1-2024

A unified framework for contextual and factoid question generation

Chenhe DONG

Ying SHEN

Shiyang LIN

Zhenzhou LIN

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

Citation

DONG, Chenhe; SHEN, Ying; LIN, Shiyang; LIN, Zhenzhou; and DENG, Yang. A unified framework for contextual and factoid question generation. (2024). *IEEE Transactions on Knowledge and Data Engineering*. 36, (1), 21-34.
Available at: https://ink.library.smu.edu.sg/sis_research/9085

A Unified Framework for Contextual and Factoid Question Generation

Chenhe Dong, Ying Shen, Shiyang Lin, Zhenzhou Lin, and Yang Deng

***Abstract*— Question generation (QG) aims to automatically generate fluent and relevant questions, where the two most mainstream directions are generating questions from unstructured contextual texts (CQG), such as news articles, and generating questions from structured factoid texts (FQG), such as knowledge graphs or tables. Existing methods for these two tasks mainly face challenges of limited internal structural information as well as scarce background information, while these two tasks can benefit each other for alleviating these issues. For example, when meeting the entity mention "United Kingdom" in CQG, it can be inferred that it is a country in European continent based on the structural knowledge "(Europe, countries_within, United Kingdom)" in FQG. And when meeting the entity "Houston Rockets" in FQG, more background information, such as "an American professional basketball team based in Houston since 1971", can be found in the related passages of CQG. To this end, we propose a unified framework for the tasks of CQG and FQG, where: (i) two types of task-sharing modules are developed to learn shared contextual and structural knowledge, where the task format is unified with a pseudo passage reformulation strategy; (ii) for the CQG task, a task-specific knowledge module with a knowledge selection and aggregation mechanism is introduced, so as to incorporate more factoid knowledge from external knowledge graphs and alleviate the word ambiguity problem; and (iii) for the FQG task, a task-specific passage module with a multi-level passage fusion mechanism is designed to extract fine-grained word-level knowledge. Experimental results in both automatic and human evaluation show the effectiveness of our proposed method.**

***Index Terms*—Question generation, multi-task learning, knowledge acquisition.**

I. Introduction

QUESTION generation (QG) aims at automatically generating questions based on various forms of input data such as image [1], [2], text [3], [4], knowledge base (KB) and knowledge graph (KG) [5], [6]. QG is also categorized into different types [7] such as follow-up, clarifying, and information seeking. In recent years, QG has raised a lot of attention and has shown its advantages in many scenarios such as intelligent tutoring systems in education [8] and dialogue systems [9]. In this paper, we target the two most mainstream information seeking QG directions in natural language processing (NLP): generating questions from unstructured contextual data (CQG), such as news articles; and generating questions from structured factoid data (FQG), such as knowledge graphs or tables.

The end-to-end sequence-to-sequence (Seq2Seq) framework has become the de-facto method to tackle QG tasks [10], [11], [12]. *For the task of CQG*, many methods have been proposed to fully explore the internal structural information (e.g., answer-relevant relations, dependency parsing relations, etc.), which can guide the model focus on more prominent phrases to generate more consistent and to-the-point questions. For example, Li et al. [13] jointly model the unstructured texts and the structured answer-related relations contained in the input texts (e.g., the structured relation "the daily mean temperature in January; is; 0.3 °C" corresponding to the answer "0.3 °C"), which helps the model capture distant dependencies to the answer and ignore the extraneous information, leading to more to-the-point generated questions. Chen et al. [4] utilize the Graph Neural Network (GNN) to embed the syntax-based and semantics-aware relations inside the texts, which helps the model discover the syntactic and semantic relationships between any pair of words and generate more fluent and consistent questions. However, the structural knowledge hidden in the context is relatively limited and how to effectively incorporate external structural information (e.g., from other related tasks, external KBs, etc.) has not been well studied. *For the task of FQG*, due to an extreme lack of background and contextual information, many works attempt to introduce external related off-the-shelf contexts (e.g., distant supervised relation contexts, entity domains and descriptions) or related KG subgraphs in a multi-hop manner to generate more fluent and diversified questions. For example, Liu et al. [5] propose a context-augmented fact encoder and a multi-level copy mechanism to incorporate diversified off-the-shelf KB contexts, Chen et al. [6] utilize GNN to encode the KG subgraphs with multiple fact triples. However, the introduced KB contexts or KG subgraphs are usually composed of short words or phrases, which still suffer from limited knowledge and cannot remarkably address the above challenges. For example, the relation "person/place_of_birth" has a distant supervised

Manuscript received 2 May 2022; revised 15 May 2023; accepted 16 May 2023. Date of publication 26 May 2023; date of current version 27 November 2023. This work was supported in part by the 173 program under Grant 2021-JCJQ-JJ-0029, in part by the Shenzhen General Research Project under Grant JCYJ20190808182805919, and in part by the National Natural Science Foundation of China under Grant 61602013. Recommended for acceptance by J. Lee. *(Chenhe Dong and Ying Shen are equal contribution.) (Corresponding author: Yang Deng.)*

Chenhe Dong, Ying Shen, Shiyang Lin, and Zhenzhou Lin are with the School of Intelligent Systems Engineering, Sun Yat-Sen University, Guangzhou 510275, China (e-mail: dongchh@mail2.sysu.edu.cn; sheny76@mail.sysu.edu.cn; linshy56@mail2.sysu.edu.cn; linzhzh6@mail2.sysu.edu.cn).

Yang Deng is with the School of Computing, National University of Singapore, Singapore 119077 (e-mail: ydeng@nus.edu.sg).

Digital Object Identifier 10.1109/TKDE.2023.3280182

Task	Task Sample	External Sample
CQG	**Source**: The United Kingdom, Spain, Portugal, ... and Austria have significant Russian-speaking communities. **Question**: In which European countries do many Russian immigrants live?	**Source**: (Europe, countries_within, United Kingdom); (Europe, countries_within, Spain); (Europe, countries_within, Portugal); ... (Europe, countries_within, Austria) **Question**: What countries make up Continental Europe?
FQG	**Source**: (Houston Rockets, owner_s, Leslie Alexander); (Houston Rockets, championships, 1994 NBA Finals); (Houston Rockets, championships, 1995 NBA Finals) **Question**: When did the sports team owned by Leslie Alexander win the championship?	**Source**: Houston has sports teams for every major professional league ... The Houston Rockets are a National Basketball Association (NBA) franchise based in the city since 1971. **Question**: Since what year have the Houston Rockets been a Houston team?

Fig. 1. Examples where CQG and FQG can benefit from each other. The answers are underlined and the beneficial cues are highlighted in the same colors of blue, orange, and green.

context "is birthplace of", which still contains limited background and contextual information.

Fortunately, these two tasks (i.e., CQG and FQG) can perfectly compensate for the limitations of each other, where the CQG task can learn more structural knowledge from the FQG task, and the FQG task can obtain more background and contextual information from the CQG task. Several intuitive examples are shown in Fig. 1. For CQG, more structural knowledge of the entity mentions (e.g., United Kingdom) in the source context can be found in the external samples from FQG, which can help generate the related fact "European countries" in the question that is not contained in the source context (based on the subject "Europe" and relation "countries_within"). And for FQG, more background information of the entity "Houston Rockets" can be found in the external samples from CQG, which can help infer the description "sports team" in the question that is not mentioned in the source fact triples. Meanwhile, for the task of CQG, apart from learning from the FQG task, external KGs can also be leveraged to further enhance the model's ability to discover important entities, which contain a considerable amount of structural factoid knowledge far exceeding that involved in the FQG task.

In this paper, we propose a unified framework for jointly learning the tasks of CQG and FQG, named UniCFQG. *For the task of CQG*, we present two strategies to incorporate external structural information, including the shared structural knowledge from the FQG task via a task-sharing graph module and the factoid knowledge from external KGs via a task-specific CQG knowledge module. In the CQG knowledge module, in order to alleviate the word ambiguity problem (e.g., "apple" can refer to a fruit or a company) when linking the entities in the text to those in the knowledge graph, we design a knowledge selection module, which consists of a Graph Convolutional Network (GCN) [14] to incorporate more related factoid relationships for each KG entity, and a knowledge attention mechanism to dynamically select the most suitable KG entities from a large candidate set. Meanwhile, to aggregate the KG entity embeddings and fuse the external knowledge into the original contextual representations, we design a knowledge aggregation module with a Convolutional Neural Network (CNN) to extract high-level local n-gram information and a GNN model to learn the fused knowledge. *For the task of FQG*, we incorporate more contextual information via a task-sharing passage module and a multi-level passage fusion module. To facilitate the joint learning with the CQG task to learn shared contextual knowledge, we design a word-level passage reformulation strategy to convert each KG subgraph in the FQG task into a pseudo passage. And to discover more internal contextual information, we present a multi-level passage fusion module. In specific, we first reformulate the passage representations at phrase-level and align them with that of the word-level reformulated passages, so as to enhance the correlation among words in the same fact phrases. Then we propose a Multi-level Fused GNN (MFGNN) to capture the internal and external relationships between words in the same and different fact phrases, which includes both word-level and phrase-level graph aggregation mechanisms.

Our contributions can be summarized as follows:

- We propose a multi-task learning framework for the tasks of CQG and FQG to compensate their limitations for each other, i.e., limited contextual and structural information.
- For the CQG task, we propose a task-sharing graph module to learn shared structural knowledge from the FQG task, and a task-specific knowledge module to incorporate factoid knowledge from external knowledge graphs.
- For the FQG task, we propose a task-sharing passage module to learn shared contextual knowledge from the CQG task, and a multi-level passage fusion module to extract fine-grained internal contextual knowledge.
- Experimental results on two popular datasets, i.e., SQuAD and WebQuestions, demonstrate the effectiveness of our method in enhancing the performances on both the CQG and FQG tasks.

