42**
G2S	4.46	4.01
BART	4.65	4.14
DSM(ours)	4.72	4.35

on WebQSP (Yih et al., 2016), a widely-adopted KBQA dataset with 2,848 (question, answer) training instances. To evaluate the quality of the generated questions by DSM, we replace part of the (question, answer) pairs in WebQSP with the generated questions. Since the training data of WebQSP has 1,409 overlapped (question, answer) pairs with that of WQ, we can easily get their corresponding subgraphs from WQ. For easy evaluation, we replace the real questions of the 1,409 instances in WebQSP with the questions generated from the corresponding subgraphs by DSM and denote the dataset as +DSM. On this partially replaced WebQSP, we train GRAFT-Net and NSM and compare their performance with the same models trained on the original WebQSP (Real) and the version removing the overlapped instances (-o). We also train GRAFT-Net and NSM on the datasets partially replaced by the pseudo questions generated by G2S+AE+RL and the BART-large model. We denote them as +G2S and +BART respectively.

Table 4 presents the comparison results of GRAFT-Net and NSM trained on various datasets. The results show: **(1) The generated (question, answer) pairs can be viewed as a means of data augmentation for KBQA,** because both GRAFT-Net and NSM trained on the datasets partially replaced by various KBQG models (*i.e.*, +G2S, +BART, +DSM) can improve the QA performance of them trained on the partially removed dataset (*i.e.*, -o). **(2) DSM can generate much better questions than others,** because the KBQA models trained on the dataset generated by DSM perform best among all the other generated datasets. **(3) The generated dataset by DSM is quite close to the real data,** which is supported by the comparable results between "+DSM" and "Real".

5.6 Human Evaluation

We perform human evaluation to further verify the effectiveness of DSM. We randomly choose 100 samples $\mathcal{S}_{100} = \{(G_j, a_j, q_j)\}_{j=1}^{100}$ from the test set of WQ dataset. Different models generate different questions for the same (subgraph, answer) pair. We evaluate the generated questions by fluency and relevance, where the former assesses whether the generated questions are readable for humans, and the latter measures the relevance between the generated question and the input (subgraph, answer) pair. We score fluency and relevance on a five-point Likert scale, with 1-point being poor and 5-point being perfect. We invite 6 annotators to score each generated question and average their scores for the proposed DSM and two baselines G2S and BART.

Table 5 presents the human evaluation results on WQ, which shows that DSM can produce more fluent and relevant questions than the other baselines, and even competes with the ground truth questions.

6 Conclusion

This work pilots studies on KBQG. We propose DSM to exploit semantic knowledge of diverse subgraphs. Instead of training on different subgraphs together, we construct semantics-specific learning tasks to reduce the learning difficulty. Specifically, we devise a GCL-based retriever to flexibly construct semantics-specific learning tasks. Besides, a MAML-like meta-learner is employed to learn on the different learning tasks, such that we can learn the semantics-specific and the semantics-agnostic knowledge shared on and across tasks. Our model shows competitive performance across the widely used benchmarks. We believe that using the MAML-like meta-learner could be inspiring for learning on datasets with high diversity.

Acknowledgments

This work is supported by National Natural Science Foundation of China (62076245, 62072460, 62172424); Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (22XNH190); National Key Research & Develop Plan(2018YFB1004401); Beijing Natural Science Foundation (4212022).

Limitations

Our model suffers from weak generalization. For example, the ground truth question is "*What is the name of an attraction in Salt Lake City that has fewer than 1012563 visitors per year?*" but we generate "*What is the largest attraction in Salt Lake City Utah?*". The model fails to generate "fewer than 1,012,563 visitors per year" because it did not see the corresponding relation "$\xrightarrow{annual\_visitors}$" during training. In another example, the ground truth question is "*Who plays Jason Morgan on General Hospital as well as Cloud Strife?*" but we generate "*Who plays Jason Morgan on General Hospital?*". We observe that the corresponding relation path "$\xrightarrow{dubbing\_performances}\xrightarrow{actor}$" appears in the training data, but the model still fails to generate "as well as Cloud Strife", because the two-hop relation path is more difficult to be generalized.

References

Sheng Bi, Xiya Cheng, Yuan-Fang Li, Yongzhen Wang, and Guilin Qi. 2020. Knowledge-enriched, type-constrained and grammar-guided question generation over knowledge bases. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2776–2786.

Horst Bunke. 1983. What is the distance between graphs. *Bulletin of the EATCS*, 20:35–39.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent bert-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162.

Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2020. Toward subgraph guided knowledge graph question generation with graph neural networks. *arXiv preprint arXiv:2004.06015*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Proceedings of the 33rd International Conference on Neural Information Procedding Systems*, pages 13063–13075.

