Unsupervised Multi-hop Question Answering by Question Generation

Liangming Pan[1] **Wenhu Chen**[2] **Wenhan Xiong**[2]
Min-Yen Kan[1] **William Yang Wang**[2]
[1]School of Computing, National University of Singapore, Singapore
[2]University of California, Santa Barbara, CA, USA
e0272310@u.nus.edu
{wenhuchen, xwhan, william}@cs.ucsb.edu
kanmy@comp.nus.edu.sg

Abstract

Obtaining training data for multi-hop question answering (QA) is time-consuming and resource-intensive. We explore the possibility to train a well-performed multi-hop QA model without referencing any human-labeled multi-hop question-answer pairs, *i.e.*, *unsupervised* multi-hop QA. We propose MQA-QG, an unsupervised framework that can generate human-like multi-hop training data from both homogeneous and heterogeneous data sources. MQA-QG generates questions by first selecting/generating relevant information from each data source and then integrating the multiple information to form a multi-hop question. Using only generated training data, we can train a competent multi-hop QA which achieves 61% and 83% of the supervised learning performance for the HybridQA and the HotpotQA dataset, respectively. We also show that pretraining the QA system with the generated data would greatly reduce the demand for human-annotated training data. Our codes are publicly available at https://github.com/teacherpeterpan/Unsupervised-Multi-hop-QA.

1 Introduction

Extractive Question Answering (EQA) is the task of answering questions by selecting a span from the given context document. Works on EQA can be divided into the single-hop (Rajpurkar et al., 2016, 2018; Kwiatkowski et al., 2019) and multi-hop cases (Yang et al., 2018; Welbl et al., 2018; Perez et al., 2020). Unlike single-hop QA, which assumes the question can be answered with a single sentence or document, multi-hop QA requires combining disjoint pieces of evidence to answer a question. Though different well-designed neural models (Qiu et al., 2019; Fang et al., 2020) have achieved near-human performance on the multi-hop QA datasets (Welbl et al., 2018; Yang et al., 2018), these approaches rely heavily on the availability of large-scale human annotation. Compared with single-hop QA datasets (Rajpurkar et al., 2016), annotating multi-hop QA datasets is significantly more costly and time-consuming because a human worker needs to read multiple data sources in order to propose a reasonable question.

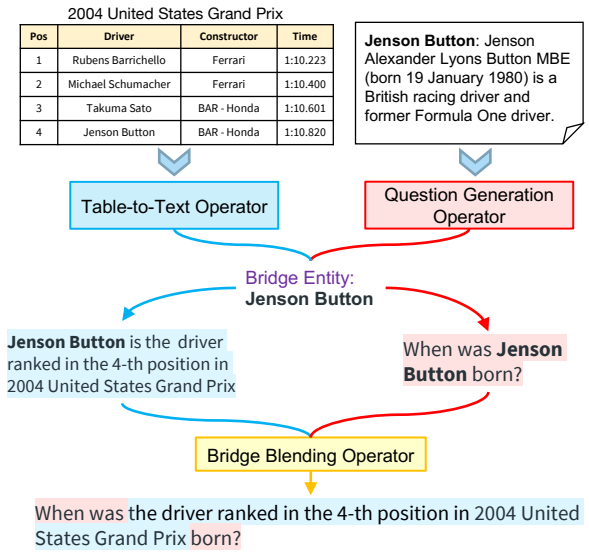


Figure 1: An overview of our approach for generating bridge-type multi-hop questions from table and text. The full set of supported input types and question types are described in Section 3.2.

To address the above problem, we pursue a more realistic setting, *i.e.*, *unsupervised* multi-hop QA, in which we assume no human-labeled multi-hop question is available for training, and we explore the possibility of *generating* human-like multi-hop question–answer pairs to train the QA model. We study multi-hop QA for both the homogeneous case where relevant evidence is in the textual forms (Yang et al., 2018) and the heterogeneous case where evidence is manifest in both tabular and textual forms (Chen et al., 2020b). Though successful attempts have been made to generate single-hop question–answer pairs by style transfer (Lewis et al., 2019) or linguistic rules (Li et al.,

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5866–5880
June 6–11, 2021. ©2021 Association for Computational Linguistics

2020), these methods are not directly applicable to the multi-hop setting as: 1) they cannot integrate information from multiple data sources, and 2) they only handle free-form text but not heterogeneous sources as input contexts.

We propose Multi-Hop Question Generator (MQA-QG), a simple yet general framework that decomposes the generation of a multi-hop question into two steps: 1) selecting relevant information from each data source, 2) integrating the multiple information to form a question. Specifically, the model first defines a set of basic *operators* to retrieve / generate relevant information from each input source or to aggregate different information. Afterwards, we define six *reasoning graphs*. Each corresponds to one type of multi-hop question and is formulated as a computation graph built upon the operators. We generate multi-hop question–answer pairs by executing the reasoning graph. Figure 1 shows an example of generating a *table-to-text question*: a) Given the inputs of (table, text), the $FindBridge$ operator locates a bridge entity that connects the contents between table and text. b) We generate a simple, single-hop question for the bridge entity from the text ($QGwithEnt$ operator) and generate a sentence describing the bridge entity from the table ($DescribeEnt$ operator). c) The $BridgeBlend$ operator blends the two generated contents to obtain the multi-hop question.

We evaluate our method on two multi-hop QA datasets: HotpotQA (Yang et al., 2018) and HybridQA (Chen et al., 2020b). Questions in HotpotQA reason over multiple texts (homogeneous data), while questions in HybridQA reason over both table and text (heterogeneous data). The experiments show that MQA-QG can generate high-quality multi-hop questions for both datasets. Without using any human-labeled examples, the generated questions alone can be used to train a surprisingly well QA model, reaching 61% and 83% of the F1 score achieved by the fully-supervised setting on the HybridQA and HotpotQA dataset, respectively. We also find that our method can be used in a few-shot learning setting. For example, after pretraining the QA model with our generated data, we can obtain 64.6 F1 with only 50 labeled examples in HotpotQA, compared with 21.6 F1 without the warm-up training.

In summary, our contributions are:

• To the best of our knowledge, this is the first work to investigate unsupervised multi-hop QA.

• We propose MQA-QG, a novel framework to generate high-quality training data without the need to see any human-annotated multi-hop question.

• We show that the generated training data can greatly benefit the multi-hop QA system in both unsupervised and few-shot learning settings.

2 Related Work

Unsupervised Question Answering. To reduce the reliance on expensive data annotation, *Unsupervised / Zero-Shot QA* has been proposed to train question answering models without any human-labeled training data. Lewis et al. (2019) proposed the first unsupervised QA model which generates synthetic (context, question, answer) triples to train the QA model using unsupervised machine translation. However, the generated questions are unlike human-written questions and tend to have a lot of lexical overlaps with the context. To address this, followup works utilized the Wikipedia cited documents (Li et al., 2020), predefined templates (Fabbri et al., 2020), or pretrained language model (Puri et al., 2020) to produce more natural questions resembling the human-annotated ones.

However, all the existing studies are focused on the SQuAD (Rajpurkar et al., 2016) dataset to answer *single-hop* and *text-only* questions. These methods do not generalize to multi-hop QA because they lack integrating and reasoning over disjoint pieces of evidence. Furthermore, they are restricted to text-based QA without considering structured or semi-structured data sources such as KB and Table. In contrast, we propose the first framework for unsupervised *multi-hop QA*, which can reason over disjoint structured or unstructured data to answer complex questions.