II. Related Work

A. Contextual Question Generation

Contextual question generation (CQG) aims to generate the question given an unstructured text with contextual information, such as the reading comprehension passages.

Early works mainly rely on heavy hand-crafted rules. Heilman et al. [15] propose an overgenerate-and-rank framework, which first converts declarative sentences into questions with

manually written rules, and then ranks these questions with a logistic regression model. Labutov et al. [16] propose an ontology-crowd-relevance workflow, including representing the texts in ontology, crowdsourcing candidate question templates, and ranking relevant templates.

In recent years, driven by advances in deep learning, Seq2Seq based neural networks have been widely used. Du et al. [17] introduce an end-to-end trainable attention-based sequence learning model. Zhou et al. [18] propose a feature-rich neural encoder-decoder model with answer position. Zhao et al. [10] propose a Seq2Seq framework with a gated self-attention encoder and a maxout pointer decoder to process long texts.

Later, many works are proposed to explore the rich semantic information lying in the answer or question. Sun et al. [19] design an answer-focused and position-aware QG model, which explicitly models answer-focused question word and relative distance to the answer. Kim et al. [3] present an answer-separated Seq2Seq model that treats the answer and passage separately to better utilize the information from both sides, and present a keyword-net to extract the key information from target answer. Ma et al. [20] design a sentence-level semantic matching module and an answer position inferring module to explore the question semantics and answer position-aware features. Liu et al. [21] propose to generate question-answer pairs from unlabelled text corpora consisting of an information extractor, a neural question generator, and a neural quality controller.

Meanwhile, some researchers also exploit the structural information hidden in the context (e.g., answer-relevant relations, dependency parsing relations, etc.) to improve the consistency of generated questions. Li et al. [13] jointly model the unstructured sentence and structured answer-related relation to generate questions to the point. Pan et al. [22] construct semantic-level graphs for the input texts and encode them with an attention-based Gated Graph Neural Network, which is able to capture global structure information and generate deep questions. Chen et al. [4] propose a reinforcement learning-based graph-to-sequence model to encode the internal structural information and alleviate exposure bias. Jia et al. [23] target generating exam-like questions based on an answer-guided Graph Convolutional Network (GCN) to capture the structural inter-sentence and intra-sentence relations.

Apart from exploiting the internal information, many recent works try to perform multi-task learning (MTL) with related tasks to integrate external information. Wang et al. [24] design a multi-agent communication framework with agents of phrase extraction and question generation to generate question-worthy phrases. Zhou et al. [25] design a hierarchical multi-task learning framework for QG with language modeling. Liu et al. [26] propose to jointly train QG with clue prediction to identify potential clue words in the input passage to be copied into the target question with GCN. Jia et al. [27] propose to train QG with paraphrase generation to generate human-like questions.

Despite the tremendous advances in the CQG task, only considering internal structural information is far from enough due to its limited amount, and previous works with MTL only consider the external knowledge at contextual-level while ignoring the external structural knowledge. To solve this problem, we propose two methods to integrate external structural information, including an MTL framework with the FQG task to learn shared structural knowledge and a task-specific knowledge module to incorporate factoid knowledge from external KGs.

B. Factoid Question Generation

Factoid question generation (FQG) aims to generate questions given the related structured texts with factoid relationships, such as the knowledge base (KB) and knowledge graph (KG).

Similar to the CQG task, early works are generally based on human-created templates. Seyler et al. [28], [29] propose to use SPARQL queries as the intermediate representation and convert them into natural language questions based on predefined templates. Song et al. [30] present an in-domain QG system, which first generates the question candidates based on several templates and rich web information, and then uses a filter model for selection.

Later, the Seq2Seq framework is widely adopted. Serban et al. [31] propose an end-to-end neural model to convert the facts in knowledge bases into natural language questions with a fact encoder and a Recurrent Neural Network (RNN) based decoder. Reddy et al. [11] present a neural model to generate question-answer pairs given a KB entity and an RNN-based method to generate corresponding questions. Wang et al. [32] offer a neural generation method with the Long Short-Term Memory (LSTM) model and a new format of the input sequence.

Due to the limited background information, many works attempt to incorporate off-the-shelf contexts to generate more fluent and diversified questions. Liu et al. [5] propose an encoder-decoder model by integrating diversified off-the-shelf contexts and multi-level copy mechanisms to generate questions referring to definitive answers. Bi et al. [33] propose a knowledge-enriched, type-constrained, and grammar-guided KBQG model, which incorporates auxiliary KB information and uses a conditional copy mechanism to modulate question semantics.

Recently, multi-hop methods based on KG subgraphs have raised more and more attention, which contain more complex relationships and background information. Kumar et al. [34] propose a Transformer-based difficulty-controllable multi-hop QG model, which estimates the difficulty based on the named entity popularity. Chen et al. [6] propose to encode the KG subgraphs with a bidirectional graph-to-sequence model based on GNN and a node-level copying mechanism. Ke et al. [35] design a graph-text joint representation learning framework, which contains a structure-aware semantic aggregation module and three new pre-training tasks.

However, the KB contexts or KG subgraphs introduced by previous works in the FQG task are often composed of short words or phrases, which still contain limited background and contextual information. For instance, the relation "person/place_of_birth" has a distant supervised context "is birthplace of", and the entity "LeBron James" has a distant supervised domain and description "human" and "American Basketball Player", respectively, both of which still contain insufficient relevant knowledge. To tackle this challenge, we propose an

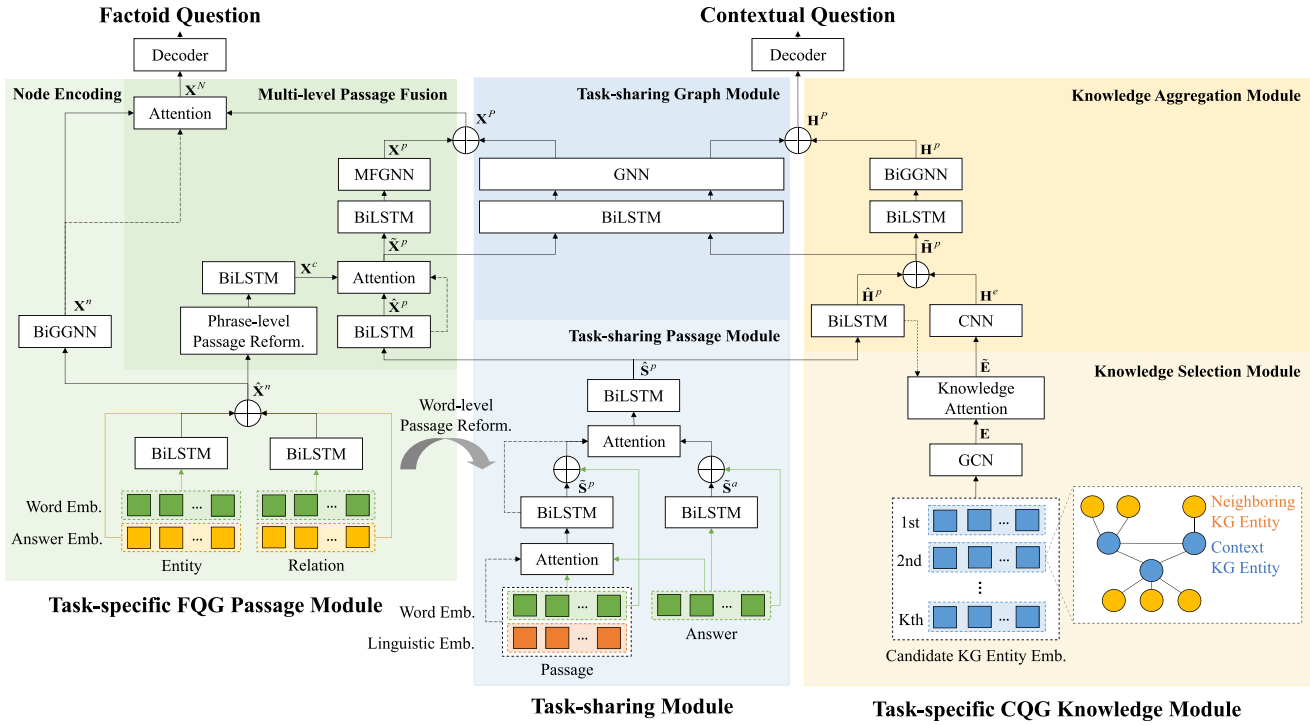

Fig. 2. An overview of UniCFQG. We design a task-sharing module to learn common task knowledge (middle), a task-specific CQG knowledge module to introduce factoid knowledge from external KGs (right), and a task-specific FQG passage module to learn fine-grained word-level knowledge (left).

MTL framework with the CQG task to learn the shared background and contextual knowledge, which can help the model in the FQG side discover relevant long-term background and contextual dependencies from the voluminous training passages in the CQG side, and is achieved by reformulating the KG subgraphs into pseudo passages at word-level. Furthermore, we present a multi-level passage fusion module to discover more internal contextual information at both word and phrase levels.

