JEMHopQA: Dataset for Japanese Explainable Multi-Hop Question Answering

Ai Ishii[1], **Naoya Inoue**[2,1], **Hisami Suzuki**[1], **Satoshi Sekine**[1]
[1]RIKEN AIP, Tokyo, Japan
[2]Japan Advanced Institute of Science and Technology, Ishikawa, Japan
ai.ishii@riken.jp, naoya-i@jaist.ac.jp, hisami.suzuki@a.riken.jp, satoshi.sekine@riken.jp

Abstract

We present JEMHopQA, a multi-hop QA dataset for the development of explainable QA systems. The dataset consists not only of question-answer pairs, but also of supporting evidence in the form of derivation triples, which contributes to making the QA task more realistic and difficult. It is created based on Japanese Wikipedia using both crowd-sourced human annotation as well as prompting a large language model (LLM), and contains a diverse set of question, answer and topic categories as compared with similar datasets released previously. We describe the details of how we built the dataset as well as the evaluation of the QA task presented by this dataset using GPT-4, and show that the dataset is sufficiently challenging for the state-of-the-art LLM while showing promise for combining such a model with existing knowledge resources to achieve better performance.

Keywords: Multi-hop Question Answering, Dataset, Explainability, Large Language Models

1. Introduction

Question Answering (QA) is one of the hallmark tasks that evaluates language understanding capabilities of NLP systems. We are currently witnessing the flourishment of highly capable large language models (LLMs) that solve this complex task, requiring both knowledge and inference skills, with an impressive accuracy (Bang et al., 2023).

However, it is not clear exactly to what extent such LLMs possess the knowledge needed to solve QA problems and how accurately they perform inference to leverage that knowledge. How often do LLMs rely on "hallucinated" knowledge during inference? Can these hallucinations be remedied by structured knowledge bases (KBs) carefully crafted by humans? In English, plenty of benchmarks are already available for such fine-grained investigation, ranging from multi-hop QA datasets (Yang et al., 2018; Trivedi et al., 2022) to human reasoning-annotated multi-hop QA datasets (Inoue et al., 2020; Ho et al., 2020; Dalvi et al., 2021; Geva et al., 2021); however, such datasets are rarely available in other languages.

In this paper, we present JEMHopQA (Japanese Explainable Multi-Hop Question-Answering), a dataset for multi-hop QA in Japanese which includes not only question-answer pairs but also supporting evidence in the form of *derivation triples*. A derivation triple is a form of knowledge representation capturing a relationship between two entities, and is widely compatible with knowledge stored in LLMs and many existing KBs. As such, our dataset allows the community to conduct a thorough evaluation of QA systems from a knowledge point of view, including the coverage of knowledge and the skill to properly operationalize the knowledge to solve a question. The dataset has the following characteristics:

- It consists of 1,179 multi-hop QA pairs with two types of questions, *composition* and *comparison*, across a diverse set of entity categories. Examples (translated into English) are in Fig. 1 and Fig. 2.
- It was created using both crowd-sourced human annotation as well as a GPT model to ensure that the data creation process scales while ensuring naturalness and diversity in data.
- The dataset has been evaluated to be sufficiently challenging even for a state-of-the-art LLM.

For the remainder of the paper, we present some background and related work in §2 and describe the task that our dataset addresses in §3. §4 and 5 describe the data creation process and dataset details, followed by an evaluation of the dataset in §6. The dataset is made publicly available along with an evaluation script and can be downloaded from https://github.com/aiishii/JEMHopQA.

2. Related Work

Multi-hop QA is a question answering task that requires to combine multiple knowledge resources via inference to obtain an answer. There are many such datasets available for the research community. Among them, HotpotQA (Yang et al., 2018), $\mathcal{R}^4\mathcal{C}$ (Inoue et al., 2020), and 2WikiMultihopQA (Ho

LREC-COLING 2024, pages 9515–9525

et al., 2020) are the datasets that include the knowledge needed to solve the question as evidence information.

HotpotQA is created using crowdsourcing and contains a substantial number of natural language questions written by native speakers to examine if machines can perform logical operations such as comparison and inference. In this dataset, each question-answer pair is supplemented by supporting facts (SFs) which are the sentences that contain the information supporting the predicted answer.

However, as pointed out by Inoue et al. (2020), SFs are the sentences that contain relevant and irrelevant portions of text with regards to the QA task, and extracting SFs is equivalent to a simple binary classification task that cannot adequately measure the inference capability of a system. To address this limitation, Inoue et al. (2020) proposed $\mathcal{R}^4\mathcal{C}$ dataset, which supplements the Hotpot QA dataset with semi-structured reasoning blocks called derivations, each of which consisting of two entities with a relation. Our dataset is inspired by this work and further extends it in a new language (Japanese) where there is no pre-existing multi-hop QA dataset.