Multi-hop Question Generation. Question Generation (QG) aims to automatically generate questions from textual inputs (Pan et al., 2019). Early work of Question Generation (QG) relied on syntax rules or templates to transform a piece of given text to questions (Heilman, 2011; Chali and Hasan, 2012). With the proliferation of deep learning, QG evolved to use supervised neural models, where most systems were trained to generate questions from *(passage, answer)* pairs in the SQuAD dataset (Du et al., 2017; Zhao et al., 2018; Kim et al., 2019).

With the advent of pretraining language models (Dong et al., 2019), the challenge of generating single-hop questions similar to SQuAD

Group	Operator	Inputs → Outputs	Description
Selection	$FindBridge$	(Table $\mathcal{T}$, Text $\mathcal{D}$) or Texts ($\mathcal{D}_1$, $\mathcal{D}_2$) → Bridge Entities $\mathcal{E}^B$	Select an entity $\mathcal{E}^B$ that links the two input texts $\mathcal{D}_1$ and $\mathcal{D}_2$ (or links the table $\mathcal{T}$ and the text $\mathcal{D}$)
	$FindComEnt$	Text $\mathcal{D}$ → Comparative Entities $\mathcal{E}^C$	Extract potential comparative entities from the input text (location, datetime, number, etc.).
Generation	$QGwithAns$	(Text $\mathcal{D}$, Answer $\mathcal{A}$) → Question $\mathcal{Q}$	Generate a single-hop question $\mathcal{Q}$ with answer $\mathcal{A}$ from the input text $\mathcal{D}$
	$QGwithEnt$	(Text $\mathcal{D}$, Entity $\mathcal{E}$) → Question $\mathcal{Q}$	Generate a single-hop question $\mathcal{Q}$ that contains the given entity $\mathcal{E}$ from the input text $\mathcal{D}$
	$DescribeEnt$	(Table $\mathcal{T}$, Entity $\mathcal{E}$) → Sentence $\mathcal{S}$	Generate a sentence $\mathcal{S}$ that describes the given entity $\mathcal{E}$ based on the information of the table $\mathcal{T}$
	$QuesToSent$	Question $\mathcal{Q}$ → Sentence $\mathcal{S}$	Convert a question $\mathcal{Q}$ into its declarative form $\mathcal{S}$
Fusion	$BridgeBlend$	(Question $\mathcal{Q}$, Sentence $\mathcal{S}$, Bridge $\mathcal{E}^B$) → Bridge-type multi-hop question $\mathcal{Q}^B$	Generate a bridge-type multi-hop question $\mathcal{Q}^B$ by fusing the single-hop question Q and the sentence S given the entity $\mathcal{E}^B$ as the bridge
	$CompBlend$	(Question $\mathcal{Q}_1$, Question $\mathcal{Q}_2$) → Comparative multi-hop question $\mathcal{Q}^C$	Generate a comparison-type multi-hop question $\mathcal{Q}^C$ by fusing two single-hop questions

Table 1: The 8 basic operators for MQA-QG, categorized into 3 groups. **Selection**: retrieve relevant information from contexts. **Generation**: generate information from a single context. **Fusion**: fuse retrieved/generated information to construct multi-hop questions. Each operator is defined as a function mapping $f(X) \rightarrow Y$.

have largely been addressed. QG research has started to generate more complex questions that require deep comprehension and multi-hop reasoning (Tuan et al., 2020; Pan et al., 2020; Xie et al., 2020; Yu et al., 2020). For example, Tuan et al. (2020) proposed a multi-state attention mechanism to mimic the multi-hop reasoning process. Pan et al. (2020) parsed the input passage as a semantic graph to facilitate the reasoning over different entities. However, these supervised methods require large amounts of human-written multi-hop questions as training data. Instead, we propose the first *unsupervised* QG system to generate multi-hop questions without the need to access those annotated data.

3 Methodology

The setup of Multi-hop QA is as follows. Given a question q and a set of input contexts $\mathcal{C} = \{C_1, \cdots, C_n\}$, where each context C_i can be a passage, table, image, etc., the QA model $p_\theta(a|q, \mathcal{C})$ predicts the answer a for the question q by integrating and reasoning over information from $\mathcal{C}$.

In this paper, we consider two-hop questions and denote the required contexts as C_i and C_j. Formally, each time our model takes as inputs $\langle C_i, C_j \rangle$ to generate a set of (q, a) pairs. We focus on two modalities: the heterogeneous case where C_i, C_j are table and text and the homogeneous case where C_i, C_j are both texts. However, the design of our framework is flexible enough to generalize to multi-hop QA for other modalities.

Our model MQA-QG consists of three components: *operators*, *reasoning graphs*, and *question filtration*. Operators are atomic operations implemented by rules or off-the-shelf pretrained models to retrieve, generate, or fuse relevant information from input contexts (C_i, C_j). Different reasoning graphs define different types of reasoning chains for multi-hop QA with the operators as building blocks. Training (q, a) pairs are generated by executing the reasoning graphs. Question filtration removes irrelevant and unnatural (q, a) pairs to give the final training set $\mathcal{D}$ for multi-hop QA.

3.1 Operators

In Table 1, we define eight basic operators and divide them into three types: 1) *selection*: retrieve relevant information from a single context, 2) *generation*: generate information from a single context, and 3) *fusion*: fuse multiple retrieved/generated information to construct multi-hop questions.

• ***FindBridge*:** Most multi-hop questions rely on the entities that connect different input contexts, *i.e.*, *bridge entities*, to integrate multiple pieces of information (Xiong et al., 2019). *FindBridge* takes two contexts (C_i, C_j) as inputs, and extracts the entities that appear in both C_i and C_j as bridge entities. For example, in Figure 1, we extract "Jenson Button" as the bridge entity.

• ***FindComEnt*:** When generating comparative-type multi-hop questions, we need to decide what property to compare for the bridge entity. *FindComEnt* extracts potential comparative properties

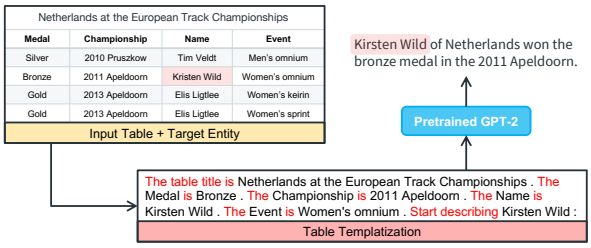


Figure 2: The implementation of *DescribeEnt* operator.

Figure 3: An example of the *BridgeBlend* operator.

from the input text. We extract entities with NER types $Nationality$, $Location$, $DateTime$, and $Number$ from the input text as comparative properties (*cf*, "Comparison" in Figure 4).

• ***QGwithAns*, *QGwithEnt*:** These two operators generate simple, single-hop questions from a single context, which are subsequently used to compose multi-hop questions. We use the pretrained Google T5 model (Raffel et al., 2019) fine-tuned on SQuAD to implement these two operators. Given the SQuAD training set of context-question-answer triples $\mathcal{D} = \{(c, q, a)\}$, we jointly fine-tune the model on two tasks. 1) *QGwithAns* aims to generate a question q with a as the answer, given (c, a) as inputs. 2) *QGwithEnt* aims to generate a question q that contains a specific entity e, given (c, e) as inputs. The evaluation of this T5-based model can be found in Appendix A.1.