III. Method

In this paper, we focus on the tasks of CQG and FQG. For the CQG task, the purpose is to generate a question Y^c given a text sentence X and an answer A^c contained in the text, which relies on the maximum conditional likelihood $Y^c = \text{argmax}_Y P(Y|X, A^c)$. For the FQG task, each input contains a KG subgraph G and an answer A^f, where the KG subgraph is a collection of triples in the format of $(subject, relation, object)$ and the answer is an entity (i.e., subject or object) in the original KB entity set. The goal is to generate questions Y^f as calculated by $Y^f = \text{argmax}_{Y'} P(Y'|G, A^f)$. An example of these two tasks is shown in Fig. 1. Our UniCFQG mainly contains three components: a task-sharing module, a task-specific CQG knowledge module, and a task-specific FQG passage module, which is shown in Fig. 2.

A. Task-Sharing Module

In this section, we propose the task-sharing passage and graph modules to learn the shared contextual and structural knowledge between the tasks of CQG and FQG.

1) Task-Sharing Passage Module: In order to facilitate the joint learning between the tasks of CQG and FQG, we first reformulate each KG subgraph of the FQG task into a word-level pseudo passage. Specifically, for the arrangement of each passage, the facts containing the same subject and relation are merged where different objects are separated by commas ","; the facts containing the same subject are merged where different relation-object pairs are separated by semicolons ";"; and the facts containing different subjects are separated by periods ".". An illustration is shown in Fig. 3.

Afterward, to extract the shared contextual knowledge between the CQG and FQG tasks, we jointly align the passages and their corresponding answers in the task-sharing passage module. To deeply incorporate the answer information, we conduct the alignment at both word and contextual levels in a progressive manner, which is implemented by utilizing the static input embeddings and dynamic contextual representations, respectively. Specifically, given word embeddings $\mathbf{G}^p$, $\mathbf{G}^a$ of the passage and answer, and passage linguistic embedding $\mathbf{L}^p$, the *word-level* task-sharing passage embedding $\tilde{\mathbf{S}}^p$ is calculated by a bidirectional long short-term memory (BiLSTM) model [36]:

$$\tilde{\mathbf{S}}^p = \text{BiLSTM}([\mathbf{G}^p; \mathbf{L}^p; \mathbf{G}^a \tilde{\boldsymbol{\beta}}^\top]), \tag{1}$$

$$\tilde{\boldsymbol{\beta}} \propto \exp\left(f(\tilde{\mathbf{W}}^c \mathbf{G}^p)^\top f(\tilde{\mathbf{W}}^c \mathbf{G}^a)\right), \tag{2}$$

where $f(\cdot)$ represents the rectified linear unit (ReLU) function, $[;]$ denotes the concatenation operation, $\tilde{\boldsymbol{\beta}}$ denotes the attention score matrix and represents the semantic similarities among words in the passage and answer, and $\tilde{\mathbf{W}}^c$ is a trainable weight

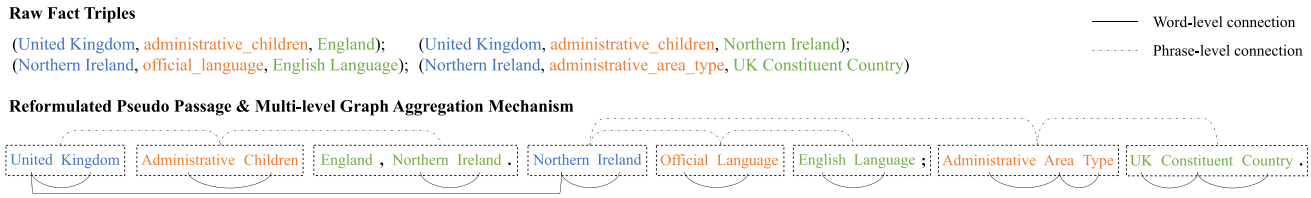

Fig. 3. Illustration for the arrangement of each FQG reformulated pseudo passage and the multi-level graph aggregation mechanism. Phrase-level connection means linking each of the words inside the connected phrases for aggregation, while word-level connection means linking the connected words directly. Phrases in blue, orange, and green denote subjects, relations, and objects, respectively.

matrix. For the passage linguistic features in CQG, apart from the case, NER, and POS embeddings, we use an entity embedding to identify whether a word is contained by a KG entity; and for those in FQG, we use a Subject Relation and Object (SRO) embedding to identify whether a word is subject, relation, or object. Then the *contextual-level* task-sharing passage embedding $\hat{\mathbf{S}}^p$ can be calculated by:

$$\hat{\mathbf{S}}^p = \text{BiLSTM}([\tilde{\mathbf{S}}^p; \tilde{\mathbf{S}}^a \hat{\boldsymbol{\beta}}^\top]), \tag{3}$$

$$\hat{\boldsymbol{\beta}} \propto \exp\left(f(\hat{\mathbf{W}}^c[\mathbf{G}^p; \tilde{\mathbf{S}}^p])^\top f(\hat{\mathbf{W}}^c[\mathbf{G}^a; \tilde{\mathbf{S}}^a])\right), \tag{4}$$

where $\tilde{\mathbf{S}}^a$ is the contextualized answer embedding and is obtained by $\tilde{\mathbf{S}}^a = \text{BiLSTM}(\mathbf{G}^a)$.

2) Task-Sharing Graph Module: To extract the shared structural knowledge, we incorporate the Graph Neural Network (GNN) into the task-sharing graph module. For the CQG task, the encoded inputs are natural passage representations enhanced by KG entity embeddings, thus we use the Bidirectional Gated GNN (BiGGNN) model [4] to encode the structural information into the passage representations, where each word in a sentence is considered a graph node, and the connections between different words are determined by their dependency parsing relationships. For the FQG task, the encoded inputs are pseudo passage representations reformulated from structured fact triples. In order to capture the internal and external relationships between words in the same and different fact phrases, we design a Multi-level Fused GNN (MFGNN) model with a multi-level graph aggregation mechanism at both word-level and phrase-level based on BiGGNN, as shown in Fig. 3. Specifically, the adjacent words in the same fact phrase (i.e., subject, relation, and object) are connected to be aggregated so as to learn internal word-level relationships, and the words in adjacent fact phrases (i.e., subject-relation, and relation-object) that belong to the same subject are connected to learn external phrase-level relationships. The first words of adjacent subjects are also connected to learn the global relationships.

Since the structural information contained in the CQG and FQG tasks is different (i.e., one is syntax dependency information while the other is factoid relationships), we introduce an additional weight-sharing structure rather than directly sharing the weights to avoid mutual interference. The task-sharing graph module takes the knowledge-enhanced passage embedding $\bar{\mathbf{H}}^p$ and the multi-level aligned passage embedding $\tilde{\mathbf{X}}^p$ as input for the CQG and FQG tasks, respectively (described in Sections II-I-B and III-C). As each KG is actually a directed graph, we adopt the bidirectional aggregation strategy in BiGGNN to deeply fuse the node information learned from both incoming and outgoing directions. For node v at the k-th graph layer, we first calculate the backward and forward aggregation vectors $\mathbf{h}^k_{\mathcal{N}_\dashv(v)}, \mathbf{h}^k_{\mathcal{N}_\vdash(v)}$ separately with an element-wise mean aggregator:

$$\mathbf{h}^k_{\mathcal{N}_\dashv(v)} = \text{Mean}(\{\mathbf{h}^{k-1}_u, \forall u \in \mathcal{N}_\dashv(v)\}), \tag{5}$$

$$\mathbf{h}^k_{\mathcal{N}_\vdash(v)} = \text{Mean}(\{\mathbf{h}^{k-1}_u, \forall u \in \mathcal{N}_\vdash(v)\}), \tag{6}$$

where $\mathcal{N}_\dashv(v), \mathcal{N}_\vdash(v)$ are incoming and outgoing neighboring node vectors of node v (including v itself). Then we use a gated fusion function to fuse the aggregated vectors of both directions. The bidirectional aggregated vector $\mathbf{h}^k_{\mathcal{N}(v)}$ at the k-th graph layer is calculated by:

$$\mathbf{h}^k_{\mathcal{N}(v)} = \mathbf{z} \odot \mathbf{h}^k_{\mathcal{N}_\dashv(v)} + (1 - \mathbf{z}) \odot \mathbf{h}^k_{\mathcal{N}_\vdash(v)}, \tag{7}$$

$$\mathbf{z} = \sigma\left(\mathbf{W}_z \mathbf{h}_z + \mathbf{b}_z\right), \tag{8}$$

where $\mathbf{z}$ is the gating vector, σ is the sigmoid function, and $\mathbf{h}_z$ is calculated by:

$$\mathbf{h}_z = \left[\mathbf{h}^k_{\mathcal{N}_\dashv(v)}; \mathbf{h}^k_{\mathcal{N}_\vdash(v)}; \mathbf{h}^k_{\mathcal{N}_\dashv(v)} \odot \mathbf{h}^k_{\mathcal{N}_\vdash(v)}; \mathbf{h}^k_{\mathcal{N}_\dashv(v)} - \mathbf{h}^k_{\mathcal{N}_\vdash(v)}\right], \tag{9}$$

where $\odot$ is component-wise multiplication operation, and $\mathbf{W}_z, \mathbf{b}_z$ are trainable parameters. After that, a Gated Recurrent Unit (GRU) model [37] is adopted, and the final aggregated vector $\mathbf{h}^k_v$ of node v at k-th layer is given by:

$$\mathbf{h}^k_v = \text{GRU}(\mathbf{h}^{k-1}_v, \mathbf{h}^k_{\mathcal{N}(v)}), \tag{10}$$

The obtained task-sharing graph representations are subsequently concatenated with task-specific passage embeddings $\mathbf{X}^p, \mathbf{H}^p$ (described in Sections III-B and III-C) to derive the final passage representations $\mathbf{X}^P, \mathbf{H}^P$ for the CQG and FQG tasks, respectively.

