Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps

Xanh Ho[♡♣], **Anh-Khoa Duong Nguyen**[♢], **Saku Sugawara**[♣], **Akiko Aizawa**[♡♣]
[♡] The Graduate University for Advanced Studies, Kanagawa, Japan
[♣] National Institute of Informatics, Tokyo, Japan
[♢] National Institute of Advanced Industrial Science and Technology, Tokyo, Japan
{xanh, saku, aizawa}@nii.ac.jp
khoa.duong@aist.go.jp

Abstract

A multi-hop question answering (QA) dataset aims to test reasoning and inference skills by requiring a model to read multiple paragraphs to answer a given question. However, current datasets do not provide a complete explanation for the reasoning process from the question to the answer. Further, previous studies revealed that many examples in existing multi-hop datasets do not require multi-hop reasoning to answer a question. In this study, we present a new multi-hop QA dataset, called 2WikiMultiHopQA, which uses structured and unstructured data. In our dataset, we introduce the evidence information containing a reasoning path for multi-hop questions. The evidence information has two benefits: (i) providing a comprehensive explanation for predictions and (ii) evaluating the reasoning skills of a model. We carefully design a pipeline and a set of templates when generating a question–answer pair that guarantees the multi-hop steps and the quality of the questions. We also exploit the structured format in Wikidata and use logical rules to create questions that are natural but still require multi-hop reasoning. Through experiments, we demonstrate that our dataset is challenging for multi-hop models and it ensures that multi-hop reasoning is required.

1 Introduction

Machine reading comprehension (MRC) aims at teaching machines to read and understand given text. Many current models (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019) have defeated humans on the performance of SQuAD (Rajpurkar et al., 2016; Rajpurkar et al., 2018), as shown on its leaderboard[1]. However, such performances do not indicate that these models can completely understand the text. Specifically, using an adversarial method, Jia and Liang (2017) demonstrated that the current models do not precisely understand natural language. Moreover, Sugawara et al. (2018) demonstrated that many datasets contain a considerable number of easy instances that can be answered based on the first few words of the questions.

Multi-hop MRC datasets require a model to read and perform multi-hop reasoning over multiple paragraphs to answer a question. Currently, there are four multi-hop datasets over textual data: ComplexWebQuestions (Talmor and Berant, 2018), QAngaroo (Welbl et al., 2018), HotpotQA (Yang et al., 2018), and $\mathrm{R}^4\mathrm{C}$ (Inoue et al., 2020). The first two datasets were created by incorporating the documents (from Web or Wikipedia) with a knowledge base (KB). Owing to their building procedures, these datasets have no information to explain the predicted answers. Meanwhile, the other two datasets were created mainly based on crowdsourcing. In HotpotQA, the authors introduced the sentence-level supporting facts (SFs) information that are used to explain the predicted answers. However, as discussed in Inoue et al. (2020), the task of classifying sentence-level SFs is a binary classification task that is incapable of evaluating the reasoning and inference skills of the model. Further, data analyses (Chen and Durrett, 2019; Min et al., 2019) revealed that many examples in HotpotQA do not require multi-hop reasoning to solve.

Recently, to evaluate the internal reasoning of the reading comprehension system, Inoue et al. (2020) proposed a new dataset $\mathrm{R}^4\mathrm{C}$ that requires systems to provide an answer and derivations. A derivation

[1] https://rajpurkar.github.io/SQuAD-explorer/

Proceedings of the 28th International Conference on Computational Linguistics, pages 6609–6625
Barcelona, Spain (Online), December 8-13, 2020

is a semi-structured natural language form that is used to explain the answers. R^4C is created based on HotpotQA and has 4,588 questions. However, the small size of the dataset implies that the dataset cannot be used as a multi-hop dataset with a comprehensive explanation for training end-to-end systems.

In this study, we create a large and high quality multi-hop dataset 2WikiMultiHopQA[2] with a comprehensive explanation by combining structured and unstructured data. To enhance the explanation and evaluation process when answering a multi-hop question on Wikipedia articles, we introduce new information in each sample, namely *evidence* that contains comprehensive and concise information to explain the predictions. Evidence information in our dataset is a set of triples, where each triple is a structured data *(subject entity, property, object entity)* obtained from the Wikidata (see Figure 1 for an example).

Paragraph A: John Cecil, 7th Earl of Exeter

[1] John Cecil, 7th Earl of Exeter (c. 1700 – 1722) was an English peer and member of the House of Lords, styled Lord Burghley from 1721 to 1722. [2] He inherited the earldom in 1721. [3] *His parents were John Cecil, 6th Earl of Exeter, and Elizabeth Brownlow, daughter of Sir John Brownlow, 3rd Baronet.*

Paragraph B: John Cecil, 6th Earl of Exeter

[4] John Cecil, 6th Earl of Exeter (15 May 1674 – 24 December 1721), known as Lord Burghley from 1678 to 1700, was a British peer and Member of Parliament. [5] *He was the son of John Cecil, 5th Earl of Exeter, and Anne Cavendish.* [6] …

Question: Who is the paternal grandfather of John Cecil, 7th Earl of Exeter?

Answer: John Cecil, 5th Earl of Exeter

Sentence-level supporting facts: 3, 5

Evidences: ("John Cecil, 7th Earl of Exeter", "father", "John Cecil, 6th Earl of Exeter") and ("John Cecil, 6th Earl of Exeter", "father", "John Cecil, 5th Earl of Exeter")

Figure 1: Example of an inference question in our dataset. The difference between our dataset and HotpotQA is the evidence information that explains the reasoning path.

Our dataset has four types of questions: comparison, inference, compositional, and bridge comparison. All questions in our dataset are created by using a set of predefined templates. Min et al. (2019) classified the comparison questions in HotpotQA in three types: multi-hop, context-dependent multi-hop, and single-hop. Based on this classification, we removed all templates in our list that make questions become single-hop or context-dependent multi-hop to ensure that our comparison questions and bridge-comparison questions are multi-hop. We carefully designed a pipeline to utilize the intersection information between the summary[3] of Wikipedia articles and Wikidata and have a special treatment for each type of question that guarantees multi-hop steps and the quality of the questions. Further, by utilizing the logical rule information in the knowledge graph, such as $father(a, b) \wedge father(b, c) \Rightarrow grandfather(a, c)$, we can create more natural questions that still require multi-hop reasoning.