2WikiMultihopQA creates a large number of new data using structured knowledge from Wikidata (Vrandečić and Krötzsch, 2014), and is similar to the $\mathcal{R}^4\mathcal{C}$ dataset in that the knowledge used for answer prediction is given as evidence information. However, since they are created based on existing KBs, they are not suitable for measuring the coverage of knowledge needed to solve a QA task. Datasets based on existing knowledge bases, such as KQA Pro (Cao et al., 2022) and KoRC (Yao et al., 2023), are similarly unsuitable for measuring knowledge coverage. On the other hand, even if a dataset is not based on an existing knowledge base, such as MuSiQue (Trivedi et al., 2022), it is difficult to directly measure knowledge coverage as it does not contain the knowledge for solving questions as evidence.

Representative QA datasets in Japanese are JSQuAD and JCommonsenseQA (both included in JGLUE (Kurihara et al., 2022), JAQKET (Suzuki et al., 2020)) and driving domain QA datasets (Takahashi et al., 2019). However, none of these datasets include multi-hop questions or derivation, making our dataset the first in Japanese explainable multi-hop QA.

3. Task Description

3.1. Task Definition

JEMHopQA presents an explainable QA task that requires QA systems to explain their answer to a question. Formally, given a question Q, the task is (i) to predict the answer A, and (ii) to generate a derivation D that justifies A as the prediction.

Question: What is the name of the mayor of the city where the Louvre Museum is located?
Derivation: ("Louvre Museum", "location", "Paris") and ("Paris", "mayor", "Anne Hidalgo")
Answer: Anne Hidalgo

Figure 1: Example of a composition question.

Question: Was "Jurassic Park" released earlier than "E.T."?
Derivation: ("E.T.", "release date", "Jun 11, 1982") and ("Jurassic Park", "release date", "Jun 11, 1993")
Answer: NO

Figure 2: Example of a comparison question.

While there are several options for the design of the derivation, we adopt a form that represents a semi-structured relationship between two entities in the form of triples that is highly compatible with measuring the coverage of existing KBs (including potential KBs stored in the LLM) and the operational capability of KBs.

Following Inoue et al. (2020), derivation D is defined as a set of *derivation steps*, where each derivation step $d_i \in D$ represents a semi-structured relationship between two entities, $d_i \equiv \langle d_i^s, d_i^r, d_i^o \rangle$. d_i^s and d_i^o represent the subject and object entities respectively (corresponding to Wikipedia article titles), and d_i^r the relation between them, expressed as a noun phrase (such as "location", "release date"). Because our QA task is multi-hop, each question-answer pair is always accompanied by two or more derivation steps[1]. To make the task more realistic, we do not limit the set of relations to any constrained set. While this makes the evaluation of the task more difficult, we cope with this issue by allowing similarity matches in our evaluation metric (see details in §3.3).

3.2. Question Types

Our task comprises of two types of questions requiring different ways to obtain the necessary knowledge and apply inference, thus posing different types of challenges to a QA system:

1. *Composition questions* require an inference over two derivation triples where they are connected by a bridge entity, which is the object entity of one triple and the subject of another.

[1] For the most part, a multi-hop question requires exactly two derivations, but some require more, as in "Who started the singing career at an earlier age between Beyoncé and Björk?", which requires two relations (birthdate and debut date) for both entities.

For example, in Fig. 1, "Paris" serves as the bridge entity -- it is the object of (Louvre Museum, location, Paris) and the subject of (Paris, mayor, Anne Hidalgo). The key to solving a composition question is to find the bridge entity which is implicit in the question itself, and apply an inference chain over the triples. The derivation steps are two triples from two Wikipedia pages (e_1, r_1, e_2) and (e_2, r_2, e_3), where e_1,e_2 are entities which are Wikipedia article headings, r_1,r_2 is a relation, and e_2 is used as a bridge entity. e_1, r_1 and r_2 are represented in the question and e_3 is the answer.

2. *Comparison questions* require obtaining two triples with the same relation, and applying logical inference over the two object entities along the relation. For example, in Fig. 2, two movie name entities "E.T." and "Jurassic Park" are compared along the relation of "release date" for inferring which one comes before. The derivation step includes two triples (e_1, r, e_2) and (e_3, r, e_4) with the same relation r, based on the two Wikipedia articles of entities e_1 and e_3.

3.3. Evaluation Metrics

In this subsection, we describe the evaluation metrics used to measure the performance of a QA system on our task.