B. Task-Specific CQG Knowledge Module

In order to enhance the model's ability to discover important entities in the source passages and generate consistent questions, we propose a knowledge selection and aggregation module to incorporate more structural knowledge from external KGs.

1) Knowledge Selection Module: For the CQG task, the inputs of the knowledge selection module are important entities in the source passages and their factoid relationships are not assigned like the inputs of the FQG task, thus we use Graph Convolutional Network (GCN) [14] to learn their relationships with each other. In addition, the entities in CQG are linked with

external KGs where each entity in the source passage might corresponds to multiple KG entity embeddings, thus we propose a knowledge attention method to select the most semantically suitable entity embedding, which is able to alleviate the word ambiguity problem (e.g., "apple" can refer to a fruit or a company) as pointed out by Deng et al. [38].

Sepcifically, to discover more factoid relationships for each candidate entity, we utilize GCN to learn the relationships between each entity and its one-hop neighboring entities in KG, as well as the relationships among all the entities in a sentence. The aggregated entity embedding $\mathbf{E}^{(l)}$ at the l-th graph layer can be formulated by:

$$\mathbf{E}^{(l)} = \sigma\left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^{(l-1)}\mathbf{W}^{(l)}\right), \tag{11}$$

where $\sigma(\cdot)$ is an activation function, $\mathbf{W}^{(l)}$ is a trainable weight matrix at the l-th layer, and $\mathbf{A}, \mathbf{D}$ is the adjacency matrix and degree matrix, respectively (with added self-connections).

Then we apply a knowledge attention method to calculate the similarity between each candidate entity embedding and the task-specific context representation $\hat{\mathbf{H}}^p = \text{BiLSTM}(\hat{\mathbf{S}}^p)$. Given top-K candidate entity embeddings of the t-th sentence word $\mathbf{E}_t = \{\mathbf{E}_{t1}, \mathbf{E}_{t2}, \ldots, \mathbf{E}_{tK}\}$ (based on their occurrence frequency) from the last GCN layer and the average-pooled passage embedding $\hat{\mathbf{H}}^p_{avg} = \text{AVG}(\hat{\mathbf{H}}^p)$, the best entity embedding $\tilde{\mathbf{E}}_t$ of the t-th word is formulated as:

$$\tilde{\mathbf{E}}_t = \sum_{i=1}^{K} \alpha_{ti}\mathbf{E}_{ti}, \tag{12}$$

$$\alpha_{ti} = \frac{\exp(\mathbf{W}_m\mathbf{M}_{ti})}{\sum_{j=1}^{K}\exp(\mathbf{W}_m\mathbf{M}_{tj})}, \tag{13}$$

$$\mathbf{M}_{ti} = \tanh\left(\mathbf{W}_{ctx}\mathbf{H}^p_{avg} + \mathbf{W}_{ent}\mathbf{E}_{ti}\right), \tag{14}$$

where $\mathbf{W}_m, \mathbf{W}_{ctx}, \mathbf{W}_{ent}$ are learnable weight matrices, and α_{ti} is the knowledge attention score applied on the i-th candidate KG entity embedding of the t-th word.

2) Knowledge Aggregation Module: After obtaining the best entity embeddings, we feed them into the knowledge aggregation module to capture their global relationships. Since the entities are always phrases and different entities might not be adjacent, we use a Convolutional Neural Network (CNN) with various sizes of filters to capture the high-level local n-gram information. The local feature $\mathbf{H}^e_t$ extracted by the t-th move with filter size n is calculated by:

$$\mathbf{H}^e_t = \tanh(\mathbf{W}_c * \mathbf{e}_t + \mathbf{b}_c), \tag{15}$$

$$\mathbf{e}_t = \{\tilde{\mathbf{E}}_{t-(n-1)/2}, \ldots, \tilde{\mathbf{E}}_t, \ldots, \tilde{\mathbf{E}}_{t+(n-1)/2}\}, \tag{16}$$

where $*$ is the convolution operation, $\mathbf{W}_c, \mathbf{b}_c$ are learnable convolution kernel matrix and bias vector, respectively, and $\mathbf{e}_t$ is the local n entity embeddings at the t-th filter move. Then the derived local feature $\mathbf{H}^e$ is concatenated with the task-specific context representation $\hat{\mathbf{H}}^p$ forming $\tilde{\mathbf{H}}^p = [\hat{\mathbf{H}}^p; \mathbf{H}^e]$. Finally, A BiLSTM model followed by a BiGGNN model is applied to refine the global sequential information and generate the final task-specific CQG passage representation $\mathbf{H}^p$.

C. Task-Specific FQG Passage Module

In this section, we introduce two types of encoding methods to learn the contextual and structural information at both node-level and word-level, including a node encoding module and a multi-level passage fusion module.

1) Node Encoding Module: For the FQG task, the inputs of the node encoding module are fact triples and different entities in each triple have contextual dependencies with each other, thus we use BiLSTM to learn contextual knowledge and use BiGGNN to extract bidirectional factoid relationships among entities, and finally derive the node-level representation of each fact phrase.

Since each KG is a directed graph and might contain various relation types between two entities, we utilize the Levi graph [39] to transform the original KGs into bipartite graphs. Specifically, the entities and relations of each KG subgraph are all treated as graph nodes, and new edges are added to connect them. Then we use two BiLSTM models to encode the texts in each entity and relation separately and use the concatenation of the last forward and backward hidden states as the corresponding node embeddings. In addition, we add answer embeddings and concatenate them with their corresponding node embeddings to incorporate the answer information. Finally, the concatenated node embeddings $\hat{\mathbf{X}}^n$ are fed into a BiGGNN model to learn the relationships between different nodes and derive the refined node representation $\mathbf{X}^n$.

2) Multi-Level Passage Fusion Module: In order to learn fine-grained word-level information, we propose a multi-level passage fusion module to align the representations between the word-level and phrase-level reformulated pseudo passages. The knowledge of different levels of pseudo passages is extracted separately and is then fused via an attention mechanism and the Multi-level Fused GNN model.

We first reformulate the KG subgraphs into word-level pseudo passages and extract their word-level contextual information as described in Section III-A. Afterward, to enhance the correlation among words in the same entity or relation, we perform phrase-level passage reformulation to incorporate phrase-level contextual information. In detail, based on the task-specific passage embedding $\hat{\mathbf{X}}^p = \text{BiLSTM}(\hat{\mathbf{S}}^p)$, we first rearrange node embedding $\hat{\mathbf{X}}^n$ into passage order similar with the word-level arrangement strategy in Section III-A, which however is performed at phrase-level. Then we extract their contextualized embedding $\mathbf{X}^c = \text{BiLSTM}(\hat{\mathbf{X}}^n)$, and use an attention mechanism to align the information between the word-level and phrase-level passage embeddings:

$$\tilde{\mathbf{X}}^p = [\hat{\mathbf{X}}^p; \mathbf{X}^c\tilde{\gamma}^\top], \tag{17}$$

$$\tilde{\gamma} \propto \exp\left(f(\tilde{\mathbf{W}}^f\hat{\mathbf{X}}^p)^\top f(\tilde{\mathbf{W}}^f\mathbf{X}^c)\right). \tag{18}$$

Subsequently, $\tilde{\mathbf{X}}^p$ is fed into a BiLSTM model followed by an MFGNN model with the graph aggregation mechanism described in Section III-A to learn word-level structural knowledge and derive $\mathbf{X}^p$. Finally, we adopt a similar attention mechanism to incorporate passage information into node embedding $\mathbf{X}^n$ and derive the final task-specific FQG node embedding $\mathbf{X}^N$ as

follows:

$$\mathbf{X}^N = [\mathbf{X}^n; \mathbf{X}^p \gamma^\top], \tag{19}$$

$$\gamma \propto \exp\left(f(\mathbf{W}^f \mathbf{X}^n)^\top f(\mathbf{W}^f \mathbf{X}^p)\right). \tag{20}$$

D. Training

During training, we use the cross-entropy loss and coverage loss [40] to train our model. The total loss function $\mathcal{L}$ is formulated as:

$$\mathcal{L} = \sum_t -\log P(y_t^*|X, y_{<t}^*) + \lambda \sum_t \sum_i \min(a_i^t, c_i^t), \tag{21}$$

where y_t^* is the word at the t-th position of the ground-truth output sequence, X is the input text or KG subgraph, a_i^t, c_i^t are the i-th element of attention and context vectors at the t-th time step respectively, and λ controls the weight of coverage loss.

E. Decoding

Following See et al. [40], we use an attention-based unidirectional LSTM model with a copying mechanism to be the decoder. We use the max-pooled passage and node embeddings of $\mathbf{H}^P, \mathbf{X}^N$ to initialize the decoder of CQG and FQG, respectively, and take $\mathbf{H}^P, \mathbf{X}^N$ as the attention memory.

IV. Experiment

A. Datasets and Metrics

For the evaluation of the CQG task, we use the Stanford Question Answering Dataset (SQuAD) [41], which is a large reading comprehension dataset consisting of questions posed by crowd workers based on Wikipedia articles. The answer to each question is a segment of text from the corresponding passages, and there are totally 107,785 question-answer pairs on 536 articles. We adopt the data split in Zhou et al. [18], which contains 86,635/8,965/8,964 examples for the train/development/test set.

For the evaluation of the FQG task, we use the WebQuestions benchmark[1] with several multi-hop question answering datasets based on Freebase, including WebQuestionsSP [42] and ComplexWebQuestions [43]. The samples in WebQuestions are composed of tuples $\{(Q, G, E)\}$, where Q is a natural language question, G is the KG subgraph from which the question is derived, and E is the set of answer entities corresponding to the question. In total, WebQuestions contain 25,703 entities, 672 relations, 2 to 100 hops, and 22,989 instances. We adopt the data split ratio of 80%/10%/10% for the train/development/test set as in Kumar et al. [34].