We conducted two different evaluations on our dataset: difficulty and multi-hop reasoning of the dataset. To evaluate the difficulty, we used a multi-hop model to compare the performance of HotpotQA and our dataset. Overall, the results from our dataset are lower than those observed in HotpotQA, while human scores are comparable on both datasets. This suggests that the number of difficult questions in our dataset is greater than that in HotpotQA. Similar to Min et al. (2019), we used a single-hop BERT model to test the multi-hop reasoning in our dataset. The result of our dataset is lower than the result of HotpotQA by 8.7 F1, indicating that a lot of examples in our dataset require multi-hop reasoning to be solved. Through experiments, we confirmed that although our dataset is generated by hand-crafted templates and the set of predefined logical rules, it is challenging for multi-hop models and requires multi-hop reasoning.

[2] 2Wiki is a combination of Wikipedia and Wikidata.

[3] Another name is "short description"; The short description at the top of an article that summarizes the content. See also https://en.wikipedia.org/wiki/Wikipedia:Short_description

In summary, our main contributions are as follows: (1) We use Wikipedia and Wikidata to create a large and high quality multi-hop dataset that has comprehensive explanations from question to answer. (2) We provide new information in each sample—evidence information useful for interpreting the predictions and testing the reasoning and inference skills of the model. (3) We use logical rules to generate a simple natural question but still require the model to undertake multi-hop reasoning when answering a question. The full dataset, baseline model, and all information that we used when constructing the dataset are available at https://github.com/Alab-NII/2wikimultihop.

2 Task Overview

2.1 Task Formalization and Metrics

We formulated (1) answer prediction, (2) sentence-level SFs prediction, and (3) evidence generation tasks as follows:

- Input: a question Q and a set of documents D.
- Output: (1) find an answer A (a textual span in D) for Q, (2) find a set of sentence-level SFs (sentences) in D that a model used to answer Q, and (3) generate a set of evidence E which consists of triples that describes the reasoning path from Q to A.

We evaluate the three tasks by using two evaluation metrics: exact match (EM) and F1 score. Following previous work (Yang et al., 2018), to assess the entire capacity of the model, we introduced joint metrics that combine the evaluation of answer spans, sentence-level SFs, and evidence as follows:

$$Joint\ F1 = \frac{2P^{joint}R^{joint}}{P^{joint} + R^{joint}} \quad (1)$$

where $P^{joint} = P^{ans}P^{sup}P^{evi}$ and $R^{joint} = R^{ans}R^{sup}R^{evi}$. (P^{ans}, R^{ans}), (P^{sup}, R^{sup}), and (P^{evi}, R^{evi}) denote the precision and recall of the answer spans, sentence-level SFs, and evidence, respectively. Joint EM is 1 only when all the three tasks obtain an exact match or otherwise 0.

2.2 Question Types

In our dataset, we have the following four types of questions: (1) comparison, (2) inference, (3) compositional, and (4) bridge comparison. The inference and compositional questions are the two subtypes of the bridge question which comprises a bridge entity that connects the two paragraphs (Yang et al., 2018).

1. **Comparison question** is a type of question that compares two or more entities from the same group in some aspects of the entity (Yang et al., 2018). For instance, a comparison question compares two or more people with the *date of birth* or *date of death* (e.g., *Who was born first, Albert Einstein or Abraham Lincoln?*).

2. **Inference question** is created from the two triples (e, r_1, e_1) and (e_1, r_2, e_2) in the KB. We utilized the logical rule to acquire the new triple (e, r, e_2), where r is the inference relation obtained from the two relations r_1 and r_2. A question–answer pair is created by using the new triple (e, r, e_2), its question is created from (e, r) and its answer is e_2. For instance, using two triples *(Abraham Lincoln, mother, Nancy Hanks Lincoln)* and *(Nancy Hanks Lincoln, father, James Hanks)*, we obtain a new triple *(Abraham Lincoln, maternal grandfather, James Hanks)*. A question is: *Who is the maternal grandfather of Abraham Lincoln?* An answer is *James Hanks* (Section 3.2).

3. **Compositional question** is created from the two triples (e, r_1, e_1) and (e_1, r_2, e_2) in the KB. Compared with inference question, the difference is that no inference relation r exists from the two relations r_1 and r_2. For instance, there are two triples *(La La Land, distributor, Summit Entertainment)* and *(Summit Entertainment, founded by, Bernd Eichinger)*. There is no inference relation r from the two relations *distributor* and *founded-by*. In this case, a question is created from the entity e and the two relations r_1 and r_2: *Who is the founder of the company that distributed La La Land film?* An answer is the entity e_2 of the second triple: *Bernd Eichinger* (Section 3.2).

4. **Bridge-comparison question** is a type of question that combines the bridge question with the comparison question. It requires both finding the bridge entities and doing comparisons to obtain the answer. For instance, instead of directly compare two films, we compare the information of the directors of the two films, e.g., *Which movie has the director born first, La La Land or Tenet?* To answer this type of question, the model needs to find the bridge entity that connects the two paragraphs, one about the film and one about the director, to get the date of birth information. Then, making a comparison to obtain the final answer.

3 Data Collection

3.1 Wikipedia and Wikidata

In this study, we utilized both text descriptions from Wikipedia[4] and a set of statements from Wikidata to construct our dataset. We used only a summary from each Wikipedia article as a paragraph that describes an entity. Wikidata[5] is a collaborative KB that stores data in a structured format. Wikidata contains a set of statements (each statement includes property and an object entity) to describe the entity. There is a connection between Wikipedia and Wikidata for each entity. From Wikidata, we can extract a triple (s, r, o), where s is a subject entity, r is a property or relation, and o is an object entity. A statement for the entity s is (r, o). An object entity can be another entity or the date value. We categorized all entities based on the value of the property *instance of* in Wikidata (Appendix A.1).