• ***DescribeEnt*:** Given a table T and a target entity e in the table, the *DescribeEnt* operator generates a sentence that describes the entity e based on the information in the table T. We implement this using the GPT-TabGen model (Chen et al., 2020a) shown in Figure 2. The model first uses template to flatten the table T into a document P_T and then feed P_T to the pre-trained GPT-2 model (Radford et al., 2019) to generate the output sentence Y. To avoid irrelevant information in P_T, we apply a template that only describes the row where the target entity locates. We then finetune the model on the ToTTo dataset (Parikh et al., 2020), a large-scale dataset of controlled table-to-text generation, by maximizing the likelihood of $p(Y|P_T; \beta)$, with β denoting the model parameters. The implementation details and the model evaluation are in Appendix A.1.

• ***QuesToSent*:** This operator convert a question q into its declarative form s by applying the linguistic rules defined in Demszky et al. (2018).

• ***BridgeBlend*:** The operator composes a bridge-type multi-hop question based on: 1) a bridge entity e, 2) a single-hop question q that contains e, and 3) a sentence s that describes e. As exemplified in Figure 3, we implement this by applying a simple yet effective rule that replaces the bridge entity e in q with "the [MASK] that s" and employ the pretrained BERT-Large (Devlin et al., 2019) to fill in the [MASK] word.

• ***CompBlend*:** This operator composes a comparison-type multi-hop question based on two single-hop questions q_1 and q_2. The two questions ask about the same comparative property p for two different entities e_1 and e_2. We form the multi-hop question by filling p, e_1, and e_2 into pre-defined templates (Further details in Appendix A.2).

3.2 Reasoning Graphs

Based on the basic operators, we define six types of *reasoning graphs* to generate questions with different types. Each reasoning graph is represented as a directed acyclic graph (DAG) $\mathcal{G}$, where each node in $\mathcal{G}$ corresponds to an operator. A node s_i is connected by an incoming edge $\langle s_j, s_i \rangle$ if the output of s_j is given as an input to s_i.

As shown in Figure 4, *Table-Only* and *Text-Only* represent single-hop questions from table and text, respectively. The remaining reasoning graphs define four types of multi-hop questions. 1) *Table-to-Text*: bridge-type question between table and text, where the answer comes from the text. 2) *Text-to-Table*: bridge-type question between table and text, where the answer comes from the table. 3) *Text-to-Text*: bridge-type question between two texts. 4) *Comparison*: comparison-type question based on two passages. These four reasoning chains can cover a large portion of questions in existing multi-hop QA datasets, such as HotpotQA and HybridQA. We generate QA pairs by executing each reasoning graph. Our framework can easily extend to other modalities and reasoning chains by defining new operators and reasoning graphs.

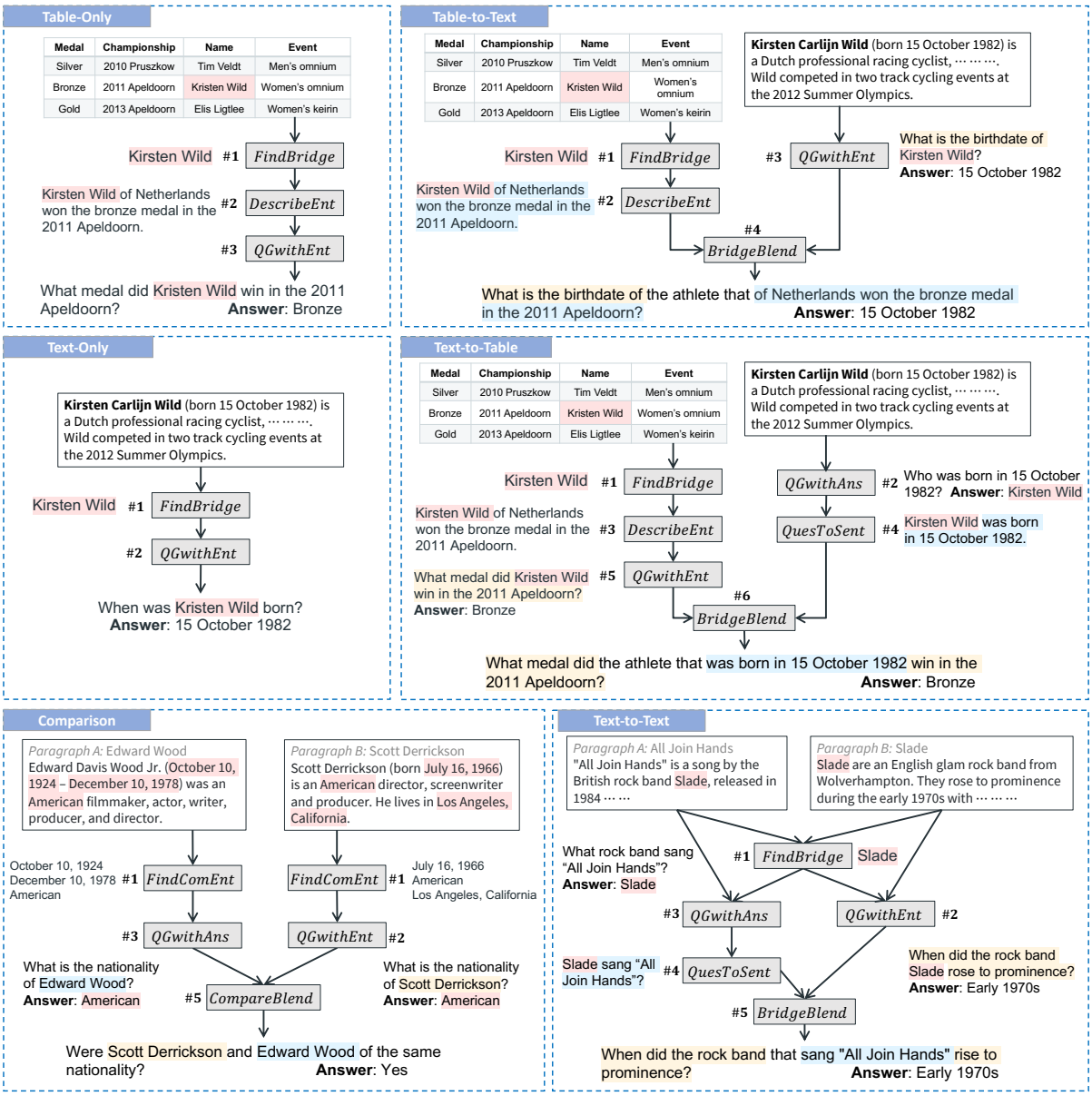


Figure 4: The 6 types of reasoning graphs for MQA-QG. Each graph is represented as a DAG of operators.

3.3 Question Filtration

Finally, we employ two methods to refine the quality of generated QA pairs. 1) **Filtration**. We use a pretrained GPT-2 model to filter out those questions that are disfluent or unnatural. The top N samples with the lowest perplexity scores are selected as the generated dataset to train the multi-hop QA model. 2) **Paraphrasing**. We train a question paraphrasing model based on the BART model (Lewis et al., 2020) to paraphrase each generated question. Our experiments show that filtration brings noticeable improvements to the QA model. However, we show in Section 4.5 that paraphrasing produces more human-like questions but introduces the semantic drift problem that harms the QA performance.