Hady Elsahar, Christophe Gravier, and Frederique Laforest. 2018. Zero-shot question generation from knowledge graphs for unseen predicates and entity types. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 218–228.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1126–1135.

William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2020a. Strategies for pre-training graph neural networks. In *Proceedings of the 8th International Conference on Learning Representations*.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. 2020b. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1857–1867.

Sathish Reddy Indurthi, Dinesh Raghu, Mitesh M Khapra, and Sachindra Joshi. 2017. Generating natural language question-answer pairs from a knowledge graph using a rnn based question generation model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 376–385.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Dongkwan Kim and Alice Oh. 2021. How to find your friendly neighborhood: Graph attention design with self-supervision. In *Proceedings of the 9th International Conference on Learning Representations*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*.

Vishwajeet Kumar, Yuncheng Hua, Ganesh Ramakrishnan, Guilin Qi, Lianli Gao, and Yuan-Fang Li. 2019. Difficulty-controllable multi-hop question generation from knowledge graphs. In *Proceddings of the 18th International Semantic Web Conference*, pages 382–398.

Juliana S Lancaster and Janet L Kolodner. 1987. Problem solving in a natural task as a function of experience. Technical report, Georgia Inst of Tech Atlanta School of Information and Computer Science.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. 2019. Graph matching networks for learning the similarity of graph structured objects. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3835–3845.

Chin-Yew Lin and FJ Och. 2004. Looking for a few good metrics: Rouge and its evaluation. In *Ntcir workshop*.

Cao Liu, Kang Liu, Shizhu He, Zaiqing Nie, and Jun Zhao. 2019. Generating questions for knowledge bases via incorporating diversified contexts and answer-aware loss. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 2431–2441.

Yixin Liu, Shirui Pan, Ming Jin, Chuan Zhou, Feng Xia, and Philip S Yu. 2021. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Zhen Peng, Yixiang Dong, Minnan Luo, Xiao-Ming Wu, and Qinghua Zheng. 2020a. Self-supervised graph representation learning via global context prediction. *arXiv preprint arXiv:2003.01604*.

Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020b. Graph representation learning via graphical mutual information maximization. In *Proceedings of the Web Conference 2020*, pages 259–270.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1150–1160.

Pau Riba, Andreas Fischer, Josep Lladós, and Alicia Fornés. 2018. Learning graph distances with message passing neural networks. In *Proceedings of the 24th International Conference on Pattern Recognition*, pages 2239–2244.

Brian H Ross. 1984. Remindings and their effects in learning a cognitive skill. *Cognitive psychology*, 16(3):371–416.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of the 15th International Semantic Web Conference*, pages 593–607.

Dominic Seyler, Mohamed Yahya, and Klaus Berberich. 2017. Knowledge questions from knowledge graphs. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 11–18.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Proceedings of the 32th AAAI Conference on Artificial Intelligenc*, pages 5446–5455.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1857–1865.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. 2018a. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242.

Ke Sun, Zhouchen Lin, and Zhanxing Zhu. 2020. Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 5892–5899.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018b. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 3104–3112.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 641–651.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*.

Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep graph infomax. In *Proceedings of the 7th International Conference on Learning Representations*.

Yanling Wang, Jing Zhang, Shasha Guo, Hongzhi Yin, Cuiping Li, and Hong Chen. 2021. Decoupling representation learning and classification for gnn-based anomaly detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1239–1248.

Yanling Wang, Jing Zhang, Haoyang Li, Yuxiao Dong, Hongzhi Yin, Cuiping Li, and Hong Chen. 2022. Clusterscl: Cluster-aware supervised contrastive learning on graphs. In *Proceedings of the ACM Web Conference 2022*, pages 1611–1621.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? In *Proceedings of the 7th International Conference on Learning Representations*.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 201–206.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020a. Graph contrastive learning with augmentations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 5812–5823.

Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. 2020b. When does self-supervision help graph convolutional networks? In *Proceedings of the 37th International Conference on Machine Learning*, pages 10871–10880.

Jie Zeng and Yukiko I Nakano. 2020. Exploiting a large-scale knowledge graph for question generation in food preference interview systems. In *Proceedings of the 25th International Conference on Intelligent User Interfaces Companion*, pages 53–54.

Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. 2018. An interpretable reasoning network for multi-relation question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2010–2022.

A Notations

As shown in Table 6, the notations in this paper are described in detail.

Table 6: Notations used in this paper.