3.2 Dataset Generation Process

Generating a multi-hop dataset in our framework involves three main steps: (1) create a set of templates, (2) generate data, and (3) post-process generated data. After obtaining the generated data, we used a model to split the data into *train*, *dev*, and *test* sets.

(1) Create a Set of Templates: For the comparison question, first, we used Spacy[6] to extract named entity recognition (NER) tags and labels for all comparison questions in the train data of HotpotQA (17,456 questions). Then, we obtained a set of templates L by replacing the words in the questions with the labels obtained from the NER tagger. We manually created a set of templates based on L for entities in the top-50 most popular entities in Wikipedia. We focused on a set of specific properties of each entity type (Appendix A.2) in the KB. We also discarded all templates that made questions become single-hop or context-dependent multi-hop as discussed in Min et al. (2019). Based on the templates of the comparison question, we manually enhanced it to create the templates for bridge-comparison questions (Appendix A.5). We manually created all templates for inference and compositional questions (Appendix A.3 and A.4).

For the inference question, we utilized logical rules in the knowledge graph to create a simple question but still require multi-hop reasoning. Extracting logical rules is a task in the knowledge graph wherein the target makes the graph complete. We observe that logical rules, such as $spouse(a, b) \wedge mother(b, c) \Rightarrow mother\_in\_law(a, c)$, can be used to test the reasoning skill of the model. Based on the results of the AMIE model (Galárraga et al., 2013), we manually checked and verified all logical rules to make it suitable for the Wikidata relations. We obtained 28 logical rules (Appendix A.3).

(2) Generate Data: From the set of templates and all entities' information, we generated comparison questions as described in Algorithm 1 (Appendix A.6). For each entity group, we randomly selected two entities: e_1 and e_2. Subsequently, we obtained the set of statements of each entity from Wikidata. Then, we processed the two sets of statements to obtain a set of mutual relations (M) between two entities. We then acquired the Wikipedia information for each entity. For each relation in M, for example, a relation r_1, we checked whether we can use this relation. Because our dataset is a span extraction dataset, the answer is extracted from the Wikipedia article of each entity. With relation r_1, we obtained the two values o_1 and o_2 from the two triples (e_1, r_1, o_1) and (e_2, r_1, o_2) of the two entities, respectively. The

[4] https://www.wikipedia.org
[5] https://www.wikidata.org
[6] https://spacy.io/

requirement here is that the value o_1 must appear in the Wikipedia article for the entity e_1, which is the same condition for the second entity e_2.

When all information passed the requirements, we generated a question–answer pair that includes a question Q, a context C, the sentence-level SFs SF, the evidence E, and an answer A. Q is obtained by replacing the two tokens *#name* in the template by the two entity labels. C is a concatenation of the two Wikipedia articles that describe the two entities. E is the two triples (e_1, r_1, o_1) and (e_2, r_1, o_2). SF is a set of sentence indices where the values o_1 and o_2 are extracted. Based on the type of questions, we undertake comparisons and obtain the final answer A.

We generated bridge questions as described in Algorithm 2 (Appendix A.6). For each entity group, we randomly selected an entity e and then obtained a set of statements of the entity from Wikidata. Subsequently, based on the first relation information in R (the set of predefined relations), we filtered the set of statements to obtain a set of 1-hop H_1. Next, for each element in H_1, we performed the same process to obtain a set of 2-hop H_2, each element in H_2 is a tuple (e, r_1, e_1, r_2, e_2). For each tuple in H_2, we obtained the Wikipedia articles for two entities e and e_1. Then, we checked the requirements to ensure that this sample can become a multi-hop dataset. For instance, the two paragraphs p and p_1 describe for e and e_1, respectively (see Figure 2). The bridge entity requirement is that p must mention e_1. The span extraction answer requirement is that p_1 must mention e_2. The 2-hop requirements are that p must not contain e_2 and p_1 must not contain e. Finally, we obtained Q, C, SF, E, and A similarly to the process in comparison questions.

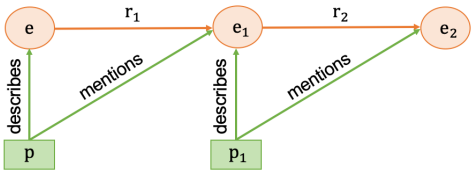

Figure 2: The Requirements for bridge questions in our dataset.

(3) Post-process Generated Data: We randomly selected two entities to create a question when generating the data; therefore, a large number of *no* questions exist in the *yes/no* questions. We performed post-processing to finalize the dataset that balances the number of *yes* and *no* questions. Questions could have several true answers in the real world. To ensure one sample has only one answer, we discarded all ambiguous cases in the dataset (Appendix A.7).

Collect Distractor Paragraphs: Following Yang et al. (2018) and Min et al. (2019), we used bigram tf-idf (Chen et al., 2017) to retrieve the top-50 paragraphs from Wikipedia that are most similar to the question. Then, we used the entity type of the two gold paragraphs (four gold paragraphs for bridge-comparison question) to select the top-8 paragraphs (top-6 for bridge-comparison question) and considered it as a set of distractor paragraphs. We shuffled the 10 paragraphs (including gold and distractor paragraphs) and obtained a context.

Dataset Statistics (A Benchmark Setting): We used a single-hop model (Section 5.1) to split the *train*, *dev*, and *test* sets. We conducted five-fold cross-validation on all data. The average F1 score of the model is 86.7%. All questions solved by the single-hop model are considered as a *train-medium* subset. The rest was split into three subsets: *train-hard*, *dev*, and *test* (balancing the number of different types of questions in each subset). Statistics of the data split can be found in Table 1. We used *train-medium* and *train-hard* as the training data in our dataset.