Derivation Derivation steps are semi-structured in the form of triples, yet both entities and relations are subject to spelling variations and paraphrases, which makes it difficult to evaluate automatically by strict string matches. Therefore, we follow Inoue et al. (2020) to allow string similarities between reference and predicted derivation steps in computing the match score.

More specifically, given a reference derivation G and a system derivation D, we compute the alignment score $c(D; G)$ and calculate precision, recall and f_1 scores based on it. Assuming $|G|$ and $|D|$ are the number of derivation triples for a given question, we define precision, recall and f_1 as follows:

$$\mathrm{pr}(D) = \frac{c(D;G)}{|D|}, \mathrm{rc}(D) = \frac{c(D;G)}{|G|}$$
$$\mathrm{f_1}(D) = \frac{2 \cdot \mathrm{pr}(D;G) \cdot \mathrm{rc}(D;G)}{\mathrm{pr}(D;G) + \mathrm{rc}(D;G)}$$

For computing an alignment score $c(D; G)$, we first select the best alignment between G and D. For this, we define $c\,(D; G, A_i)$ which is the alignment score D given a particular alignment A_i, and choose the best alignment so as to maximize the sum of component alignment scores for a given question:

$$c(D; G, A_i) = \sum_{(d_i, g_i) \in A_i} a(d_i, g_i)$$
$$\mathrm{c}(D; G) = \max_{A_i \in \mathcal{A}(D,G)} c(D; G, A_i),$$

where $a(d_i, g_i)$ is a similarity score $[0,\ 1]$ between two derivation steps d_i and g_i. In this work, we use normalized Levenshtein distance over tokenized words[2], and as a preprocessing step, address word spelling variations with a synonym dictionary[3] augmented with the spelling variations observed in our dataset[4].

For evaluating the derivation correctness, we use three scorers: f_1^{ent}, f_1^{rel}, and f_1^{full}. f_1^{ent} is the average of the similarity $a(d_i, g_i)$ over subject and object entities, f_1^{rel} is $a(d_i, g_i)$ of relations of d_i and g_i, f_1^{full} is an average of $a(d_i, g_i)$ over subject and object entities and relations.

Answer We use Exact Match (EM, equivalent to accuracy) as well as Similarity Match (SM) score that considers the string similarity between the candidate and the reference answers with synonym extensions when the answer is a noun phrase, using the same similarity computation as described above.

4. Data Creation

In this section, we describe the details of the data creation method of JEMHopQA, whose goal is to create a natural sounding dataset for explainable QA with diverse topic and answer types. First, we select the Wikipedia pages to create the dataset from, and then proceed to creating the composition and comparison questions. For generating questions, we use both crowd-sourced human annotations and an LLM to ensure the scalability of the data creation while ensuring the naturalness and diversity of the data. Crowd-sourced tasks were run on Lancers[5], a crowdsourcing platform in Japan, and for LLM, we used GPT-3.5. Finally, we perform post-processing to ensure quality. We describe each of these steps in turn below.

[2]For tokenization, we used Sudachi available at https://github.com/WorksApplications/SudachiPy

[3]https://github.com/WorksApplications/SudachiDict/blob/develop/docs/synonyms.md

[4]https://github.com/WorksApplications/chikkar

[5]https://www.lancers.jp/

4.1. Wikipedia Page Selection

As a preprocessing step, we select a subset of Wikipedia pages, since not all Wikipedia pages are suitable for the generation of multi-hop QA data. For example, pages on abstract concepts tend not to include hyperlinks to other entities which are critical for multi-hop question generation. To choose a suitable subset, we use entity categories of the article headings. For the entity categories, we referred to Extended Named Entity (ENE) proposed by Sekine (2008) and used in Sekine et al. (2020) which provides Wikipedia pages annotated with ENE labels[6]. Sekine's ENE is hierarchical; we used the second level categorization consisting of 26 categories[7].

We then remove the pages that are labeled as CONCEPT (pages on abstract concepts), IGNORED (e.g., meta pages which do not explain an entity or concept, such as "list of X") and those that included R18-restricted material. Furthermore, as the crowd workers are not expected to be familiar with rarely used entities, we select only popular Wikipedia pages using pageview scores. For this, we used the average of Popularity scores of Wikipedia pages obtained from the dump file of Cirrussearch from 2017 to 2021 and ranked the pages by popularity. In order to have the final dataset to reflect the ENE category distribution of Wikipedia as a whole, we computed the distribution and selected the pages for each ENE category from the top of the ranked list until the final set contained 10K pages.

4.2. Creation of Composition Questions

In this subsection, we explain how we generate composition questions. This takes two steps: (1) creating derivation triples that share a bridge entity; (2) writing a question that uses the derivation triples from (1). We used both crowdsourcing and LLM for the data creation to see the effectiveness of these methods.