4 Experiments

We evaluate our framework on two multi-hop QA datasets: HotpotQA (Yang et al., 2018) and HybridQA (Chen et al., 2020b). HotpotQA focuses on multi-hop QA over homogeneous inputs, while HybridQA deals with multi-hop QA over heterogeneous information. HotpotQA contains ∼100K crowd-sourced multi-hop questions, where each question requires reasoning over two supporting Wikipedia documents to infer the answer. HybridQA contains ∼70K human-labeled multi-hop questions, where each question is aligned with a structured Wikipedia table and multiple passages linked with the entities in the table. The questions

Split	Train	Dev	Test	Total
HotpotQA				
Bridge	72,991	5,918	–	78,909 (81 %)
Comparison	17,456	1,487	–	18,943 (19 %)
Total	90,447	7,405	–	97,852
HybridQA				
In-Passage	35,215	2,025	2,045	39,285 (56 %)
In-Table	26,803	1,349	1,346	29,498 (43 %)
Compute	664	92	72	828 (1.1 %)
Total	62,682	3,466	3,463	69,611

Table 2: Basic statistics of HotpotQA and HybridQA.

are designed to aggregate both tabular information and text information, *i.e.*, lack of either form renders the question unanswerable.

Table 2 shows the statistics of these two datasets and Appendix B.1 gives their data examples. There are two types of multi-hop questions in HotpotQA: bridge-type (81%) and comparison-type (19%). For HybridQA, questions are divided by whether their answers come from the table (In-Table question, 56%) or from the passage (In-Passage question, 44%). Around 80% HybridQA questions requires bridge-type reasoning.

4.1 Unsupervised QA Results

Question Generation. In HybridQA, we extract its table–text corpus consisting of (T, D) input pairs, where T denotes the table and set of its linked passages D. We generate two multi-hop QA datasets $\mathcal{Q}_{tbl \to txt}$ and $\mathcal{Q}_{txt \to tbl}$ with MQA-QG by executing the "Table-to-Text" and "Text-to-Table" reasoning graphs for each (T, D), resulting in a total of 170K QA pairs. We then apply question filtration to obtain the training set $\mathcal{Q}_{hybrid}$ with 100K QA pairs. Similarly, for HotpotQA, we first generate $\mathcal{Q}_{bge}$ and $\mathcal{Q}_{com}$, which contains only the bridge-type questions and only the comparison-type questions, respectively. Afterward, we merge them and filter the questions to obtain the final training set $\mathcal{Q}_{hotpot}$ with 100K QA pairs. In Appendix B.2, we gives the statistics of all the generated datasets.

Question Answering For HybridQA, we use the HYBRIDER (Chen et al., 2020b) as the QA model, which breaks the QA into linking and reasoning to cope with heterogeneous information, achieving the best result in HybridQA. For HotpotQA, we use the SpanBERT (Joshi et al., 2020) since it achieved promising results on HotpotQA with reproducible codes. We use the standard Exact Match (EM) and F_1 metrics to measure the QA performance.

Baselines. We compare MQA-QG with both supervised and unsupervised baselines. For HybridQA, we first include the two supervised baselines *Table-Only* and *Passage-Only* in Chen et al. (2020b), which only rely on the tabular information or the textual information to find the answer. As we are the first to target unsupervised QA on HybridQA, there is no existing unsupervised baseline for direct comparison. Therefore, we construct a strong baseline *QDMR-to-Question* that generate questions from Question Decomposition Meaning Representation (QDMR) (Wolfson et al., 2020), a logical representation specially designed for multi-hop questions. We first generate QDMR expressions from the input (table, text) using pre-defined templates and then train a Seq2Seq model (Bahdanau et al., 2014) to translate QDMR into question. Details of this baseline are introduced in Appendix C. For HotpotQA, we introduce three unsupervised baselines. *SQuAD-Transfer* trains SpanBERT on SQuAD and then transfers it for multi-hop QA. *Bridge-Only* / *Comparison-Only* use only the bridge-type / comparison-type questions by MQA-QG to train the QA model.

Performance Comparison. Table 3 and Table 4 summarizes the QA performance on HybridQA and HotpotQA, respectively. For HybridQA, we use the reported performance of HYBRIDER as the supervised benchmark (S3) and apply the same model setting of HYBRIDER to train the unsupervised version, *i.e.*, using our generated QA pairs as the training data (U2 and U3). For HotpotQA, the original paper of SpanBERT only reported the results for the MRQA-2019 shared task (Fisch et al., 2019), which only includes the bridge-type questions in HotpotQA. Therefore, we retrain the SpanBERT on the full HotpotQA dataset to get the supervised benchmark (S4) and using the same model setting to train the unsupervised versions (U7 and U8).

Our unsupervised model MQA-QG attains 30.5 F_1 on the HybridQA test set and 68.6 F_1 on the HotpotQA dev set, outperforming all the unsupervised baselines (U1, U4, U5, U6) by large margins. Without using their human-annotated training data, the $F1$ gap to the fully-supervised version is only 19.5 and 14.2 for HybridQA and HotpotQA, respectively. In particular, the results of U2 and U3 even outperform the two weak supervised baselines (S1 and S2) in HybridQA. This demonstrates the effectiveness of MQA-QG in generating good multi-hop questions for training the QA model.

Model		In-Table	In-Passage	Total
		EM / F_1	EM / F_1	EM / F_1
Supervised	S1. Table-Only (Chen et al., 2020b)	14.7 / 19.1	2.4 / 4.5	8.4 / 7.1
	S2. Passage-Only (Chen et al., 2020b)	9.2 / 13.5	26.1 / 32.4	19.5 / 25.1
	S3. HYBRIDER (Chen et al., 2020b)	**51.2 / 58.6**	**39.6 / 46.4**	**42.9 / 50.0**
Unsupervised	U1. QDMR-to-Question	25.7 / 29.7	12.8 / 16.5	17.7 / 21.4
	U2. MQA-QG *-w/o* Filtration	33.0 / 37.1	18.6 / 23.4	23.8 / 28.2
	U3. MQA-QG	**36.2 / 40.6**	**19.8 / 25.0**	**25.7 / 30.5**

Table 3: Performance comparison between supervised models and unsupervised models on **HybridQA**.

Model		Bridge	Comparison	Total
		EM / F_1	EM / F_1	EM / F_1
Supervised	S4. SpanBERT (Joshi et al., 2020)	**68.2 / 83.5**	**74.2 / 80.3**	**69.4 / 82.8**
Unsupervised	U4. Bridge-Only	55.4 / 71.4	12.4 / 19.1	46.7 / 60.9
	U5. Comparison-Only	9.8 / 14.5	38.2 / 45.0	15.5 / 20.6
	U6. SQuAD-Transfer	54.6 / 69.7	25.3 / 35.2	48.7 / 62.8
	U7. MQA-QG *-w/o* Filtration	55.2 / 71.2	44.8 / 52.9	53.1 / 67.5
	U8. MQA-QG	**56.5 / 72.2**	**48.8 / 54.4**	**54.9 / 68.6**

Table 4: Performance comparison between supervised models and unsupervised models on **HotpotQA**.