4 Data Analysis

Question and Answer Lengths We quantitatively analyze the properties of questions and answers for each type of question in our dataset. The statistics of the dataset are presented in Table 2. The compositional question has the greatest number of examples, and the inference question has the least

Name	Split	#Examples
train-medium	train	154,878
train-hard	train	12,576
dev	dev	12,576
test	test	12,576
Total		192,606

Table 1: Data statistics.

Type of Q	#Examples	#Avg. Q	#Avg. A
Comparison	57,989	11.97	1.58
Inference	7,478	8.41	3.15
Compositional	86,979	11.43	2.05
Bridge-comparison	40,160	17.01	2.01
Total	192,606	12.64	1.94

Table 2: Question and answer lengths across the different type of questions. **Q** is the abbreviation for "question", and **A** is for "answer".

number of examples. To ensure one question has only one possible answer, we used the information from Wikidata and removed many inference questions that may have more than one answer. The average question length of the inference questions is the smallest because they are created from one triple. The average question length of the bridge-comparison questions is the largest because it combines both bridge question and comparison question. The average answer lengths of comparison and bridge-comparison questions are smaller than inference and compositional questions. This is because there are many *yes/no* questions in the comparison questions.

Reasoning Type	Example
Comparison question: comparing two entities	**Paragraph A:** Theodor Haecker (June 4, 1879 - April 9, 1945) was a . . . **Paragraph B:** Harry Vaughan Watkins (10 September 1875 – 16 May 1945) was a Welsh rugby union player . . . **Q:** Who lived longer, Theodor Haecker or Harry Vaughan Watkins?
Compositional question: inferring the bridge entity to find the answer	**Paragraph A:** Versus (Versace) is the diffusion line of Italian . . . , a gift by the founder Gianni Versace to his sister, Donatella Versace. . . . **Paragraph B:** Gianni Versace . . . Versace was shot and killed outside . . . **Q:** Why did the founder of Versus die?
Inference question: using logical rules and inferring the bridge entity	**Paragraph A:** Dambar Shah (? – 1645) was the king of the Gorkha Kingdom . . . He was the father of Krishna Shah. . . . **Paragraph B:** Krishna Shah (? – 1661) . . . He was the father of Rudra Shah. **Q:** Who is the grandchild of Dambar Shah?
Bridge-comparison question: inferring the bridge entity and doing comparisons	**Paragraph A:** FAQ: Frequently Asked Questions is a feature-length dystopian movie, written and directed by Carlos Atanes and released in 2004. . . . **Paragraph B:** The Big Money . . . directed by John Paddy Carstairs . . . **Paragraph C:** Carlos Atanes is a Spanish film director . . . **Paragraph D:** John Paddy Carstairs was a prolific British film director . . . **Q:** Are both director of film FAQ: Frequently Asked Questions and director of film The Big Money from the same country?

Table 3: Types of multi-hop reasoning in our dataset.

Multi-hop Reasoning Types Table 3 presents different types of multi-hop reasonings in our dataset. Comparison questions require quantitative or logical comparisons between two entities to obtain the answer. The system is required to understand the properties in the question (e.g., *date of birth*). Compositional questions require the system to answer several primitive questions and combine them. For instance, to answer the question *Why did the founder of Versus die?*, the system must answer two sub-questions sequentially: (1) *Who is the founder of Versus?* and (2) *Why did he/she die?*. Inference questions require that the system understands several logical rules. For instance, to find the *grandchild*, first, it should find the *child*. Then, based on the *child*, continue to find the *child*. Bridge-comparison questions require both finding the bridge entity and doing a comparison to obtain the final answer.

Answer Types We preserved all information when generating the data; hence, we used the answer information (both string and Wikidata id) to classify the types of answers. Based on the value of the property *instance of* in Wikidata, we obtained 708 unique types of answers. The top-5 types of answers in our dataset are: *yes/no* (31.2%), date (16.9%; e.g., July 10, 2010), film (13.5%; e.g., *La La Land*), human (11.7%; e.g., *George Washington*), and big city (4.7%; e.g., *Chicago*). For the remaining types of answers (22.0%), they are various types of entities in Wikidata.

5 Experiments

5.1 Evaluate the Dataset Quality

We conducted two different evaluations on our dataset: evaluate the difficulty and the multi-hop reasoning. To evaluate the difficulty, we used the multi-hop model as described in Yang et al. (2018) to obtain the results on HotpotQA (distractor setting) and our dataset. Table 4 presents the results. For the SFs prediction task, the scores on our dataset are higher than those on HotpotQA. However, for the answer prediction task, the scores on our dataset are lower than those on HotpotQA. Overall, on the joint metrics, the scores on our dataset are lower than those on HotpotQA. This indicates that given the human performance on both datasets is comparable (see Section 5.3), the number of difficult questions in our dataset is greater than that in HotpotQA.

Dataset	Answer		Sp Fact		Joint	
	EM	F1	EM	F1	EM	F1
HotpotQA	44.48	58.54	**20.68**	**65.66**	10.97	40.52
Our Dataset	**34.14**	**40.95**	26.47	66.94	**9.22**	**26.76**

Table 4: Results (%) of the multi-hop model on HotpotQA (Yang et al., 2018) and our dataset. "**Sp Fact**" is the abbreviation for the sentence-level supporting facts prediction task.

Similar to Min et al. (2019), we used a single-hop BERT model (Devlin et al., 2019) to test the multi-hop reasoning in our dataset. The F1 score on HotpotQA is 64.6 (67.0 F1 in Min et al. (2019)); meanwhile, the F1 score on our dataset is 55.9. The result of our dataset is lower than the result of HotpotQA by 8.7 F1. It indicates that a large number of examples in our dataset require multi-hop reasoning to be solved. Moreover, it is verified that our data generation and our templates guarantee multi-hop reasoning. In summary, these results show that our dataset is challenging for multi-hop models and requires multi-hop reasoning to be solved.