Following previous works, we use both automatic and human evaluation metrics to assess our model. We adopt BLEU [44], ROUGE [45], and METEOR [46] for automatic evaluation. For human evaluation, we mainly consider three aspects, including: (i) *fluency*, which evaluates whether the question is grammatically correct and fluent; (ii) *relevancy*, which evaluates whether the question is relevant to the source text; and (iii) *answerability*, which evaluates whether the question can be answered by the given answer. We randomly select 50 samples from the generated questions and ask three annotators to score them with the rating score in the range 1-5.

B. Baselines

For the task of CQG, we compare with the following baseline methods:

- *NQG++* [18] consists of a feature-rich encoder and an attention-based decoder.
- *s2sa-at-mp-gsa* [10] proposes a gated self-attention encoder with a maxout pointer in decoder.
- *ASs2s* [3] proposes an answer-separated model to extract answer information.
- *LM-QG* [25] introduces language modeling as an auxiliary task to help QG in a hierarchical MTL structure.
- *Sent-Relation* [13] introduces answer-relevant relation to help generated questions keep to the point.
- *CQC-QG* [26] presents a multi-task labeling strategy with GCN to discover potential clue words to be copied into the target question.
- *CS2S-VR-A* [21] proposes to generate question-answer pairs with the information extractor, question generator, and quality controller.
- *PG-QG* [27] proposes an MTL framework between paraphrase and question generation.
- *G2S+BERT* [4] proposes a syntax-based static Graph2Seq model with a deep answer alignment network.
- *EQG-RACE* [23] proposes an answer-guided GCN for examination-type QG.

And for the task of FQG, we compare with the following baseline methods:

- *L2A* [17] is an LSTM-based Seq2Seq model with attention mechanism and takes linearized sequences of KG subgraphs as input.
- *Transformer* [47] is a Transformer-based encoder-decoder model with sequences of word embeddings as input.
- *MHQG+AE* [34] is a Transformer-based model with answer encoding and takes sequences of TransE embeddings as input.
- *G2S+AE* [6] proposes a bidirectional Graph2Seq model to encode the input KG subgraphs.
- *T5* [48] and *BART* [49] are the state-of-the-art pre-trained models for text-to-text generation and are applied for the KG-to-text task with linearized KGs.
- *JointGT (T5)* and *JointGT (BART)* [35] designs several pre-training tasks to enhance graph-text alignment based on backbones of T5 and BART.

C. Implementation Details

For the knowledge graph used in the CQG knowledge module, we adopt a subset of FreeBase [50], FB5M,[2] as the KG source, which contains 3,988,105 entities, 7,523 relations, and 17,872,174 facts. Since the entity linking is not the focus of

[1]https://github.com/liyuanfang/mhqg

[2]https://research.fb.com/downloads/babi/

TABLE I
AUTOMATIC EVALUATION RESULTS ON THE SQUAD DATASET FOR THE TASK OF CQG. *BOLD* AND UNDERLINED INDICATE METHODS WITH BEST AND SECOND-BEST PERFORMANCES, RESPECTIVELY. METHODS WITH † ARE CONDUCTED WITH THE RELEASED CODES

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR
NQG++ [18]	42.36	26.33	18.46	13.51	-	-
s2sa-at-mp-gsa [10]	44.51	29.07	21.06	15.82	44.24	19.67
LM-QG [25]	42.80	28.43	21.08	16.23	-	-
Sent-Relation [13]	44.40	29.48	21.54	16.37	44.73	20.68
CGC-QG [26]	46.58	30.90	22.82	17.55	44.53	21.24
CS2S-VR-A [21]	45.28	29.58	21.45	16.13	43.98	20.59
PG-QG [27]	43.63	29.21	21.79	16.93	-	20.58
G2S+BERT [4] †	47.13	31.70	23.40	17.88	45.82	21.59
EQG-RACE [23] †	45.04	29.71	21.74	16.50	44.29	20.34
UniCFQG (ours)	**47.56**	**32.19**	**23.98**	**18.52**	**46.24**	**21.88**

TABLE II
AUTOMATIC EVALUATION RESULTS ON THE WEBQUESTIONS DATASET FOR THE TASK OF FQG

Methods	BLEU-4	ROUGE-L	METEOR
L2A [17]	6.01	26.95	25.24
Transformer [47]	8.94	32.63	13.79
MHQG+AE [34]	11.57	35.53	29.69
T5 [48]	28.78	55.12	30.55
BART [49]	29.61	55.42	31.48
G2S+AE [6]	29.45	55.45	30.96
JointGT (T5) [35]	28.95	54.47	31.29
JointGT (BART) [35]	**30.02**	**55.60**	**32.05**
UniCFQG (ours)	29.96	55.54	31.49

this work, we use a widely-adopted package, TagMe,[3] as an off-the-shelf tool for entity linking in our experiments, which is widely adopted in related studies [51], [52]. Other recent libraries can also be considered to extract the entity mentions, such as BLINK.[4]

We use both pre-trained 300-dim GloVe [53] and 1024-dim BERT [54] embeddings to initialize the word embeddings, and use 300-dim TransE [55] embeddings to initialize the KG entity embeddings, which are trained with OpenKE [56] under the default settings. We keep the most frequent 70,000 and 20,000 words in the training set of the CQG and FQG tasks, respectively, and select the top-5 most frequently mentioned entities and their top-5 one-hot neighbors as candidates in the CQG knowledge module. The dimensions of the case, POS, and NER embeddings in CQG are set to 3, 12, and 8, respectively, while those of the answer and SRO embeddings in FQG are both set to 32. The dimension of entity embeddings in CQG is set to 9 to align with the input dimension of FQG (i.e., 32). The dimensions of all other hidden layers are set to 300. The number of layers for the graph networks in CQG and FQG is set to 3 and 4, respectively. The convolutional filter sizes are set to 2 and 3. The variational dropout rates [57] over the word embeddings and RNN layers are set to 0.4 and 0.3, respectively.

[3]https://github.com/marcocor/tagme-python
[4]https://github.com/facebookresearch/BLINK

During training, we use Adam [58] as the optimizer and set the initial learning rate to 0.001. The learning rate is reduced by half when the validation score (BLEU-4) stops improving for three epochs, and the training process is terminated when there are no improvements for ten epochs. The batch size is set to 50. The beam search widths of CQG and FQG are set to 15 and 9, respectively. The coverage loss ratio λ is set to 0.4 and 0 for CQG and FQG, respectively. We use label smoothing for FQG and set the ratio to 0.2. We also adopt schedular teacher forcing [59] to alleviate the exposure bias problem, where the initial teacher forcing probability is set to 0.75 and is exponentially increased to $0.75 * 0.9999^i$ at the i-th training step. The hyperparameters are tuned on the development set, and all the experimental results are averaged over three runs. The experiments are conducted on a GeForce RTX 3090 GPU.

D. Experimental Results

1) Automatic Evaluation: The automatic evaluation results on the SQuAD and WebQuestions test sets are reported in Tables I and II. For the CQG task, compared with the previous method G2S+BERT, our model achieves an improvement of 0.64%/0.42%/0.29% in the metric of BLEU-4/ROUGE-L/METEOR. And compared with EQG-RACE, the corresponding improvement is 2.02%/1.95%/1.54%. For the FQG task, compared with the counterpart G2S+AE, the gain of BLEU-4/METEOR is 0.51%/0.53%. And in comparison to previous pre-trained methods, our method can still achieve competitive results. For example, compared with T5, the improvement of BLEU-4/ROUGE-L/METEOR is 1.18%/0.42%/0.94%. The results show that UniCFQG has outstanding performances on both the CQG and FQG tasks.

In our opinion, the reason that traditional pre-trained methods not working well on the FQG task can be attributed to the significantly different input formats of training samples between pre-training and fine-tuning. During pre-training, the inputs are long consecutive passages, while during fine-tuning, each input is composed of separate factoid triples. Such a discrepancy leads to the less training effectiveness. And for JointGT, since it designs several graph-to-text pre-training tasks based on huge amount of crawled pre-training dataset, it has better performances than the

TABLE III
HUMAN EVALUATION RESULTS FOR THE TASK OF CQG

Methods	Flu.	Rel.	Ans.
CGC-QG [26]	3.89	3.53	3.33
G2S+BERT [4]	4.15	3.75	3.25
EQG-RACE [23]	3.79	3.41	2.71
UniCFQG (ours)	**4.33**	**4.15**	**3.75**

TABLE IV
HUMAN EVALUATION RESULTS FOR THE TASK OF FQG

Methods	Flu.	Rel.	Ans.
T5 [48]	4.25	4.33	4.05
G2S+AE [6]	4.23	4.49	4.27
JointGT (T5) [35]	4.07	4.37	4.29
UniCFQG (ours)	**4.39**	**4.65**	**4.52**

vanilla pre-trained models T5 and BART, and the performance of JointGT (BART) is better than our methods. The number of parameters in our framework is around 50 million, while that of JointGT (T5) and JointGT (BART) is 265 and 160 million, respectively.

2) Human Evaluation: The human evaluation results are listed in Tables III and IV. As can be seen, our UniCFQG still has better effects on both CQG and FQG tasks. In terms of all the metrics including fluency (flu.), relevancy (rel.), and answerability (ans.), our method shows significant improvements against its counterparts. For the CQG task, our method achieves an improvement of 0.18/0.40/0.42 in the metric of flu./rel./ans. compared with other best-performed baselines. And for the FQG task, the corresponding improvement is 0.14/0.16/0.23. These results validate the superiority of the proposed methods from practical perspectives.