Step (1): Creation of derivation triples for composition questions The task is to annotate the relation between the subject entity (page title) and the object entity (entity of a hyperlink found in Infobox or Abstract). In the crowdsourcing task, the user interface highlights the relevant part of the Wikipedia article (right in Fig. 3) when crowd worker selects the box for describing the relation between a particular subject-object entity pair (left in Fig. 3). Crowd workers read the text before and after the highlighted section and type the relation into the text box. They were paid 16 yen per task instance. For GPT-3.5, we created a prompt using the same information available to crowd workers and instructed it to respond with a relation. An example prompt is given in Fig. 6.

Step (2): Creation of composition questions This task generates questions from the derivation triples between the two pages sharing a bridge entity generated in (1). In crowdsourcing, a crowd worker selects triples on the triple selection screen (left in Fig. 4) and fills in the text box on the input screen (right in Fig. 4) with the question generated using the selected triples. They were paid 11 yen for this task. For GPT-3.5, we use a prompt that includes the pre-determined derivation triples and asks it to output a question based on them (Fig. 6 bottom).

4.3. Creation of Comparison Questions

For creating comparison questions, we start by creating candidate page pairs using the page category information assigned to Wikipedia articles. Namely, we randomly extract pairs of pages with the same page category within the 10K articles obtained in §4.1 and filter out those that did not have the same "instance_of" category in Wikidata[8]. For the extracted pairs, we randomly assign one of the three answer types, YES, NO (as in Fig. 2) or OTHER (where the answer is an entity, as in "Which one is a World Heritage Site, Notre Dame Cathedral or the Hôtel des Invalides?"), so that the final ratio of these question types becomes 1:1:2.

For this task we only used crowdsourcing as we were able to obtain enough samples from this method alone. The crowdsourcing setup is the same as for the composition question creation task, and the workers were paid 11 yen per this task instance.

In this task, a crowd worker sees two pages side by side, and is asked to create a question and the associated derivation triples. Referring to the comparison screen (left in Fig. 5), they enter the questions and associated triples in the text boxes of the input screen (right in Fig. 5).

4.4. Post-processing

As a result of the data creation process described above, we obtained 443 composition and 971 comparison questions respectively using crowdsourcing, and 2,445 composition questions from GPT-3.5. However, this dataset contains many errors and

[6]We used CirrusSearch Dump file 2021-08-23 provided by https://2023.shinra-project.info and an HTML-version of Japanese Wikipedia 2021-08-20.

[7]We used ENE version 9.0, which includes 294 categories in 4 levels of hierarchy.

[8]https://www.wikidata.org/

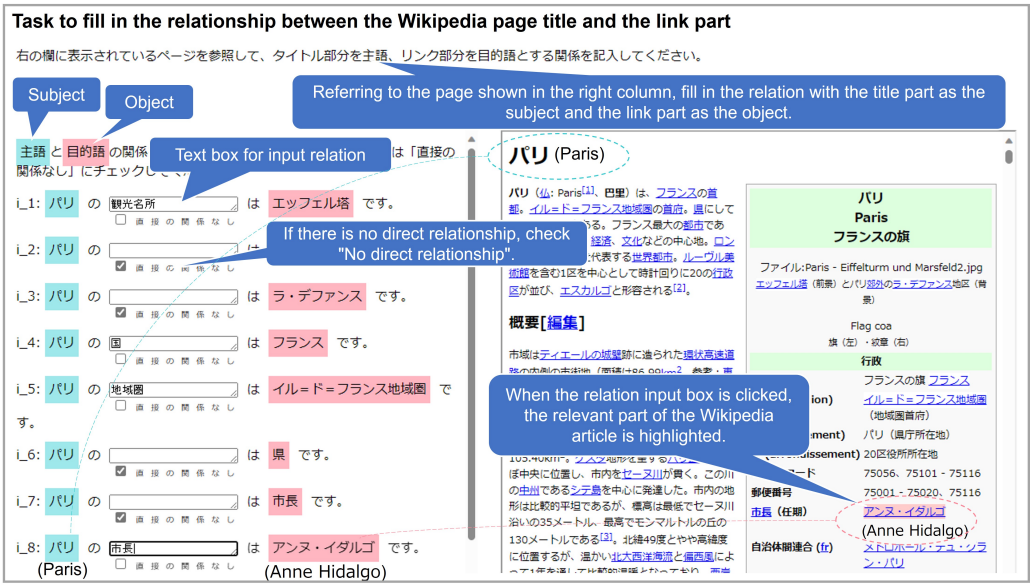

Figure 3: Crowdsourcing interface for creation of derivation triples for composition question.