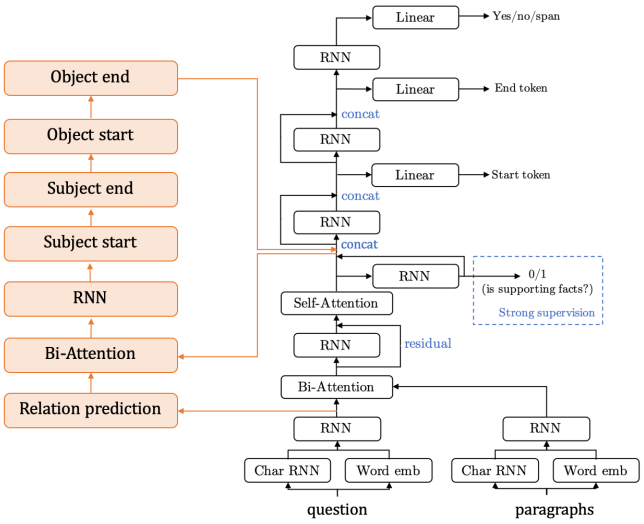

Figure 3: Our baseline model. The right part is the baseline model of HotpotQA (Yang et al., 2018).

5.2 Baseline Results

We modified the baseline model in Yang et al. (2018) and added a new component (the orange block in Figure 3) to perform the evidence generation task. We re-used several techniques of the previous baseline, such as bi-attention, to predict the evidence. Our evidence information is a set of triples, with each triple including *subject entity*, *relation*, and *object entity*. First, we used the question to predict the relations and then used the predicted relations and the context (after predicting sentence-level SFs) to obtain the subject and object entities.

Table 5 presents the results of our baseline model. We used the evaluation metrics as described in Section 2.1. As shown in the table, the scores of the sentence-level SFs prediction task are quite high. This is a binary classification task that classifies whether each sentence is a SF. As discussed, this task is incapable of evaluating the reasoning and inference skills of the model. The scores of the evidence generation task are quite low which indicates this task is difficult. Our error analysis shows that the model can predict one correct triple in the set of the triples. However, accurately obtaining the set of triples is extremely challenging. This is the reason why the EM score is very low. We believe that adding the evidence generation task is appropriate to test the reasoning and inference skills.

Split/Task	**Answer**		**Sp Fact**		**Evidence**		**Joint**	
	EM	F1	EM	F1	EM	F1	EM	F1
Dev	35.30	42.45	23.85	64.31	1.08	14.77	0.37	5.03
Test	36.53	43.93	24.99	65.26	1.07	14.94	0.35	5.41

Table 5: Results (%) of the baseline model.

To investigate the difficulty of each type of question, we categorized the performance for each type of question (on the test split). Table 6 shows the results. For the answer prediction task, the model obtained high scores on inference and compositional questions. Meanwhile, for the sentence-level SFs prediction task, the model obtained high scores on comparison and bridge-comparison questions. Overall, the joint metric score of the inference question is the lowest. This indicates that this type of question is more challenging for the model. The evidence generation task has the lowest score for all types of questions when compared with the other two tasks. This suggests that the evidence generation task is challenging for all types of questions.

Type of Question	**Answer**		**Sp Fact**		**Evidence**		**Joint**	
	EM	F1	EM	F1	EM	F1	EM	F1
Comparison	26.49	27.86	26.76	65.02	0.00	12.40	0.00	2.45
Inference	41.10	62.60	10.77	49.45	0.00	2.85	0.00	1.40
Compositional	50.40	59.94	18.28	57.44	2.57	17.65	0.84	9.19
Bridge-Comparison	18.47	20.45	43.74	89.16	0.00	19.17	0.00	3.60

Table 6: Results (%) of the baseline model on different types of questions.

5.3 Human Performance

We obtained a human performance on 100 samples that are randomly chosen from the test split. Each sample was annotated by three workers (graduate students). We provided the question, context, and a set of predefined relations (for the evidence generation task) and asked a worker to provide an answer, a set of sentence-level SFs, and a set of evidence. Similar to the previous work (Yang et al., 2018), we computed the upper bound for human performance by acquiring the maximum EM and F1 for each sample. All the results are presented in Table 7.

The workers achieved higher performance than that of the model. The human performance for the answer prediction task is 91.0 EM and 91.8 F1. There still seems to be room for improvement, which might be because the mismatch information between Wikipedia and Wikidata makes questions unanswerable

Setting	Answer		Sp Fact		Evidence		Joint	
	EM	F1	EM	F1	EM	F1	EM	F1
Model	50.00	58.48	29.00	69.90	0.00	16.74	0.00	9.79
Human (average)	80.67	82.34	85.33	92.63	57.67	75.63	53.00	66.69
Human Upper Bound (UB)	91.00	91.79	88.00	93.75	64.00	78.81	62.00	75.25

Table 7: Comparing baseline model performance with human performance (%) on 100 random samples.

(see Section 5.4 for an analysis). The human performance of the answer prediction task on our dataset (91.8 F1 UB) shows a relatively small gap against that on HotpotQA (98.8 F1 UB; borrowed from their paper). Although the baseline model is able to predict the answer and sentence-level SFs, it is not very effective at finding the evidence. We also observe that there is a large gap between the performance of human and the model in the evidence generation task (78.8 and 16.7 F1). Therefore, this could be a new challenging task for explaining multi-hop reasoning. We conjecture that the main reason why the score of the evidence generation task was low is the ambiguity in the names of Wikidata. For example, in Wikidata, one person can have multiple names. We use only one name in the ground truth, while the workers can use other names. Future research might explore these issues to ensure the quality of the dataset. Overall, our baseline results are far behind human performance. This shows that our dataset is challenging and there is ample room for improvement in the future.