Symbol	Description
$\mathcal{D}, N$	training data and its size
$\mathcal{D}_i$=(G_i, a_i, q_i)	a sample with subgraph G_i, answer a_i, and question q_i
A_i	the adjacent matrix of G_i
$\mathcal{S}_i$	the support set of $\mathcal{D}_i$
r, e, π	a relation, an entity, and a path
$\mathbf{r}, \mathbf{h}, \mathbf{z}_i$	the embeddings of r, e, and G_i
k, m	support set size and positive sample size
n, n_i	batch size and the number of entities in G_i
$\mathcal{P}_i, \mathcal{G}_p$	the set of paths in G_i and the set of subgraphs with π_p
f_θ, g_ϕ	the QG function and GNN encoder

B Experiment

B.1 Experiment Settings

For performing subgraph contrastive learning in Algo. 1, the key settings include: (1) We implement the graph encoder using the GIN framework (Xu et al., 2019) and employ the sum-style graph convolution, which sums the neighbor embeddings of a node during message passing to capture the structural properties of nodes. The layer number L is set to 1, as the average number of triplets in a subgraph is only 2.7 in PQ and 5.8 in WQ. (2) For initializing the embedding of a node, the embeddings of all its connected relations are averaged. Each relation is embedded by BERT (Devlin et al., 2019). (3) For implementing the contrastive loss, we set the number of the positive samples m to 20, which is the selected optimal value shown in Figure 3. (4) Following the setting of supervised contrastive learning (Khosla et al., 2020), we set the temperature parameter τ to be 0.07. In addition, we set the input feature dimension as 1024, the node representation dimension as 1024, the learning rate as 0.001, the batch size n as 16, the optimizer as Adam, the patience as 15, and the maximum epochs as 100 for early stopping.

In the proposed DSM, f is instantiated as BART-base. For fine-tuning BART-base in Line 7 of Algo. 3, we set the learning rate as 5e-5, batch size as 8, the patience as 15, and the maximum epochs as 50 for early stopping. BART-base has a 6-layers encoder and a 6-layers decoder. BART-large has a 12-layers encoder and a 12-layers decoder.

For fine-tuning BART-base by the meta-learner in Lines 8-14 of Algo. 3, we set the learning rate α for the meta-train step as 5e-5, the learning rate β for the meta-test step as 3e-5, the number of tasks in a batch as 8, the meta-train steps for each task (*i.e.*, the update steps for Line 11 in Algo. 3) as 1. We set the epochs for the meta-learner (*i.e.*, the loop times of Lines 9-13 in Algo. 3) to 5, as the BART-base model is fine-tuned before the meta-learner training process, resulting in quick convergence.

For inference, we fine-tune the BART-base model on the support set of a new sample by 5 epochs, and then infer the top beam as the generated question of the sample.

B.2 Case Study

Due to the space limitation, we only present 13 questions generated by DSM, G2S, and BART in Table 7 on WQ. We also show the corresponding query subgraph and the support set for the top-3 cases in Figure 4. The results show that: (1) The generated question derived by our model is much closer to the ground truth question than the baselines. (2) The retrieved top-2 subgraphs in the support set are quite similar to the query subgraph in the relation semantics and the structures, so that the experience of question generation on these similar subgraphs can benefit the question generation of the query subgraph.