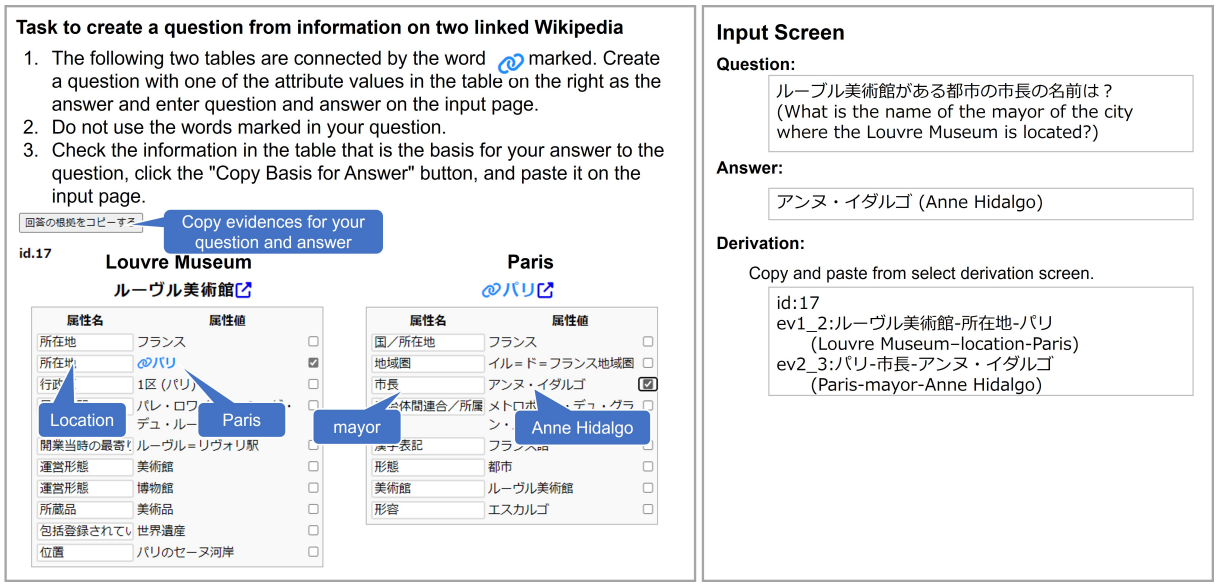

Figure 4: Crowdsourcing interface for creation of composition questions.

Figure 5: Crowdsourcing interface for creation of comparison questions.

Prompt for Creation of derivation triples:
Read the TEXT, answer each RELATION what each WORD in the TEXT is to the TITLE.
- Each WORD is a TARGET separated by a new line.
- Each RELATION should be answered in a straightforward manner.
- If there is no direct relationship between TITLE and WORD, answer "none".
- Output the RELATION corresponding to the WORD in JSON format as shown in the OUTPUT of EXAMPLE.
—EXAMPLE:
TITLE: Tracy Wilson
TEXT: Tracy Wilson (September 25, 1961) is a female figure skater from Racine, Quebec,... (omitted)
TARGET:
Canada
Figure skater... (omitted)
OUTPUT:.
{"Canada": "Place-of-origin", "Figure Skater": "Occupation", (omitted)}

Prompts for Composition Question Creation:
Create a question according to the following instructions.
Instructions:
- Create a QUESTION with TITLE, ATTRIBUTE1, ATTRIBUTE2, and ANSWER information, where ANSWER is the answer.
- The QUESTION should be as natural a statement as possible.
(omitted)
- Please answer only the QUESTION as output.
—EXAMPLE:
(omitted)

Figure 6: Example of prompt for creating triples and composition questions.

thus requires post-processing. Table 1 describes the statistics around the data clean-up process.

As mentioned above, we first ran an automated clean-up process because the dataset contains errors due to mistakes or lack of understanding by the crowd workers and GPT. This process checks the premises of the task (e.g., entities that are not the bridge or answer entity are missing from a composition question). Then the authors manually verified the dataset and either removed the instances with errors or corrected them.

As shown in the table, 90% of the comparison questions remained in the final dataset, while only 65.4% and 41.6% of the generated data made the final set for composition questions, for crowdsourced and GPT-derived methods respectively. This is due to the complexity of the respective tasks, as the composition task requires two steps as opposed to one in the comparison task. Moreover, during the creation of composition questions step (2), crowd workers had access to the user interface to select a suitable triple for making a question from derivation triple candidates (Fig. 4), whereas the GPT model did not have a process to make such a selection, which explains the large number of automatic removal for this method. A common GPT error (10%) in manual removal was the case where the bridge entity was not utilized to make the question multi-hop. For example, given the triples (Beyoncé, spouse, Jay-Z) and (Jay-Z, hometown, State of New York), the question generated was “Where is Beyoncé from?”. Another type of error seen in both crowd and GPT (11% of manual removal of both) is when the relationship between two entities does not uniquely determine the subject-object entity pair. For example, the question “Who is the mother of Paul McCartney's son?”