5.4 Analysis of Mismatched Examples between Wikipedia and Wikidata

As mentioned in Section 5.3, there are unanswerable questions in our dataset due to the mismatch information between Wikipedia articles and Wikidata knowledge. In the dataset generation process, for a triple (s, r, o), we first checked whether the object entity o appears or not in the Wikipedia article that describes the entity s. Our assumption is that the first sentence in the article in which the object entity o appears is the most important, which we decided to use for the QA pair generation. For instance, we obtained a triple: *(Lord William Beauclerk, mother, Lady Diana de Vere)* from Wikidata, and we obtained a paragraph p from the Wikipedia article that describes *"Lord William Beauclerk"*. We used the object entity *"Lady Diana de Vere"* to obtain the first sentence in p *"Beauclerk was the second son of Charles Beauclerk, 1st Duke of St Albans, and his wife Lady Diana de Vere, ..."* From this sentence, we can infer that the mother of *"Lord William Beauclerk"* is *"Lady Diana de Vere"*. However, because we only checked whether the object entity o appears in the sentence or not, there could be a semantic mismatch between the sentence and the triple. For instance, we obtained a triple: *(Rakel Dink, spouse, Hrant Dink)* from Wikidata, while we obtained the first sentence from Wikipedia article: *"Rakel Dink (born 1959) is a Turkish Armenian human rights activist and head of the Hrant Dink Foundation."* Obviously, from this sentence, we cannot infer that *"Hrant Dink"* is the spouse of *"Rakel Dink"*. Therefore, we defined heuristics to exclude these mismatched cases as much as possible. In particular, we found that some examples have subject entities that are similar/equal to their object entities and are likely to become mismatched cases. For such cases, we manually checked the samples and decided to use or remove them for our final dataset. Nonetheless, there are still cases that our heuristics cannot capture. To estimate how many mismatched cases our heuristics cannot capture in the dataset, we randomly selected 100 samples in the training set and manually checked them. We obtained eight out of 100 samples that have a mismatch between Wikipedia article and Wikidata triple. For the next version of the dataset, we plan to improve our heuristics by building a list of keywords for each relation to check the correspondence between Wikipedia sentence and Wikidata triple. For instance, we observed that for the relation "*mother*", the sentences often contain phrases: "*son of*", "*daughter of*", "*his mother*", and "*her mother*".

6 Related Work

Multi-hop questions in MRC domain Currently, four multi-hop MRC datasets proposed for textual data: ComplexWebQuestions (Talmor and Berant, 2018), QAngaroo (Welbl et al., 2018), HotpotQA (Yang et al., 2018), and $\mathrm{R}^4\mathrm{C}$ (Inoue et al., 2020). Recently, Chen et al. (2020) introduced the

HybridQA dataset—a multi-hop question answering over both tabular and textual data. The dataset was created by crowdsourcing based on Wikipedia tables and Wikipedia articles.

Multi-hop questions in KB domain Question answering over the knowledge graph has been investigated for decades. However, most current datasets (Berant et al., 2013; Bordes et al., 2015; Yih et al., 2015; Diefenbach et al., 2017) consist of simple questions (single-hop). Zhang et al. (2018b) introduced the METAQA dataset that contains both single-hop and multi-hop questions. Abujabal et al. (2017) introduced the ComplexQuestions dataset comprising 150 compositional questions. All of these datasets are solved by using the KB only. Our dataset is constructed based on the intersection between Wikipedia and Wikidata. Therefore, it can be solved by using structured or unstructured data.

Compositional Knowledge Base Inference Extracting Horn rules from the KB has been studied extensively in the Inductive Logic Programming literature (Quinlan, 1990; Muggleton, 1995). From the KB, there are several approaches that mine association rules (Agrawal et al., 1993) and several mine logical rules (Schoenmackers et al., 2010; Galárraga et al., 2013). We observed that these rules can be used to test the reasoning skill of the model. Therefore, in this study, we utilized the logical rules in the form: $r_1(a, b) \wedge r_2(b, c) \Rightarrow r(a, c)$. ComplexWebQuestions and QAngaroo datasets are also utilized KB when constructing the dataset, but they do not utilize the logical rules as we did.

RC datasets with explanations Table 8 presents several existing datasets that provide explanations. HotpotQA and $\mathrm{R^4C}$ are the most similar works to ours. HotpotQA provides a justification explanation (collections of evidence to support the decision) in the form of a set of sentence-level SFs. $\mathrm{R^4C}$ provides both justification and introspective explanations (how a decision is made). Our study also provides both justification and introspective explanations. The difference is that the explanation in our dataset is a set of triples, where each triple is a structured data obtained from Wikidata. Meanwhile, the explanation in $\mathrm{R^4C}$ is a set of semi-structured data. $\mathrm{R^4C}$ is created based on HotpotQA and has 4,588 questions. The small size of the dataset implies that it cannot be used for training end-to-end neural network models involving the multi-hop reasoning with comprehensive explanation.

Task/Dataset	**Explanations**		**Size**
	Justification	Introspective	
Our work	✓	✓	192,606
$\mathrm{R^4C}$ (Inoue et al., 2020)	✓	✓	4,588
CoS-E (Rajani et al., 2019)		✓	19,522
HotpotQA (Yang et al., 2018)	✓		112,779
Science Exam QA (Jansen et al., 2016)		✓	363

Table 8: Comparison with other datasets with explanations.

7 Conclusion

In this study, we presented 2WikiMultiHopQA—a large and high quality multi-hop dataset that provides comprehensive explanations for predictions. We utilized logical rules in the KB to create more natural questions that still require multi-hop reasoning. Through experiments, we demonstrated that our dataset ensures multi-hop reasoning while being challenging for the multi-hop models. We also demonstrated that bootstrapping the multi-hop MRC dataset is beneficial by utilizing large-scale available data on Wikipedia and Wikidata.

Acknowledgments

We would like to thank An Tuan Dao, Johannes Mario Meissner Blanco, Kazutoshi Shinoda, Napat Thumwanit, Taichi Iki, Thanakrit Julavanich, and Vitou Phy for their valuable support in the procedure of constructing the dataset. We thank the anonymous reviewers for suggestions on how to improve the dataset and the paper. This work was supported by JSPS KAKENHI Grant Number 18H03297.