E. Ablation Study

To assess the effectiveness of each module in our model, we conduct an ablation study by progressively removing each part from our UniCFQG as shown in Tables V and VI. We first sequentially remove the BERT embeddings and the task-sharing module, and as can be seen, the task-sharing module contributes 0.24%/0.23%/0.26% to the performances of BLEU-4/ROUGE-L/METEOR on the CQG task and contributes 0.22% to the performance of BLEU-4 on the FQG task, which demonstrate that both of the CQG and FQG tasks can learn helpful knowledge from each other. Then we separately remove the CQG knowledge module and FQG passage module, where the performances are further decreased by 0.33%/0.31%/0.22% and 0.13%/0.46%/0.16% on the metric of BLEU-4/ROUGE-L/METEOR in the CQG and FQG task, respectively. This shows the effectiveness of the external factoid knowledge and fine-grained word-level knowledge brought by the reformulated pseudo passages. Finally, we remove all the components of our UniCFQG, and the performance of ROUGE-L on the CQG task is further decreased by 0.16%, whilst the performance of BLEU-4/ROUGE-L/METEOR on the FQG task is decreased by 0.39%/0.12%/0.29%. Meanwhile, compared with those baselines in Tables I and II, we can observe that after removing all the modules, the performances of our method are lower than many baselines. For example, in CQG, our BLEU-4 score is lower than that of CGC-QG, PG-QG, and G2S+BERT. And in FQG, our BLEU-4 score is lower than that of BART, G2S+AE, and JointGT (BART). The above results further verify the advantage of each module of our method.

In addition, we evaluate the effectiveness of attention mechanisms in different modules of our model as shown in Table VII. We mainly report the average score of the CQG and FQG tasks in terms of BLEU-4, ROUGE-L, and METEOR. We first evaluate the word-level and contextual-level answer alignment attention mechanisms in the task-sharing passage module, which are respectively formulated by (2) and (4). From the results, we can observe that both of these attention mechanisms benefit the overall performance; and by removing both of them, the performance of BLEU-4/ROUGE-L/METEOR is significantly decreased by 2.35%/3.61%/2.02%. This demonstrates that the answer information is crucial for generating to-the-point questions. Then we evaluate the knowledge attention mechanism in the task-specific CQG knowledge selection module as formulated by (13). We only select the top-1 frequent KG entity embedding for each entity in the source passage and remove the knowledge attention mechanism to access its effectiveness. From the results, we can see that knowledge attention also contributes a lot to the model performance, especially for the metric of BLEU-4, which is decreased by 0.29% after removing the attention mechanism. This shows that knowledge attention can effectively alleviate the word ambiguity problem and help the model generate more semantically consistent questions. Finally, we evaluate the multi-level attention mechanism between word-level and phrase-level passage embeddings in the task-specific FQG multi-level passage fusion module, which is formulated by (18). We remove both of the phrase-level passage reformulation and multi-level attention mechanisms to show the attention effect, and the performance of BLEU-4/ROUGE-L/METEOR is decreased by 0.24%/0.16%/0.15%. This proves that enhancing the correlation among words in the same entity or relation is important and beneficial, and conducting attention with such fine-grained knowledge can help the model generate more fluent questions.

F. Analysis for Consistency and Diversity

In this section, we use additional metrics to evaluate the consistency and diversity of generated questions in the tasks of CQG and FQG, respectively, as shown in Tables VIII and IX. *For the CQG task*, we use the entity matching ratio (i.e., the proportion of matching entities, which also appear in the corresponding source passages, between the generated and ground-truth questions) to evaluate the consistency between generated questions and source contexts; and use entity recognition ratio (i.e., the proportion of generated entities from all entities in source passages)

TABLE V
ABLATION RESULTS FOR THE EFFECTIVENESS OF DIFFERENT MODULES IN THE CQG TASK. CK, FP, TS DENOTES THE CQG KNOWLEDGE MODULE, THE FQG PASSAGE MODULE, AND THE TASK-SHARING MODULE, RESPECTIVELY

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR
UniCFQG (ours)	**47.56**	**32.19**	**23.98**	**18.52**	**46.24**	**21.88**
w/o BERT	46.72	31.08	22.87	17.47	45.06	21.21
w/o BERT, TS	46.05	30.58	22.52	17.23	44.83	20.95
w/o BERT, TS, CK	45.73	30.23	22.18	16.90	44.52	20.73
w/o BERT, TS, FP	46.36	30.75	22.60	17.25	44.55	20.95
w/o BERT, TS, CK, FP	45.84	30.27	22.19	16.90	44.36	20.73

TABLE VI
ABLATION RESULTS FOR THE EFFECTIVENESS OF DIFFERENT MODULES IN THE FQG TASK

Methods	BLEU-4	ROUGE-L	METEOR
UniCFQG (ours)	**29.96**	**55.54**	**31.49**
w/o BERT	29.71	55.38	31.33
w/o BERT, TS	29.49	55.41	31.33
w/o BERT, TS, CK	29.49	55.23	31.31
w/o BERT, TS, FP	29.36	54.95	31.17
w/o BERT, TS, CK, FP	29.10	55.11	31.02

TABLE VII
ABLATION RESULTS FOR THE EFFECTIVENESS OF ATTENTION MECHANISMS IN DIFFERENT MODULES

Methods	BLEU-4	ROUGE-L	METEOR
UniCFQG w/o BERT	**23.59**	**50.22**	**26.27**
w/o TS word-level	23.36	49.85	26.04
w/o TS context.-level	23.31	50.03	26.12
w/o TS all	21.24	46.61	24.25
w/o CQG know.-level	23.30	50.16	26.16
w/o FQG multi-level	23.35	50.06	26.12

TABLE VIII
RESULTS FOR THE CONSISTENCY OF GENERATED QUESTIONS IN CQG

Methods	Match ratio	Recog. ratio
G2S+BERT [4]	35.85	46.62
UniCFQG (ours)	**41.51**	**48.32**
w/o TS, FP	39.62	47.13
w/o TS, FP, CK	35.85	46.73

TABLE IX
RESULTS FOR THE DIVERSITY OF GENERATED QUESTIONS IN FQG

Methods	Distinct-1	Distinct-2
G2S+AE [6]	8.92	25.35
UniCFQG (ours)	**9.10**	**27.51**
w/o TS	9.03	26.60
w/o TS, FP	8.86	26.08

to assess the model's ability to recognize potential entities. From the results, we can see that our method significantly outperforms its counterpart G2S+BERT with an improvement of 5.66%/1.70% on the metric of matching/recognition ratio, and both the task-sharing module and CQG knowledge module contribute a lot to generate consistent questions with improvements of 1.89%/1.19% and 3.76%/0.40%, respectively. *For the FQG task*, we use Distinct-n [60] to measure the diversity. From the results, we can observe that our method also outperforms its counterpart G2S+AE by a large margin with an improvement of 0.18%/2.16% on the metric of Distinct-1/2, and both the task-sharing module and FQG passage module have large positive impacts on generating diversified questions with contributions of 0.07%/0.91% and 0.17%/0.52%, respectively.

G. Analysis for Task Mutual Benefit

To better illustrate the mutual benefit between the tasks of CQG and FQG, we conduct several experiments with different proportions of training samples. The results are shown in Fig. 4, and as can be seen, the BLEU-4 improvements of our method against the single-task learning (STL) methods for both tasks get more and more prominent as the sample ratio increases. Specifically, for the CQG task, the BLEU-4 improvement between STL and our UniCFQG increases from nearly 0.2% to nearly 0.7% as the sample ratio increases from 10% to 30%; and for the FQG task, the BLEU-4 improvement increases from nearly 0.6% to nearly 1.3%. This indicates that more related information can be discovered by increasing the training samples and leads to better task mutual benefit.

H. Case Study

In this section, we provide several examples of generated questions for the tasks of CQG and FQG in Fig. 5 to intuitively show the effectiveness of our method.

In the first case of CQG, benefit from the task-sharing module and task-specific CQG knowledge module, the KG entity mention "Super Bowl 50 halftime show" gains more attention and is entirely predicted in the generated question, while its counterpart G2S+BERT fails to capture the entire entity and only predict part of it, i.e., "Super Bowl". And there exists a repetition and evidence error of "Bruno Mars" in the prediction of G2S+BERT corresponding to "Super Bowl XLVIII" in the source, while our method avoids such a problem. In the second

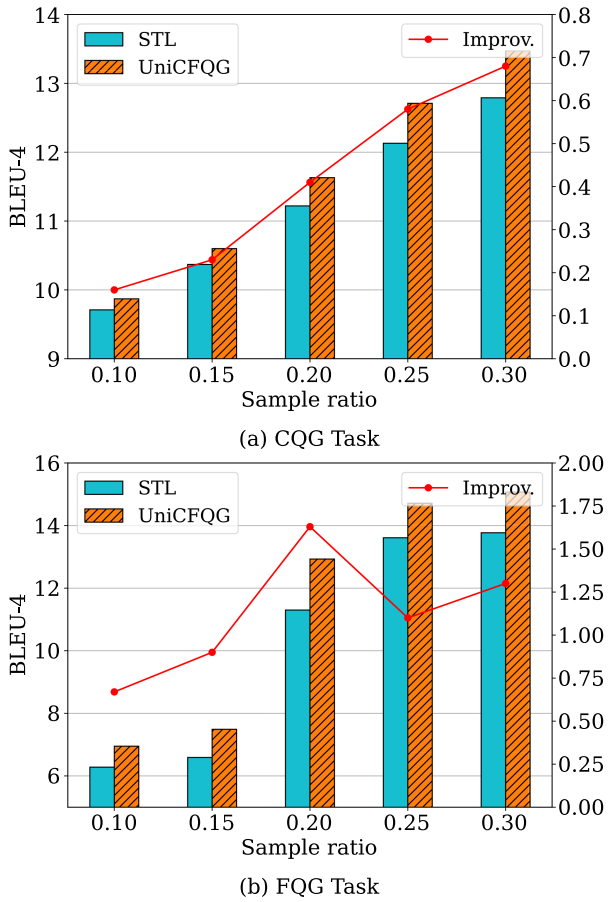

Fig. 4. Results for the task mutual benefit between CQG and FQG with different sample ratios.