	Composition		**Comparison**
	Crowd	GPT	Crowd
Original	443	2,445	971
After automated clean-up	376	464	822
Manual removal	130 (34.6%)	271 (58.4%)	82 (10.0%)
Manual correction	207 (55.1%)	102 (22.0%)	267 (32.5%)
Final	246 (65.4%)	193 (41.6%)	740 (90.0%)

Table 1: Data clean-up statistics.

	Composition	**Comparison** (YES, NO, OTHER)	**ALL**
Train	392	667 (174+173+320)	1,059
Test	47	73 (22+23+28)	120
ALL	439	740 (196+196+348)	1,179

Table 2: Distribution of question types in JEMHopQA.

5. Dataset

5.1. Dataset Statistics

Table 2 and Table 3 summarize the details of the final dataset. We have split the dataset randomly into train and test sets. As mentioned above, we have 1:1:2 ratio of YES/NO/OTHER (=entity) answer types for comparison questions, and along with composition questions, we have a diverse and balanced dataset in terms of answer types.

5.2. Question Type Diversity

Table 4 shows the types of questions in our dataset according to how they are answered. Composition questions are answered either by providing an entity (as in Fig. 1) or a date (e.g. “Which year was the mayor of Paris born?”) via a bridge entity. Comparison question can be solved in one of the

	Question avg./ max.	Answer avg./ max.	Derivations 2/3/4
Composition	27.4/55	5.6/31	390/47/2
Comparison	34.6/84	4.1/21	726/1/13
ALL	31.9/84	4.6/31	1,116/48/15

Table 3: Length (in character) of questions and answers; number of instances per the number of derivation triples.

	Train	Test
Composition (total)	392 (37.0%)	47 (39.2%)
Entity answer	255 (24.1%)	32 (26.7%)
Number answer	137 (12.9%)	15 (12.5%)
Comparison (total)	667 (63.0%)	73 (60.8%)
Number comparison	297 (28.1%)	33 (27.5%)
Shared predicate	208 (19.6%)	30 (25.0%)
Entity selection	162 (15.3%)	10 (8.3%)

Table 4: Distribution of question types by answer type.

three ways: (i) compare the numbers in the triples (as in Fig. 2); (ii) check if the two subject entities share a predicate (i.e., relation-object pair, as in "Did Barrack and Michele Obama go to the same college?"); or (iii) check if either of the subject entity satisfies the predicate (e.g., "which one of the countries border China – India or Thailand?)

5.3. Topic Category Diversity

Table 5 shows the distribution of named entity categories of the articles used in our final dataset, according to the second-level categories in ENE version 9.0 (Sekine et al., 2020). There are 11 relevant categories to our dataset at this level, excluding temporal and number categories. Of these 11, JEMHopQA covers 8 categories – the 3 that were not covered (Disease, Deity and Color) are the categories that do not render themselves easily into a bridge entity or a comparison. Our data distribution simulates the original Wikipedia distribution better than Wikipedia TopView, which is the distribution of articles by page view[9]. The latter is dominated by Person category (~55%) while our dataset successfully avoids this skew and remains closer to the original Wikipedia ratio (30-40%).

6. Evaluation Using GPT-4

Recent LLMs such as GPT models show a surprising level of performance on NLP tasks that require both knowledge and reasoning. How well do these

[9]Derived from 1K monthly top page view data (2017-2022) from https://pageviews.wmcloud.org/topviews/?project=ja.wikipedia.org

ENE	Wikipedia	Wikipedia TopView	JEMHop QA
Person	31.16%	54.58%	39.36%
Product	24.13%	29.75%	31.17%
Facility	12.38%	0.63%	9.84%
Location	8.97%	1.94%	10.98%
Organization	8.10%	9.57%	12.21%
Event	3.15%	3.72%	0.76%
Natural Object	2.58%	0.88%	0.13%
Individual Living Thing	0.34%	0.45%	0.34%
Disease	0.23%	0.68%	0.00%
Deity	0.14%	0.05%	0.00%
Color	0.02%	0.00%	0.00%

Table 5: Distribution of entity category.

LLMs answer the questions in JEMHopQA, and to what extent can they give accurate derivation steps? We evaluate GPT-4, one of the largest publicly available LLMs, using our dataset.