References

Abdalghani Abujabal, Mohamed Yahya, Mirek Riedewald, and Gerhard Weikum. 2017. Automated template generation for question answering over knowledge graphs. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1191–1200, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Rakesh Agrawal, Tomasz Imieliundefinedski, and Arun Swami. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, page 207–216, New York, NY, USA. Association for Computing Machinery.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, October. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. volume abs/1506.02075.

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for Multi-hop reasoning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4026–4032, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July. Association for Computational Linguistics.

Wenhu Chen, Hanwen Zha, Zhi yu Chen, Wenhan Xiong, Hong Wang, and Wei Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. *ArXiv*, abs/2004.07347.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Dennis Diefenbach, Thomas Tanon, Kamal Singh, and Pierre Maret. 2017. Question answering benchmarks for Wikidata. 10.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 413–422, New York, NY, USA. Association for Computing Machinery.

Naoya Inoue, Pontus Stenetorp, and Kentaro Inui. 2020. R4C: A benchmark for evaluating RC systems to get the right answer for the right reason. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6740–6750, Online, July. Association for Computational Linguistics.

Peter Jansen, Niranjan Balasubramanian, Mihai Surdeanu, and Peter Clark. 2016. What's in an explanation? Characterizing knowledge and inference requirements for elementary science exams. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2956–2965, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. volume abs/1907.11692.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Prismatic Inc, Steven J. Bethard, and David Mcclosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *In ACL, System Demonstrations*.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4249–4257, Florence, Italy, July. Association for Computational Linguistics.

Stephen Muggleton. 1995. Inverse entailment and Progol.

J. R. Quinlan. 1990. Learning logical definitions from relations. volume 5, page 239–266, USA, September. Kluwer Academic Publishers.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! Leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy, July. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July. Association for Computational Linguistics.

Stefan Schoenmackers, Jesse Davis, Oren Etzioni, and Daniel Weld. 2010. Learning first-order Horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1088–1098, Cambridge, MA, October. Association for Computational Linguistics.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium, October-November. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana, June. Association for Computational Linguistics.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China, July. Association for Computational Linguistics.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. 2018b. Variational reasoning for question answering with knowledge graph. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*.

A Data Collection Details

A.1 Data Preprocessing

We used both dump[7] and online version of Wikipedia and Wikidata. We downloaded the dump of English Wikipedia on January 1, 2020, and the dump of English Wikidata on December 31, 2019. From Wikidata and Wikipedia, we obtained 5,950,475 entities. Based on the value of the property *instance of* in Wikidata, we categorized all entities into 23,763 groups. In this dataset, we focused on the most popular entities (top-50 for comparison questions). When checking the requirements to ensure the multi-hop reasoning of the dataset, several entities in the multi-path are not present in the dump version; in such situations, we used the online version of Wikipedia and Wikidata.

We observed that the quality of the dataset depends on the quality of the intersection information between Wikipedia and Wikidata. Specifically, for the property related to date information, such as *publication date* and *date of birth*, information between Wikipedia and Wikidata is quite consistent. Meanwhile, for the property *occupation*, information between Wikipedia and Wikidata is inconsistent. For instance, the Wikipedia of the entity *Ebenezer Adam* is as follows: "*Ebenezer Adam was a Ghanaian educationist and politician.*"; meanwhile, the value from Wikidata of the property *occupation* is *politician*. In such situations, we manually check all samples related to the property to ensure dataset quality. For the property related to the country name, we handled many different similar names by using the aliases of the entity and the set of demonyms. Moreover, to guarantee the quality of the dataset, we only focused on the set of properties with high consistency between Wikipedia and Wikidata.

We used both Stanford CoreNLP (Manning et al., 2014) and Spacy to perform sentence segmentation for the context.

A.2 Comparison Questions

Table 9 presents all information of our comparison question. We can use more entities and properties from Wikidata to create a dataset. In this version of the dataset, we focused on the top-50 popular entities in Wikipedia and Wikidata. To ensure dataset quality, we used the set of properties as described in the table. For each combination between the entity and the property, we have various templates for asking questions to ensure diversity in the questions.

A.3 Inference Questions

We argued that logical rules are difficult to apply to multi-hop questions. We obtained a set of 50 inference relations, but we cannot use all of it into the dataset. For instance, the logical rule is $placeofbirth(a, b) \wedge country(b, c) \Rightarrow nationality(a, c)$; this rule easily fails after checking the requirements. To guarantee the multi-hop reasoning of the question, the document describing a person a having a place of birth b should not contain the information about the country c. However, most paragraphs describing humans often contain information on their nationality.

The other issue is ensuring that each sample has only one correct answer on the two gold paragraphs. With the logical rule being $child(a, b) \wedge child(b, c) \Rightarrow grandchild(a, c)$, if a has more than one child, for instance a has three children b_1, b_2 and b_3, then each b has their own children. Therefore, for the question "*Who is the grandchild of a?*", there are several possible answers to this question. To address this issue in our dataset, we only utilized the relation that has only one value in the triple on Wikidata. That is the reason why the number of inference questions in our dataset is quite small. Table 10 describes all inference relations used in our dataset.

In most cases, this rule will be correct. However, several rules can be false in some cases. In such situations, based on the Wikidata information, we double-checked the new triple before deciding whether to use it. For instance, the rule is $doctoral\_advisor(a, b) \wedge employer(b, c) \Rightarrow educated\_at(a, c)$, a has an advisor is b, b works at c, and we can infer that a studies at c. There can be exceptions that b works at many places, and c is one of them, but a does not study at c. We used Wikidata to check whether a studies at c before deciding to use it.

To obtain the question, we used the set of templates in Table 11.

[7] https://dumps.wikimedia.org/

Entity Type	Property	#Templates
Human	date of birth	7
	date of death	3
	date of birth and date of death (year old)	2
	occupation	18
	country of citizenship	11
	place of birth	1
Film	publication date	5
	director	2
	producer	2
	country of origin	7
Album	publication date	5
	producer	2
Musical group	inception	4
	country of origin	7
Song	publication date	5
Museum, Airport, Magazine, Railway station, Business, Building, Church building, High school, School, University	inception	1-3
	country	4
Mountain, River, Island, Lake, Village	country	4

Table 9: Templates of Comparison questions.