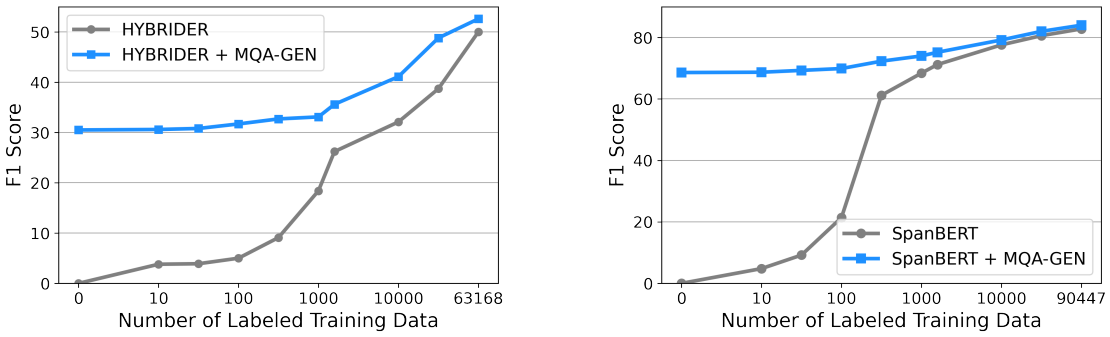
Setting	Components				Reasoning Types		Performance		
	Text	Table	Fusion	Filtration	Table→Text	Text→Table	In-Table	In-Passage	Total
							EM / F_1	EM / F_1	EM / F_1
A1	✓						12.4 / 14.9	2.7 / 4.3	6.4 / 8.3
A2		✓					19.4 / 23.3	3.4 / 5.5	9.6 / 12.3
A3	✓	✓					14.8 / 19.2	5.6 / 7.8	9.1 / 12.1
A4	✓	✓	✓		✓		11.1 / 15.2	17.3 / 21.9	14.9 / 19.4
A5	✓	✓	✓			✓	41.5 / 47.9	0.2 / 1.9	16.2 / 19.8
A6	✓	✓	✓		✓	✓	33.0 / 37.1	18.6 / 23.4	23.8 / 28.2
A7	✓	✓	✓	✓	✓	✓	36.2 / 40.6	19.8 / 25.0	25.7 / 30.5

Table 5: Ablations on the HybridQA development set. **Text/Table**: whether we utilize the information in the text/table. **Fusion**: whether we fuse the information from table and text. **Filtration**: whether we perform question filtration. **Reasoning Types**: which types of multi-hop questions are generated.

4.2 Ablation Study

To understand the impact of different components in MQA-QG, we perform an ablation study on the HybridQA development set. In Table 5, we compare our full model (A7) with six ablation settings by removing certain the model components (A1–A4) or by restricting the reasoning types (A5 and A6). We make three key observations.

Single-hop questions vs. multi-hop questions. A1 to A3 generates single-hop questions using the reasoning graph of *Text-Only* (A1), *Table-Only* (A2), or a union of them (A3). Afterwards, we use them to train the HYBRIDER model and test the multi-hop QA performance. In these cases, the model is trained to answer questions based on either table or text but lacking the ability to reason between table and text. As shown in Table 5, A1–A3 achieves a low performance of EM and F1, especially for In-Passage questions, showing that single-hop questions alone are insufficient to train a good multi-hop QA system. This reveals that learning to reason between different contexts is essential for multi-hop QA and justifies the necessity of generating multi-hop questions. However, for HotpotQA, we observe that the benefit of multi-hop questions is not as evident as in HybridQA: the SQuAD-Transfer (U6) achieves a relatively good F1 of 62.8. A potential reason is that the examples of HotpotQA contain reasoning shortcuts through which models can directly locate the answer by word-matching, without the need of multi-hop reasoning, as observed by Jiang and Bansal (2019).



(a) HybridQA (b) HotpotQA

Figure 5: The few-shot learning experiment. The figure shows the F1 score on the HybridQA (a) / HotpotQA (b) development set for progressively larger training dataset sizes. Note the difference in scales for the Y-axes.

Effect of reasoning types. When we train the model with only the Text-to-Table questions (A5), the model achieves 47.9 F1 for In-Table questions and nearly zero performance for In-Passage questions. However, training with only the Table-to-Text questions (A4) also benefits the In-Table questions (15.2 F1). We believe the reason is that the information in the text can also answer some In-Table questions. Using both reasoning types (A6), the model improves on average by 8.6 F1 compared with the models using a single reasoning type (A4, A5). This shows that it is beneficial to train the multi-hop QA model with diverse reasoning chains.

Effect of question filtration. Question filtration also helps to train a better QA model, leading to a +2.3 F1 for HybridQA and +1.1 F1 for HotpotQA. We find that the GPT-2 based model can filter out most ungrammatical questions but would keep valid yet unnatural questions such as "Where was the event that is held in 2016 held?".

4.3 Few-shot Multi-hop QA

We then explore MQA-QG's effectiveness in the few-shot learning setting where only a few human-labeled (q, a) pairs are available. We first train the unsupervised QA model based on the training data generated by our best model. Then we fine-tune the model with limited human-labeled data. The blue line in Figure 5(a) and Figure 5(b) shows the F1 scores with different numbers of labeled training data for HybridQA and HotpotQA, respectively. We compare this with training the QA model directly on the human-labeled data without unsupervised QA pretraining (grey lines in Figure 5).

With progressively larger training dataset sizes, our model performs consistently better than the model without unsupervised pretraining for both two datasets. The performance improvement is especially prominent in very data-poor regimes; for example, our approach achieves 69.3 F1 with only 100 labeled examples in HotpotQA, compared with 21.4 F1 without unsupervised pretraining (47.9 absolute gain). The results show pretraining QA with MQA-QG greatly reduce the demand for human-annotated data. It can be used to provide a "warm start" for online learning QA system in which training data are quite limited for a new domain.

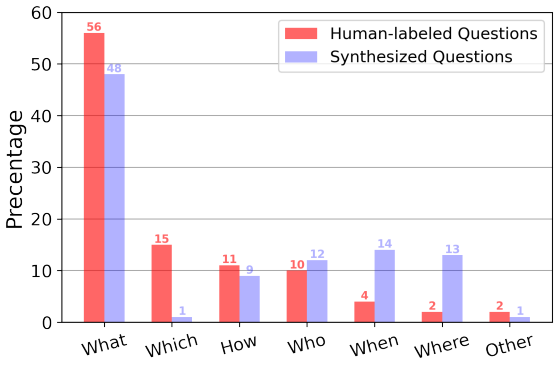


Figure 6: Question type distribution for our generated dataset and the human-labeled dataset for HybridQA.

4.4 Analysis of Generated Questions

Although the generated questions are used to optimize for downstream QA performance, it is still instructive to examine the output QA pairs to better understand our system's advantages and limitations. In Figure 6, we plot the question type distribution for both the human-labeled dataset and the generated data for HybridQA. We find that the two datasets have a similar question type distribution, where "What" questions constitute the major type.