6.1. Setup

For evaluation, we use the JEMHopQA test set, which consists of 47 composition questions and 73 comparison questions. We use `gpt-4-0613` model via OpenAI API[10] with a temperature parameter of 1.0. We experimented with the following five types of prompts[11] (translated into English):

1. **Zero-shot**: ask a question only, as in:

> Answer the following question with a simple noun phrase or by YES or NO.
> Are Jaws and Mr. Nobody both produced by Universal Pictures?
> =>

2. **5-shot**: include 5 random samples from the training set as few-shot examples, as in:

> Answer the following question with a simple noun phrase or by YES or NO.
> What is the name of the mayor of the city where the Louvre is located? => Anne Hidalgo
> Which city has a larger population, Nara-city or Dubai? => Dubai
> (...3 more examples)
> Are Jaws and Mr. Nobody both produced by Universal Pictures?
> =>

[10]https://platform.openai.com/

[11]We do not report results for the zero-shot CoT setting because it was difficult to find good CoT prompts that did not cause formatting errors without using examples in our preliminary experiment. Even after fixing the formatting errors by hand, they didn't perform as well as the few-shot settings.

3. **Chain-of-Thought (CoT) 5-shot**: following (Wei et al., 2022), add an instruction to provide a CoT reasoning path, along with 5 few-shot samples.

> Answer the following question with a simple noun phrase or by YES or NO, along with the evidence for each step of reasoning.
> What is the name of the mayor of the city where the Louvre is located? => (Louvre Museum, location, Paris); (Paris, mayor, Anne Hidalgo) => Anne Hidalgo
> Which is has a larger population, Nara City or Dubai? => (Nara City, population, about 352,000); (Dubai, population, 3,310,022) => Dubai
> (...3 more examples)
> Are Jaws and Mr. Nobody both produced by Universal Pictures?
> =>

4. **Gold D**: provide gold derivation D, as in:

> Answer the following question with a simple noun phrase or by YES or NO.
> Are Jaws and Mr. Nobody both produced by Universal Pictures? =>(Jaws, production company, Universal Pictures);(Mr. Nobody, production company, Perfect World Pictures)
> =>

5. **Gold D 5-shot**: include 5 random samples from the training set as few-shot examples with gold derivation D, as in:

> Answer the following question with a simple noun phrase or by YES or NO.
> What is the name of the mayor of the city where the Louvre is located? => (Louvre Museum, location, Paris); (Paris, mayor, Anne Hidalgo) => Anne Hidalgo
> Which is has a larger population, Nara City or Dubai? => (Nara City, population, about 352,000); (Dubai, population, 3,310,022) => Dubai
> (...3 more examples)
> Are Jaws and Mr. Nobody both produced by Universal Pictures? =>(Jaws, production company, Universal Pictures);(Mr. Nobody, production company, Perfect World Pictures)
> =>

The maximum token limit is set to 32 for the zero-shot and 5-shot prompts, and to 256 for the CoT prompt. Due to the sampling-based decoding of GPT-4 API, we run each experiment three times and report the average of all runs.

Our evaluation metrics are Answer Exact Match (EM) and Similarity Match (SM) defined in §3.3.

6.2. Results

How well can GPT-4 answer multi-hop QA correctly? Let us first look at the cases where GPT-4 needs to answer the questions without access to gold derivation. The results are shown in Table 6. It shows that a few-shot setting is effective, boosting the accuracy from 48.9% to 55.6% on EM. Furthermore, CoT setting improved the accuracy by about 4%.

	Answer EM	Answer SM
Zero-shot	0.489	0.507
5-shot	0.556	0.571
CoT 5-shot	**0.597**	**0.629**

Table 6: Results of GPT-4 with different prompts.

	Answer EM / SM	Derivation $f_1^{\text{ent}}/f_1^{\text{rel}}/f_1^{\text{full}}$
composition	0.305/0.385	0.552/0.72/0.606
comparison	0.785/0.785	0.724/0.707/0.718
ALL	0.597/0.629	0.656/0.712/0.674

Table 7: Detailed results of GPT-4 CoT 5-shot.

In order to better understand the GPT performance, we present and analyze the CoT 5-shot results in more detail in Table 7. It shows that GPT-4 CoT is much better at answering comparison questions (78%) than composition questions (30%). f_1^{ent}, a measure of the correctness of the entities in the derivation, was 0.552 for the composition question and 0.724 for the comparison questions, with a gap of about 17%.This is because comparison questions explicitly mention the two subject entities being compared along with a comparison scale, which makes the generation of derivation triples more accessible to GPT-4, while in composition questions, bridge entities are implicit, and therefore more difficult to identify correctly.