A.4 Compositional Questions

For this type of question, we utilized various entities and properties on Wikidata. We used the following properties (13 properties) as the first relation: *composer, creator, director, editor, father, founded by, has part, manufacturer, mother, performer, presenter, producer, and spouse*. Further, we used the following properties (22 properties) as the second relation: *date of birth, date of death, place of birth, country of citizenship, place of death, cause of death, spouse, occupation, educated at, award received, father, place of burial, child, employer, religion, field of work, mother, inception, country, founded by, student of, and place of detention*. A compositional question was created by combining the first relation and the second relation (ignore duplicate case).

We used the following entities (15 entities) to create this type of question: *human, film, animated feature film, album, university, film production company, business, television program, candy, written work, literary work, musical group, song, magazine, newspaper*. We obtained a total of 799 templates.

Relation 1	Relation 2	Inference Relation
spouse	spouse	co-husband/co-wife
spouse	father	father-in-law
spouse	mother	mother-in-law
spouse	sibling	sibling-in-law
spouse	child	child/stepchild
father	father	paternal grandfather
father	mother	paternal grandmother
father	spouse	mother/stepmother
father	child	sibling
father	sibling	uncle/aunt
mother	mother	maternal grandmother
mother	father	maternal grandfather
mother	spouse	father/stepfather
mother	child	sibling
mother	sibling	uncle/aunt
child	child	grandchild
child	sibling	child
child	mother	wife
child	father	husband
child	spouse	child-in-law
sibling	sibling	sibling
sibling	spouse	sibling-in-law
sibling	mother	mother
sibling	father	father
doctoral student	educated at	employer
doctoral student	field of work	field of work
doctoral advisor	employer	educated at
doctoral advisor	field of work	field of work

Table 10: Inference relation information in our dataset.

Relation	Template(s)
aunt, child-in-law, child, co-husband, co-wife, father-in-law, father, grandchild, grandfather, grandmother, husband, mother-in-law, mother, sibling-in-law, sibling, stepchild, stepfather, stepmother, uncle, wife	Who is the #relation of #name? Who is #name's #relation?
educated at	Which #instance_of_answer did #name study at? Which #instance_of_answer did #name graduate from?
employer	Which #instance_of_answer does #name work at? Where does #name work?
field of study	What is the field of study of #name?

Table 11: Templates of Inference question.

A.5 Bridge-comparison Questions

The top-3 popular entities on Wikipedia and Wikidata are *human*, *taxon*, and *film*. In this type of question, we focused on the combination between *human* and *film*. Table 12 presents the combination between the relations from the two entities *human* and *film* in our dataset.

Relation 1	Relation 2
director	date of birth
director	date of death
director	country of citizenship
producer	date of birth
producer	date of death
producer	country of citizenship

Table 12: Bridge-comparison question's information.

For each row in Table 12, we have several ways to ask a question. For instance, in the first row, with the combination of the two relations *director* and *date of birth*, we have various ways to ask a question, as shown in Table 13. To avoid ambiguous cases, we ensured that each film we used has only one director or one producer. A total of 62 templates was obtained for this type of question.

Templates
Which film has the director born first, #name or #name?
Which film whose director was born first, #name or #name?
Which film has the director who was born first, #name or #name?
Which film has the director born earlier, #name or #name?
Which film has the director who was born earlier, #name or #name?
Which film whose director is younger, #name or #name?
Which film has the director born later, #name or #name?
Which film has the director who was born later, #name or #name?
Which film has the director who is older than the other, #name or #name?
Which film has the director who is older, #name or #name?

Table 13: Templates of Bridge-comparison questions.

A.6 Generate Data

The algorithms for generating comparison questions and bridge questions are described in Algorithm 1 and Algorithm 2, respectively.

A.7 Post-process Generated Data

For the bridge questions, we created the data from the two triples (e, r_1, e_1) and (e_1, r_2, e_2). When we have another triple (e, r_1, e_{1*}) that has the same entity and the property with the first triple, it becomes an ambiguous case. Hence, we discarded all such cases in our dataset based on the information from Wikidata.

For the comparison questions, when a question is asked for comparing two entities about numerical values and the values of the two entities are equal, we remove it.

Algorithm 1: Comparison Question Generation Procedure

Input: Set of all templates, all entities in the same group, Wikipedia and Wikidata information for each entity

Output: A question–answer pair with these information: question Q, answer A, context C, sentence-level SFs SF, and evidences E

```
while not finished do
    Randomly choose two entities e_1 and e_2;
    Obtain all triples (relations and objects) of each entity from Wikidata;
    Obtain a set of mutual relations (M) between two entities;
    Obtain Wikipedia information of each entity;
    for each relation in M do
        if pass requirements then
            Choose a template randomly;
            Generate a question Q;
            Obtain a context C;
            Obtain an evidence E;
            Compute an answer A;
            Compute sentence-level SFs SF;
        end
    end
end
```

Algorithm 2: Bridge Question Generation Procedure

Input: Set of relations R, Wikipedia and Wikidata information for each entity

Output: A question–answer pair with these information: question Q, answer A, context C, sentence-level SFs SF, evidences E

```
1  while not finished do
2      Randomly choose an entity e;
3      Obtain a set of statements (relations and objects) of the entity from Wikidata;
4      Filter the set of statements based on the first relation information in R to obtain a set of 1-hop
         H_1;
5      For each element in H_1, do the same process (from Line 3) to obtain a set of 2-hop H_2, each
         element in H_2 is a tuple (e, r_1, e_1, r_2, e_2);
6      for each tuple in H_2 do
7          Obtain Wikipedia articles for two entities: e and e_1;
8          if pass requirements then
9              Choose a template randomly based on r_1 and r_2;
10             Generate a question Q;
11             Obtain a context C;
12             Obtain an evidence E;
13             Obtain an answer A;
14             Compute sentence-level SFs SF;
15         end
16     end
17 end
```