Type	#	Generated Question	Answer
Table-to-Text	1	When did the one that won the Eurovision Song Contest in 1966 join Gals and Pals?	1963
	2	How many students attend the teams that played in the Dryden Township Conference?	1900
Text-to-Table	3	What album did the Oak Ridge Boys release in 1989?	American Dreams
	4	When was the name that is the name of the bridge that crosses Youngs Bay completed?	1921
Text-to-Text	5	Which Canadian cinematographer is best known for his work on Fargo?	Craig Wrobleski
	6	What is illegal in the country that is Bashar Hafez al - Assad 's father?	Cannabis
Comp.	7	Who was born first, Terry Southern or Neal Town Stephenson?	Terry Southern
	8	Are Beth Ditto and Mary Beth Patterson of the same nationality?	Yes

Table 6: Examples of multi-hop question–answers generated by MQA-QG, categorized by reasoning graphs. The two major error types are highlighted: red for *inaccurate reference* and blue for *redundancy*.

HybridQA	*In-Table*	*In-Passage*	*Total*
	EM / F_1	EM / F_1	EM / F_1
MQA-QG	**36.2 / 40.6**	**19.8 / 25.0**	**25.7 / 30.5**
+ Paraphrasing	37.7 / 43.5	12.1 / 15.8	21.8 / 26.2
HotpotQA	*Bridge*	*Comparison*	*Total*
	EM / F_1	EM / F_1	EM / F_1
MQA-QG	**56.5 / 72.2**	**48.8 / 54.4**	**54.9 / 68.6**
+ Paraphrasing	51.7 / 67.0	45.7 / 51.1	50.5 / 63.8

Table 7: Unsupervised multi-hop QA performance with/without question paraphrasing.

However, our model generates more "When" and "Where" questions but fewer "Which" questions. This is because the two reasoning graphs we apply for HybridQA are bridge-type questions while "Which" questions mostly compare.

Table 6 shows representative examples generated by our model. Most questions are fluent and exhibit encouraging language variety, such as Examples 2, 3, 5. Our model also shows almost no sign of semantic drift, meaning most of the questions are valid despite sometimes being unnatural. The two major deficiencies are *inaccurate references* (in red) and *redundancies* (in blue), shown in Examples 1, 4, 6. This can be addressed by incorporating minimal supervision to guide the fusion process; *i.e.*, more flexible paraphrasing in fusion.

4.5 Effects of Question Paraphrasing

As discussed in Section 3.3, to generated more natural-looking questions, we attempted to train a BART-based question paraphrasing model to paraphrase each generated question. We finetune the pretrained BART model on the Quora Question Paraphrasing dataset[1], which contains over 100,000 question pairs with equivalent semantic meaning.

The evaluation results are shown in Table 7. Surprisingly, we observe a performance drop for both the HybridQA and the HotpotQA dataset, with a 4.3 and 4.8 decrease in F1, respectively. We observe that paraphrasing indeed produces more fluent questions by rewriting the redundancy parts of the original questions into more concise expression. However, paraphrasing introduces the "semantic drift" problem, *i.e.*, the paraphrased question changes the semantic meaning of the original question. We believe this severally hurts the QA performance because it produces noisy samples with inconsistent question and answer. Therefore, we argue that in unsupervised multi-hop QA, semantic faithfulness is more important than fluency for the generated questions. This explains why we design hand-crafted reasoning graphs to ensure the semantic faithfulness. However, how to generate fluent human-like questions while keeping semantic faithfulness is an important future direction.

5 Conclusion and Future Works

In this work, we study unsupervised multi-hop QA and propose a novel framework MQA-QG to generate multi-hop questions via composing reasoning graphs built upon basic operators. The experiments show that our model can generate human-like questions that help to train a well-performing multi-hop QA model in both the unsupervised and the few-shot learning setting. Further work is required to include more flexible paraphrasing at the fusion stage. We can also design more reasoning graphs and operators to generate more complex questions and support more input modalities.

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative. UCSB authors are not supported by any of the above projects.

[1]https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Yllias Chali and Sadid A. Hasan. 2012. Towards automatic topical question generation. In *International Conference on Computational Linguistics (COLING)*, pages 475–492.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7929–7942.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. 2020b. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *CoRR*, abs/1809.02922.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 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 *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 13042–13054.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1342–1352.

Alexander R. Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Template-based question generation from retrieved sentences for improved unsupervised question answering. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4508–4513.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *The 2nd Workshop on Machine Reading for Question Answering (MRQA@EMNLP)*, pages 1–13.

Michael Heilman. 2011. Automatic factual question generation from text. *Language Technologies Institute School of Computer Science Carnegie Mellon University*, 195.

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2726–2736.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics (TACL)*, 8:64–77.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 6602–6609.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics (TACL)*, 7:452–466.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation (WMT@ACL)*, pages 228–231.

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 *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7871–7880.

Patrick S. H. Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4896–4910.

Zhongli Li, Wenhui Wang, Li Dong, Furu Wei, and Ke Xu. 2020. Harvesting and refining question-answer pairs for unsupervised QA. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6719–6728.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020. Asking questions the human way: Scalable question-answer generation from text corpus. In *International World Wide Web Conference (WWW)*.

Liangming Pan, Wenqiang Lei, Tat-Seng Chua, and Min-Yen Kan. 2019. Recent advances in neural question generation. *CoRR*, abs/1905.08949.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1463–1475.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. *CoRR*, abs/2004.14373.

Ethan Perez, Patrick S. H. Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880.

Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6140–6150.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 784–789.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2383–2392.

Luu Anh Tuan, Darsh J. Shah, and Regina Barzilay. 2020. Capturing greater context for question generation. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics (TACL)*, 6:287–302.

Tomer Wolfson, Mor Geva, Ankit Gupta, Yoav Goldberg, Matt Gardner, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics (TACL)*, 8:183–198.

Yuxi Xie, Liangming Pan, Dongzhe Wang, Min-Yen Kan, and Yansong Feng. 2020. Exploring question-specific rewards for generating deep questions. In *International Conference on Computational Linguistics (COLING)*, pages 2534–2546.

Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Hong Wang, Shiyu Chang, Murray Campbell, and William Yang Wang. 2019. Simple yet effective bridge reasoning for open-domain multi-hop question answering. In *The 2nd Workshop on Machine Reading for Question Answering (MRQA@EMNLP)*, pages 48–52.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2369–2380.

Jianxing Yu, Xiaojun Quan, Qinliang Su, and Jian Yin. 2020. Generating multi-hop reasoning questions to improve machine reading comprehension. In *International World Wide Web Conference (WWW)*, pages 281–291.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3901–3910.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *CCF International Conference of Natural Language Processing and Chinese Computing (NLPCC)*, pages 662–671.

A Implementation Details of Operators

In this section, we give the detailed implementation of four key operators, including *QGwithAns*, *QGwithEnt*, *DescribeEnt*, and *CompBlend*. We also separately evaluate their performance.

A.1 The *QGwithAns*, *QGwithEnt*, and *DescribeEnt* Operators

In summary, *QGwithAns*, *QGwithEnt* are T5-based question generation model trained on the SQuAD dataset, and *DescribeEnt* is a GPT-2 based model trained on the ToTTo dataset.

Implementation Details For the question generation model (the *QGwithAns* and *QGwithEnt* operators), we use the SQuAD data split from Zhou et al. (2017) to fine-tune the Google T5 model (Radford et al., 2019). We implement this based on the pretrained T5 model provided by https://github.com/patil-suraj/question_generation.