Task	Example
CQG	**Source**: The Super Bowl 50 halftime show was headlined by the British rock group Coldplay with special guest performers Beyoncé and Bruno Mars, who headlined the Super Bowl XLVII and Super Bowl XLVIII halftime shows, respectively. **Reference**: Which group headlined the Super Bowl 50 halftime show? **G2S+BERT** [4]: Which british rock group headlined the Super Bowl and Bruno Mars and Bruno Mars? **UniCFQG (ours)**: Who headlined the Super Bowl 50 halftime show?
	Source: Major bequests include Reverend Chauncy Hare Townshend's collection of 154 gems bequeathed in 1869, Lady Cory's 1951 gift of major diamond jewellery from the 18th and 19th centuries, ... **Reference**: In which year was Reverend Chauncy Hare Townshend's collection of gems was bequeathed to the museum? **G2S+BERT** [4]: In what year was Lady Cory's collection of 154 gems bequeathed? **UniCFQG (ours)**: In what year was Townshend's collection of 154 gems bequeathed?
FQG	**Source**: (Houston Rockets, championships, 1994 NBA Finals); (Houston Rockets, championships, 1995 NBA Finals); (Houston Rockets, head_coach, Kevin McHale) **Reference**: In what seasons did the NBA team coached by Kevin McHale win the championship? **G2S+AE** [6]: In what years did the coach of the team whose head coach is Kevin McHale? **UniCFQG (ours)**: In what years did the basketball team whose head coach is Kevin McHale win the championship?
	Source: (Arthur Miller, influenced_by, William Shakespeare); (William Shakespeare, influenced_by, Lucian) **Reference**: Who influenced Arthur Miller that was influenced by Lucian? **G2S+AE** [6]: Who influenced by Lucian and was influenced by Lucian? **UniCFQG (ours)**: Who was influenced by Lucian and influenced Arthur Miller?

Fig. 5. Case study for tasks of CQG and FQG. The answers are underlined, and the correct and incorrect predicted words are displayed in blue and red, respectively.

case, the prediction of G2S+BERT also has an evidence error of "Lady Cory's" corresponding to "major diamond jewellery" in the source, while our method makes a correct prediction.

Moreover, in the first case of FQG, thanks to the task-sharing module and task-specific FQG passage module, the model can generate more fluent and diversified questions and learn more fine-grained knowledge, such as "win the championship", while its counterpart G2S+AE fails to achieve this. G2S+AE also has a repetition and evidence error of "the coach of the team", while our method avoids such a problem. In the second case, G2S+AE ignores the subject "Arthur Miller" in the source and repetitively predicts "influenced by Lucian", while our method correctly predicts the fact "influcenced Arthur Miller". These cases strongly demonstrate the effectiveness of our method.

I. Error Analysis

In this section, we provide the error analysis results of our method based on its several failure cases on the CQG and FQG tasks to depict the limitations of our method and promote the advance of further studies. We analyze 50 cases in total (25 of each task) as shown in Fig. 6, and the error cases can be primarily divided into five categories including repetition, answer mismatch, information loss, evidence error, and syntax error.

Repetition: The repetition of words or phrases is the most frequent problem in both CQG and FQG tasks, which accounts for around 28% and 52% respectively. The significant higher frequency of the problem in FQG might be caused by its limited input contents, i.e., the model might frequently attend to the same source content during decoding and generate similar words. Repetition is a common phenomenon in text generation tasks, which is induced by the maximization-based sampling strategy during the decoding process (e.g., beam search). Currently, many methods have been proposed to tackle this problem [61], [62], and there is still a long distance and large improvement room to solve it.

Answer Mismatch: Answer mismatch is a frequent problem in QG tasks, which refers to the phenomenon that the generated question is unable to be answered by the given answer in the source inputs. Statistically, there are about 28% cases in CQG have such mismatch problem. On the contrary, due to the limited information contained in the inputs of FQG, it is easier for the model to identify the answer phrases and thus only 8% of the FQG cases have such problem.

Error Type	Example
Repetition	**Source**: The reforms provoked disturbances, including ... **Reference**: What did the reforms cause? **Prediction**: What did the reforms of the reforms?
Answer Mismatch	**Source**: The V&A covers 12.5 acres (51,000 m^2) and 145 galleries. **Reference**: How many galleries does the V&A have? **Prediction**: How many acres does the V&A have?
Information Loss	**Source**: For the 2012-13 school year annual tuition was $38,000, with a total cost of attendance of $57,000. **Reference**: What is the total cost of attendance in 2012-13? **Prediction**: What was the total cost of attendance in 2012?
Evidence Error (Intrinsic)	**Source**: The Soviet Union and the People's Republic of China supported post World War II communist movements ... **Reference**: Who along with Russia supported post WW-II communist movements? **Prediction**: Along with the Soviet Union, what country supported the World War II?
Evidence Error (Extrinsic)	**Source**: (Arthur Miller, influenced_by, Henrik Ibsen); (Henrik Ibsen, place_of_burial, Vår Frelsers gravlund) **Reference**: What person buried in Vår Frelsers gravlund was an influence on Arthur Miller? **Prediction**: Who influenced Arthur Miller and was buried in Vår Frelsers gravlund re search?
Syntax Error	**Source**: (China, tv_shows_filmed_here, The Bride with White Hair); (China, official_language, Standard Mandarin). **Reference**: What is the main language used where the Bride with White Hair was filmed? **Prediction**: The Bride with White Hair was filmed in what language is spoken?

Fig. 6. Examples for the most frequent error types of our method in the tasks of CQG and FQG. The incorrect predicted words are displayed in red.

Information Loss: The information loss mainly occurs in CQG with a proportion of 16% due to the excess input information. In most cases, the missing information is auxiliary, such as the specific location, time, person, etc. This indicates that more effective methods need to be designed to discover the structured auxiliary information in the source contexts.

Evidence Error: The evidence error can be categorized into *intrinsic and extrinsic evidence errors*, where the intrinsic error refers to the incorrect combination of phrases or clauses from the source inputs, and the extrinsic error refers to the introduction of irrelevant words that are not contained by the source inputs. In CQG, since the relatively more sufficient input contents, there are mainly intrinsic errors with a proportion of 8%. And in FQG, due to the limited input information, there are mainly extrinsic errors with a proportion of 12% as well as 4% intrinsic errors.

Syntax Error: The syntax error is also a common problem in text generation tasks, including the incorrect choices of spelling, punctuation, grammars, etc. In CQG and FQG, the syntax error accounts for around 20% and 24%, respectively. Recently, many researchers have been committed to solving this problem [63], [64], which is still a challenging problem and needs deeper and further research.

V. Conclusion and Future Work

In this paper, we propose a multi-task learning framework to jointly learn the tasks of CQG and FQG. For the CQG task, a task-specific knowledge module with a knowledge selection and aggregation module is designed to incorporate more factoid knowledge from external KGs and alleviate the word ambiguity problem. For the FQG task, a task-specific passage module with a a multi-level passage fusion module is proposed to extract the fine-grained knowledge at word-level. In addition, two types of task-sharing modules are presented to learn shared contextual and structural knowledge, where the input formats of CQG and FQG are aligned by reformulating the fact triples in FQG into pseudo passages similar with CQG. Extensive experimental results on two widely adopted datasets show the effectiveness of our method.

In the future, we will integrate more types of CQG and FQG tasks (e.g., multi-hop CQG, table-to-text FQG, etc.) to investigate the transferability of our UniCFQG in the multi-task learning scenario. Moreover, we will explore more efficient method to conduct the multi-task learning, such as the prompt-based learning techniques, and try to discover more effective MTL strategies in the low-resource setting to aid the demands in the industry. We will also investigate more appropriate strategies of concatenating the contextual and knowledge representations in the task-specific CQG knowledge module to alleviate the problems of structural alignment and semantic proximity diversity. Finally, we will explore the utilization of pre-trained language models instead of the LSTM-based model structure in our MTL framework.