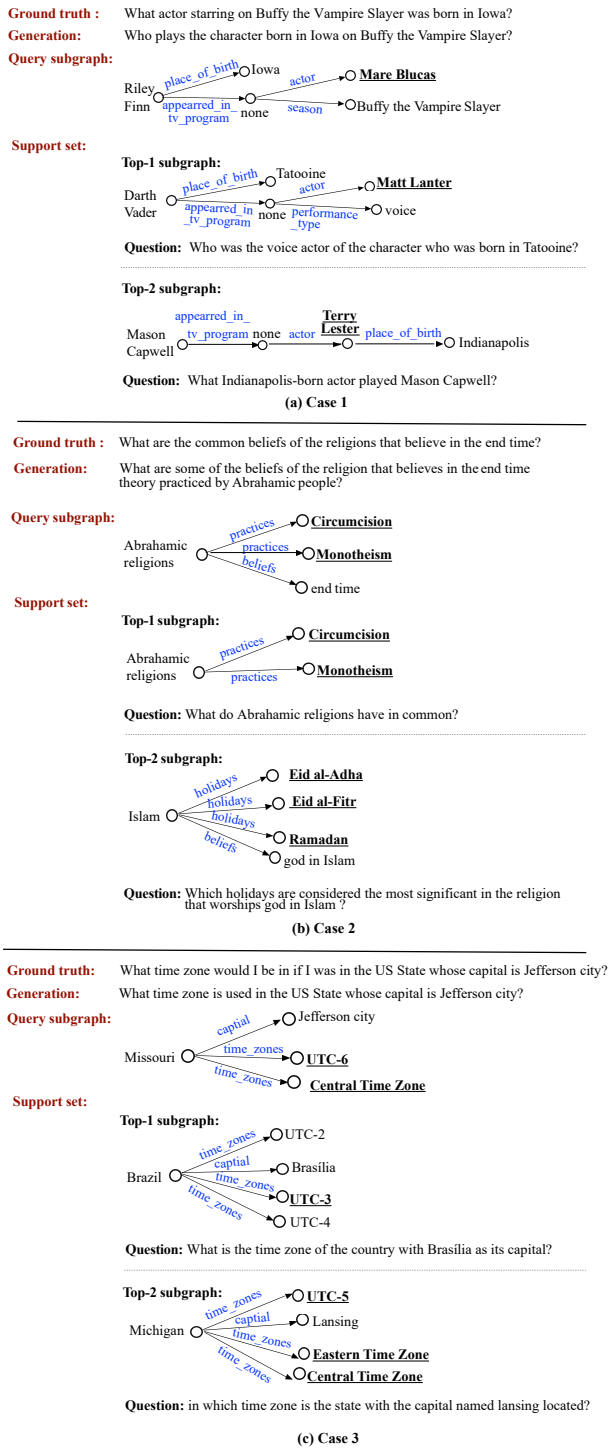


Figure 4: Case study of three generated questions. The support set is built by our proposed GCL-based retriever.

Table 7: Comparison of generated questions on WQ.

Ground Truth	G2S	BART	DSM
What actor starring on Buffy the Vampire Slayer was born in Iowa ?	Who plays Riley Finn in Buffy Vampire ?	Who plays the character born in Iowa in Buffy the Vampire Slayer ?	Who plays the character born in Iowa on Buffy the Vampire Slayer ?
What are the common beliefs of the religions that believe in the end times ?	What do the people who believes in end time ?	What did the person who believed in the end time practice ?	What are some of the beliefs of the religion that believes in the end time theory practiced by Abrahamic people ?
What time zone would I be in if I was in the US State whose capital is Jefferson city ?	What time zone is the state whose capital is Jefferson city ?	What is the time zone where Jefferson city is the capital ?	What time zone is used in the US State whose capital is Jefferson city ?
What country export to Sudan and has Giza necropolis ?	What country exports to Sudan and Sudan ?	What country that exports to Sudan is Giza neccropolis located in ?	What country exports to Sudan and is home to the Giza necropolis ?
What movie produced by Brad Lewis did Alyson Stoner starred in ?	What movie was produced by Brad Lewis ?	What movie produced by Brad Lewis was Alyson Stoner in ?	What movie produced by Brad Lewis did Alyson Stoner play in ?
Which english language shows did Henry Winkler produce ?	What show produced by Henry Winkler has Henry Winkler as an actor ?	What english speaking shows did Henry Winkler produce ?	What english speaking shows did Henry Winkler produce ?
Which type of monarchy does Japan have that is similar as the Kingdom of Prussia ?	What type of government is used in Japan and Japan ?	What type of government can be found in both Japan and the Kingdom of Prussia ?	What type of government is used in both Japan and the Kingdom of Prussia ?
What actress played Rose Loomis and has ties to John F. Kennedy ?	Which actor who played the character Rose Loomis ?	What actress who portrayed Rose Loomis was John F. Kennedy dating ?	Which actress played Rose Loomis and also dated John F. Kennedy ?
in the Tortall universe what language do native American Indians speak ?	What language is spoken in the <unk> and the fictional universe the Tortall universe ?	What language do native American Indians speak ?	What language, found in the fictional Tortall universe, do native American Indians speak ?
What person born in Batlesville was the first leader of the AFL ?	Who was born in <unk> and was born in <unk> ?	Who was born in Bartlesville and was a person of the AFL ?	Who was born in Bartlesville and was a member of the AFL first team ?
Who participated in the third joint debate at Jonesboro and influenced Walt Whitman's poetry ?	Who was the speaker of the speaker at <unk> ?	Which speaker participated in the third joint debate at Jonesboro ?	Which speaker featured in the third joint debate at Jonesboro influenced Walt Whitman ?
What country is home to Nova Roma and borders Bolivia ?	What country bordering Bolivia and Nova Roma ?	What country borders Bolivia and Nova Roma ?	What country borders Bolivia and is home to Nova Roma ?
What movie featuring Rihanna was released last ?	What is the earliest released film that Rihanna starred in ?	What is the latest released film that Rihanna starred in ?	What is the latest film that Rihanna has been in that was released last in 2012 ?

"<unk>" represents a word that does not appear in the vocabulary.