For the table-to-text generation model (the *DescribeEnt* operator), we adopt the GPT-TabGen model proposed in Chen et al. (2020b). The model first uses a template to flatten the input table T into a document P_T and then feed P_T to the pre-trained GPT-2 model to generate the output sentence Y. We fine-tune the model on the ToTTo dataset (Parikh et al., 2020), a large-scale dataset for controlled table-to-text generation. In ToTTo, given a Wikipedia table and a set of highlighted table cells, the objective is to produce a one-sentence description that best describes the highlighted cells. The original dataset contains 120,761 human-labeled training samples and 7,700 testing samples. To implement the *DescribeEnt* operator, we select the ToTTo samples that focuses on describing a given target entity e rather than the entire table, based on the following criteria: 1) the highlighted cells are in the same row and contains the target entity, 2) the description starts with the target entity. This gives us 15,135 training (T, e, s) triples and 1,194 testing triples, where T is the table, e is the target entity, and s is the target description.

Evaluation Setup We employ BLEU-4 (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), and ROUGE-L (Lin, 2004) to evaluate the performance of our implementation. For question generation, we compare the T5-based model with several state-of-the-art QG models, using their reported performance on the Zhou split of SQuAD. For the table-to-text generation, we compare GPT-TabGen with the Seq2Seq baseline with attention.

Evaluation Results Table 8 shows the evaluation results comparing against all baseline methods. For question generation, the Google-T5 model achieves a BLEU-4 of 21.32, outperforming NQG++, S2ga-mp-gsa, and CGC-QG by large margins. This is as expected since these three baselines are based on Seq2Seq and do not apply language model pretraining. Compared with the current state-of-the-art model UniLM, the Google-T5 model achieves comparable results, with slightly lower BLEU-4 but higher METEOR. For the table-to-text generation model, we find that GPT2-TabGen outperforms Seq2Seq with attention by 5.61 in BLEU-4. When switching to GPT-2-Medium as the pretraining model, the BLEU-4 further improves by 2.04. In our final model MQA-QG, we use the Google-T5 and the GPT2-Medium in the operators.

A.2 The *CompBlend* Operator

The inputs of the *CompBlend* operator are two single-hop questions Q_1 and Q_2 that ask about the same comparative property p; for example, Q_1 = "What is the nationality of Edward Wood?", Q_2 = "What is the nationality of Scott Derrickson", and p = "Nationality". We then identify the entity appearing in Q_1 and Q_2, denoted as e_1 and e_2, respectively. To form the multi-hop question, we fill in the comparing entities $e1$ and $e2$ into the corresponding templates that we define for the comparative property p. One of the resulting comparison question for the above example is "Are Edward Wood and Scott Derrickson of the same nationality?". This paper considers four comparative properties and defined a total number of 11 templates for them, summarized in Table 9.

B Dataset Details

In this section, we give further details for both the HotpotQA and the HybridQA dataset, as well as the generated datasets by our model MQA-QG.

B.1 HotpotQA and HybridQA Examples

Figure 7 gives data examples for the HotpotQA and the HybridQA dataset. The evidence used to compose the multi-hop question is highlighted, with different colors denoting information from different input contexts.

Operator	Model	BLEU-4	METEOR	ROUGE-L
QGwithAns & QGwithEnt	NQG++ (Zhou et al., 2017)	13.51	18.18	41.60
	S2ga-mp-gsa (Zhao et al., 2018)	15.82	19.67	44.24
	CGC-QG (Liu et al., 2020)	17.55	21.24	44.53
	Google-T5 (Radford et al., 2019)	21.32	**27.09**	43.60
	UniLM (Dong et al., 2019)	**23.75**	25.61	**52.04**
DescribeEnt	Seq2Seq Attention (Bahdanau et al., 2014)	28.31	27.61	56.63
	GPT2-TabGen (Chen et al., 2020b)	33.92	32.46	55.61
	GPT2-Medium (Chen et al., 2020b)	**35.94**	**33.74**	**57.44**

Table 8: Performance evaluation of the *QGwithAns*, *QGwithEnt*, and *DescribeEnt* operator for different models. The best performance is in bold. We adopt the Google-T5 and the GPT2-Medium in our model MQA-QG.

Comparative Property	#	**Question Template**	**Answer**
born, birthdate	1	Who was born first, $e1$ or $e2$?	$e1$ / $e2$
located, location	2	Are $e1$ and $e2$ located in the same place?	Yes / No
	3	Which one is located in $a1$, $e1$ or $e2$?	e_1
	4	Which one is located in a_2, $e1$ or $e2$?	e_2
	5	Are both $e1$ and $e2$ located in a_1?	Yes / No
nationality, nation, country	6	Are $e1$ and $e2$ of the same nationality?	Yes / No
	7	Which person is from $a1$, $e1$ or $e2$?	e_1
	8	Which person is from $a2$, $e1$ or $e2$?	e_2
live, live place, hometown	9	Are $e1$ and $e2$ living in the same place?	Yes / No
	10	Which person lives in $a1$, $e1$ or $e2$?	e_1
	11	Which person lives in $a2$, $e1$ or $e2$?	e_2

Table 9: The comparative properties and their corresponding question templates used in the *CompBlend* operator. $a1$ / $a2$ denotes the answer for the single-hop question Q_1 / Q_2.

B.2 Statistics of generated datasets

For baselines and ablation study, we generate different synthetic training sets by executing different reasoning graphs. For example, we generate two datasets with single-hop questions $\mathcal{Q}_{tbl}$ and $\mathcal{Q}_{txt}$ for HybridQA by executing the "Table-Only" and "Text-Only" reasoning graphs, respectively. They are applied to train the ablation model A1 and A2. Table 10 summarizes all the generated datasets generated by our model MQA-QG. The column "Train Model" denotes each dataset is used to train which model in our experiments.

C Baseline: QDMR-to-Question

In this section, we introduce our proposed *QDMR-to-Question*, a strong unsupervised multi-hop QA baseline for HybridQA. We propose this baseline to investigate whether we can generate multi-hop questions from logical forms and compare them with our model MQA-QG.

The QDMR Representation The basic idea of QDMR-to-Question is first to generate a structured meaning representation from the source contexts and then convert it into the multi-hop question. We use the Question Decomposition Meaning Representation (QDMR) (Wolfson et al., 2020), a logical representation specially designed for multi-hop questions as the intermediate question representation. QDMR expresses complex questions via atomic operations that can be executed in sequence to answer the original question. Each atomic operation either selects a set of entities, retrieves information about their attributes, or aggregates information over entities. For example, the QDMR for the question "How many states border Colorado?" is "1) Return Colorado; 2) Return border states of #1; 3) Return the number of #2". In contrast to semantic parsing, QDMR operations are expressed through natural language.