References

[1] B. N. Patro, S. Kumar, V. K. Kurmi, and V. Namboodiri, "Multimodal differential network for visual question generation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 4002–4012.
[2] R. Krishna, M. Bernstein, and L. Fei-Fei, "Information maximizing visual question generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2008–2018.
[3] Y. Kim, H. Lee, J. Shin, and K. Jung, "Improving neural question generation using answer separation," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 6602–6609.
[4] Y. Chen, L. Wu, and M. J. Zaki, "Reinforcement learning based graph-to-sequence model for natural question generation," in *Proc. Int. Conf. Learn. Representations*, 2020.
[5] C. Liu, K. Liu, S. He, Z. Nie, and J. Zhao, "Generating questions for knowledge bases via incorporating diversified contexts and answer-aware loss," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2019, pp. 2431–2441.
[6] Y. Chen, L. Wu, and M. J. Zaki, "Toward subgraph guided knowledge graph question generation with graph neural networks," 2020, *arXiv: 2004.06015*.
[7] M. Gaur, K. Gunaratna, V. Srinivasan, and H. Jin, "ISEEQ: Information seeking question generation using dynamic meta-information retrieval and knowledge graphs," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 10 672–10 680.
[8] X. Yao, G. Bouma, Y. Zhang, P. Piwek, and K. Boyer, "Semantics-based question generation and implementation," *Dialogue Discourse*, vol. 3, pp. 11–42, 2012.
[9] V. Rus, B. Wyse, P. Piwek, M. Lintean, S. Stoyanchev, and C. Moldovan, "Question generation shared task and evaluation challenge – status report," in *Proc. 13th Eur. Workshop Natural Lang. Gener.*, 2011, pp. 318–320.
[10] Y. Zhao, X. Ni, Y. Ding, and Q. Ke, "Paragraph-level neural question generation with maxout pointer and gated self-attention networks," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 3901–3910.

[11] S. Reddy, D. Raghu, M. M. Khapra, and S. Joshi, "Generating natural language question-answer pairs from a knowledge graph using a RNN based question generation model," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 376–385.

[12] C. Dong et al., "A survey of natural language generation," *ACM Comput. Surv.*, vol. 55, 2022, Art. no. 173.

[13] J. Li, Y. Gao, L. Bing, I. King, and M. R. Lyu, "Improving question generation with to the point context," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2019, pp. 3216–3226.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[15] M. Heilman and N. A. Smith, "Good question! Statistical ranking for question generation," in *Proc. Conf. North Amer. Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2010, pp. 609–617.

[16] I. Labutov, S. Basu, and L. Vanderwende, "Deep questions without deep understanding," in *Proc. Conf. Assoc. Comput. Linguistics*, 2015, pp. 889–898.

[17] X. Du, J. Shao, and C. Cardie, "Learning to ask: Neural question generation for reading comprehension," in *Proc. Conf. Assoc. Comput. Linguistics*, 2017, pp. 1342–1352.

[18] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, and M. Zhou, "Neural question generation from text: A preliminary study," in *Proc. Int. Conf. Natural Lang. Process. Chin. Comput.*, 2018, pp. 662–671.

[19] X. Sun, J. Liu, Y. Lyu, W. He, Y. Ma, and S. Wang, "Answer-focused and position-aware neural question generation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 3930–3939.

[20] X. Ma, Q. Zhu, Y. Zhou, and X. Li, "Improving question generation with sentence-level semantic matching and answer position inferring," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 8464–8471.

[21] B. Liu, H. Wei, D. Niu, H. Chen, and Y. He, "Asking questions the human way: Scalable question-answer generation from text corpus," in *Proc. Web Conf.*, 2020, pp. 2032–2043.

[22] L. Pan, Y. Xie, Y. Feng, T.-S. Chua, and M.-Y. Kan, "Semantic graphs for generating deep questions," in *Proc. Conf. Assoc. Comput. Linguistics*, 2020, pp. 1463–1475.

[23] X. Jia, W. Zhou, X. Sun, and Y. Wu, "EQG-RACE: Examination-type question generation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 13 143–13 151.

[24] S. Wang, Z. Wei, Z. Fan, Y. Liu, and X. Huang, "A multi-agent communication framework for question-worthy phrase extraction and question generation," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 7168–7175.

[25] W. Zhou, M. Zhang, and Y. Wu, "Multi-task learning with language modeling for question generation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2019, pp. 3394–3399.

[26] B. Liu et al., "Learning to generate questions by learning what not to generate," in *Proc. World Wide Web Conf.*, 2019, pp. 1106–1118.

[27] X. Jia, W. Zhou, X. Sun, and Y. Wu, "How to ask good questions? Try to leverage paraphrases," in *Proc. Conf. Assoc. Comput. Linguistics*, 2020, pp. 6130–6140.

[28] D. Seyler, M. Yahya, and K. Berberich, "Generating quiz questions from knowledge graphs," in *Proc. World Wide Web Conf.*, 2015, pp. 113–114.

[29] D. Seyler, M. Yahya, and K. Berberich, "Knowledge questions from knowledge graphs," in *Proc. ACM SIGIR Int. Conf. Theory Inf. Retrieval*, 2017, pp. 11–18.

[30] L. Song and L. Zhao, "Question generation from a knowledge base with web exploration," 2016, *arXiv:1610.03807*.

[31] I. V. Serban et al., "Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus," in *Proc. Conf. Assoc. Comput. Linguistics*, 2016, pp. 588–598.

[32] H. Wang, X. Zhang, and H. Wang, "A neural question generation system based on knowledge base," in *Proc. Int. Conf. Natural Lang. Process. Chin. Comput.*, 2018, pp. 133–142.

[33] S. Bi, X. Cheng, Y.-F. Li, Y. Wang, and G. Qi, "Knowledge-enriched, type-constrained and grammar-guided question generation over knowledge bases," in *Proc. Int. Conf. Comput. Linguistics*, 2020, pp. 2776–2786.

[34] V. Kumar, Y. Hua, G. Ramakrishnan, G. Qi, L. Gao, and Y.-F. Li, "Difficulty-controllable multi-hop question generation from knowledge graphs," in *Proc. Int. Semantic Web Conf.*, 2019, pp. 382–398.

[35] P. Ke et al., "JointGT: Graph-text joint representation learning for text generation from knowledge graphs," in *Proc. Findings Assoc. Comput. Linguistics*, 2021, pp. 2526–2538.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997.

[37] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[38] Y. Deng et al., "Knowledge as a bridge: Improving cross-domain answer selection with external knowledge," in *Proc. Int. Conf. Comput. Linguistics*, 2018, pp. 3295–3305.

[39] F. W. Levi, *Finite Geometrical Systems*. Calcutta, West Bengal, India: The University of Calcutta, 1942.

[40] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. Conf. Assoc. Comput. Linguistics*, 2017, pp. 1073–1083.

[41] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2016, pp. 2383–2392.

[42] W.-T. Yih, M. Richardson, C. Meek, M.-W. Chang, and J. Suh, "The value of semantic parse labeling for knowledge base question answering," in *Proc. Conf. Assoc. Comput. Linguistics*, 2016, pp. 201–206.

[43] A. Talmor and J. Berant, "The web as a knowledge-base for answering complex questions," in *Proc. Conf. North Amer. Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 641–651.

[44] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. Conf. Assoc. Comput. Linguistics*, 2002, pp. 311–318.

[45] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Workshop Text Summarization Branches Out*, 2004, pp. 74–81.

[46] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proc. ACL Workshop Intrinsic Extrinsic Eval. Measures Mach. Transl. Summarization*, 2005, pp. 65–72.

[47] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[48] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[49] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. Conf. Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.

[50] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 1247–1250.

[51] Y. Deng et al., "Multi-task learning with multi-view attention for answer selection and knowledge base question answering," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 6318–6325.

[52] 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.

[53] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[54] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[55] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.

[56] X. Han et al., "OpenKE: An open toolkit for knowledge embedding," in *Proc. Conf. Empirical Methods Natural Lang. Process.: Syst. Demonstrations*, 2018, pp. 139–144.

[57] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2575–2583.

[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, Y. Bengio and Y. LeCun, Eds., 2015.

[59] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 1171–1179.

[60] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A diversity-promoting objective function for neural conversation models," in *Proc. Conf. North Amer. Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 110–119.

[61] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston, "Neural text generation with unlikelihood training," in *Proc. Int. Conf. Learn. Representations*, 2020.
[62] Y. Su, T. Lan, Y. Wang, D. Yogatama, L. Kong, and N. Collier, "A contrastive framework for neural text generation," 2022, *arXiv:2202.06417*.
[63] F. Stahlberg and S. Kumar, "Seq2Edits: Sequence transduction using span-level edit operations," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2020, pp. 5147–5159.
[64] M. Tarnavskyi, A. Chernodub, and K. Omelianchuk, "Ensembling and knowledge distilling of large sequence taggers for grammatical error correction," in *Proc. Conf. Assoc. Comput. Linguistics*, 2022, pp. 3842–3852.

Chenhe Dong received the BS degree from Northeastern University. He is currently working toward the MS degree with the School of Intelligent Systems Engineering, Sun Yat-Sen University. His research interests include deep learning and natural language generation.

Ying Shen received the master's (erasmus mundus) degree in natural language processing from the University of Franche-Comté, France and the University of Wolverhampton, England, and the PhD degree from the University of Paris Ouest Nanterre La Défense, France, specialized in computer science. She is now an associate professor with the School of Intelligent Systems Engineering, Sun Yat-Sen University. Her research interests include natural language processing and deep learning.

Shiyang Lin received the BS degree from the Nanjing University of Posts and Telecommunications. He is currently working toward the MS degree with the Sun Yat-sen University. His research interests include natural language processing, deep learning, and graph representation learning.

Zhenzhou Lin received the BS degree from the Sun Yat-Sen University of Automation. He is currently working toward the MS degree with the School of Intelligent Systems Engineering, Sun Yat-Sen University. His research interests include natural language processing, natural language generation, and deep learning.

Yang Deng received the BS degree from the Beijing University of Posts and Telecommunications, the MS degree from Peking University, and the PhD degree from the Chinese University of Hong Kong. He is currently a postdoctoral research fellow with NExT++, School of Computing, National University of Singapore. His research interests include natural language processing, information retrieval, and deep learning.