Based on the QDMR representation, Wolfson et al. (2020) crowdsourced BREAK, a large-

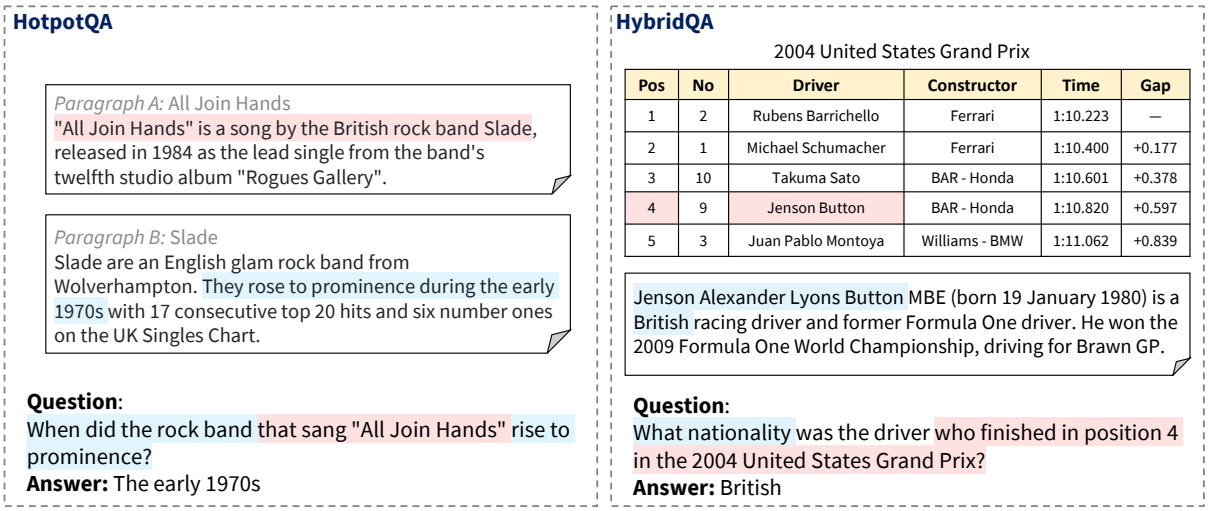


Figure 7: Data examples for the HotpotQA and the HybridQA dataset. Different colors (red and blue) highlight the evidences that are required to answer the multi-hop question from different sources.

	Dataset	Size	Description	Train Model
HotpotQA	$\mathcal{Q}_{bge}$	129,508	Bridge-type Questions	U4. Bridge-Only
	$\mathcal{Q}_{com}$	115,162	Comparison-type Questions	U5. Comparison-Only
	$\mathcal{Q}_{bge+com}$	244,220	$\mathcal{Q}_{bge} \cup \mathcal{Q}_{comp}$	U7. MQA-QG *-w/o* Filtration
	$\mathcal{Q}_{hotpot}$	100,000	$filtration(\mathcal{Q}_{bge+com})$	U8. MQA-QG
HybridQA	$\mathcal{Q}_{tbl}$	56,448	Table-Only Questions	A2
	$\mathcal{Q}_{txt}$	47,332	Text-Only Questions	A1
	$\mathcal{Q}_{txt+tbl}$	103,780	$\mathcal{Q}_{txt} \cup \mathcal{Q}_{tbl}$	A3
	$\mathcal{Q}_{txt \to tbl}$	56,448	Text-to-Table Questions	A5
	$\mathcal{Q}_{tbl \to txt}$	70,661	Table-to-Text Questions	A4
	$\mathcal{Q}_{txt \leftrightarrow tbl}$	127,109	$\mathcal{Q}_{txt \to tbl} \cup \mathcal{Q}_{tbl \to txt}$	U2. MQA-QG *-w/o* Filtration
	$\mathcal{Q}_{hybrid}$	100,000	$filtration(\mathcal{Q}_{txt \leftrightarrow tbl})$	U3. MQA-QG

Table 10: Basic statistics of all the generated datasets by our model MQA-QG.

scale question decomposition dataset consisting of 83,978 (QDMR, question) pairs over ten datasets.

Multi-hop Question Generation Given the table-text (T, D) as inputs, we first generate QDMR representations using two pre-defined templates that represent the *Table-to-Text* question and the *Text-to-Table* question, respectively. The templates with examples are given in Table 11. We generate QDMRs by randomly filling in the templates. Afterward, we translate the QDMR representation into a natural language question. To this end, we train a Seq2Seq model with attention (Bahdanau et al., 2014) on the BREAK dataset, where the input is a QDMR expression, and the target is the corresponding natural language form labeled by humans. We directly apply this Seq2Seq model trained on BREAK as the translator to transform our QDMR representations into multi-hop questions.

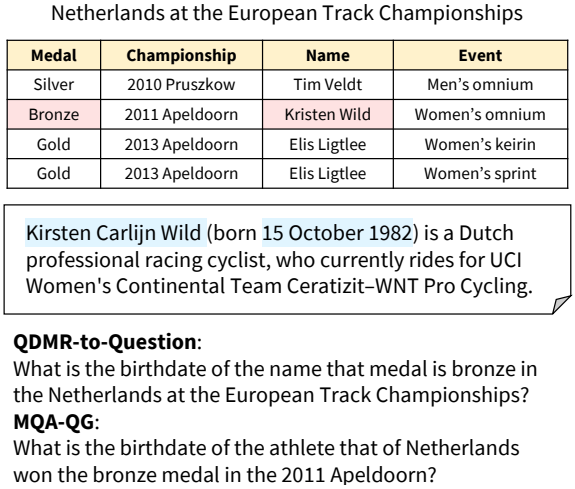


Figure 8: Examples of generated questions for the QDMR-to-Question model and the MQA-QG.

QDMR Template	Example	Question
Table-to-Text		
1) Return $\langle column\ A \rangle$	1) Return Driver	What is the birthdate of the driver that pos is 4 in the 2004 United States Grand Prix?
2) Return #1 that $\langle column\ B \rangle$ is $\langle row\ A \rangle$	2) Return #1 in Pos 4	
3) Return #2 in $\langle table\ title \rangle$	3) Return #2 in 2004 United States Grand Prix	
4) Return what is the $\langle text\ attribute \rangle$ of #3	4) Return what is the birthdate of #3	
Text-to-Table		
1) Return $\langle column\ A \rangle$	1) Return Driver	What is the pos of the driver in the 2004 United States that was born in 19 January, 1980?
2) Return #1 in $\langle table\ title \rangle$	2) Return #1 in 2004 United States Grand Prix	
3) Return #2 that $\langle predicate \rangle$ $\langle object \rangle$	3) Return #2 that born 19 January 1980	
4) Return what is the $\langle column\ B \rangle$ of #3	4) Return what is the Pos of #3	

Table 11: The QDMR templates used in the *QDMR-to-Question* model for HybridQA.

Evaluation and Discussions As shown in Section 4.1, QDMR-to-Question achieves 21.4 F1 on the HybridQA dataset, lower than our model MQA-QG by 9.1 F1. A typical example of generated question is shown in Figure 8. We believe that the main reason for the low performance of QDMR-to-Question is that it lacks a global understanding of the table semantics. Specifically, the model lacks an understanding of the table headers' semantic meaning and the semantic relationship between different headers because table columns and table rows are randomly selected to fill in the QDMR template. For example, in Figure 8, the model generates an unnatural expression "the name that medal is bronze" because it directly copies the table header "name" and "medal" without understanding them. Instead, as our MQA-QG applies the GPT2-based table-to-text model, which encodes the entire table as an embedding, it tends to produce more natural expressions that consider the general table semantics. For the same example, MQA-QG generates a better expression "the athlete that won the bronze medal".