Given the correct derivation triple, can GPT-4 infer correctly? Given that the majority of wrong answers were shown to be due to entity identification, we examined GPT-4's ability to infer when given a correct derivation triple as input. For this, we used the Gold D and Gold D 5-shot settings, where correct derivation triples are given. In order to assess the impact of the order in which these derivations are given, we tested both the consistent (the order in which the triples appear in the dataset) and random (RND) orders. The evaluation results are shown in Table 8.

EM of Gold D and Gold D 5-shot, respectively, improved by about 40% compared from Zero-shot

	Answer EM	Answer SM
Gold D	0.906	0.924
Gold D (RND)	0.914	0.939
Gold D 5-shot	0.956	0.976
Gold D 5-shot (RND)	0.953	0.969

Table 8: Results of GPT-4 providing gold D

	Wikidata (W)	Shinra (S)	W+S	GPT-4	GPT-4+W+S
Full coverage	30.0%	50.0%	63.3%	40.0%	77.5%
Partial coverage	27.5%	29.2%	22.5%	23.3%	15.8%
No coverage	42.5%	20.8%	14.2%	36.6%	6.7%

Table 9: Coverage of derivation steps in the test set by existing KBs and GPT-4.

and 5-shot results without derivations as shown in Table 6. Furthermore, the accuracy of the Gold 5-shot setting was approximately 95%. This means that for GPT-4, the difficulty in solving JEMHopQA lies almost exclusively in the identification of the derivation, and that if a correct derivation is given, it can infer the answer correctly. The difference between the results for changing the orders of the gold derivations is small, indicating that GPT-4 is not answering using the order.

6.3. Manual Analysis of GPT-4 Errors

What factors contributed to the erroneous answers? To further investigate the GPT-4 behavior, we manually analyzed the 49 wrong answers produced by the CoT 5-shot setting. We found that only one of them had an error in inference (i.e., the derivations were correct, but the answer was wrong); the remaining 48 cases had errors in derivation (i.e., hallucination). Of the 49, 33 were composition questions, where about 70% of errors stem from mis-identifying the bridge entity. This confirms that the main difficulty of composition questions is in the identification of implicit bridge entities.

We also observed that the model often gives a correct answer for a wrong reason: of the 71 questions where the model correctly answered, 34% of them (24/71) contained an error in the supporting derivation triples. Out of these errors, 87% (21/24) were comparison questions, where the derivation steps were factually wrong but correct enough only for the purpose of comparison. For example, in "Who was born first, X or Y? Answer: X", the model predicted wrong birthdays for X and Y, but the relative order was right, which provides enough information for the model to answer the question correctly. The remaining 13% (3/24) were composition questions, where the bridge entity was a non-existent organization or individual in Wikipedia, but the final answer was correct.

Can errors in the derivation triple be remedied by using external KBs? In total, GPT-4 "hallucinated" wrong derivation triples in about 60% of the 120 questions. We investigated whether this knowledge hallucination issue can be remedied by using external KBs. For this investigation, we used two existing Japanese KBs on Wikipedia, namely Wikidata and Shinra[12](Sekine et al., 2019), The latter extracts attribute-value pairs from Wikipedia articles and structures them according to the ENE categories in Sekine et al. (2020). Knowledge representation in both KBs is compatible with the derivation triples used in our task, allowing for a straightforward application to our task.

In Table 9, the first three columns show the coverage of the KBs on the 120 samples from the Test set, whether the required derivation triples for a question is found in Wikidata or Shina or the union of both. As a multi-hop question requires two or more triples to answer, a partial coverage statistic is also given. They show that Wikidata and Shinra can only provide full evidence to JEMHopQA questions 30% and 50% of the time respectively, and 63% for both KBs combined. Some examples of the triples not covered by these KBs include finer-grained relations than those in KBs (e.g., “sister” instead of “sibling”) or those that do not have any counterpart in these KBs for their specificity (e.g., “piano_lesson_start_date”).

The last two columns show the coverage of derivation triples of GPT-4 and GPT-4 combined with both KBs. While the coverage of GPT-4 is not higher than these KBs combined, they seem to complement each other as the combination of all of them can cover 77% of the derivation triples for the Test set. This indicates that a further improvement in the task presented by JEMHopQA is possible by combining LLM with existing KBs. Such an investigation is left for future work.

7. Conclusions

In this paper, we introduced JEMHopQA, a dataset for multi-hop QA that includes evidence in the form of derivation triples to make the QA task more challenging and realistic. The dataset was created using both crowdsourced human annotations as well as a GPT model to ensure the scalability of the data creation process while ensuring the naturalness and diversity of the data. Our experiments show that even GPT-4, one of the largest publicly available LLMs, struggles with our dataset due to hallucination in reasoning. We also show that the hallucinated knowledge can be potentially corrected by existing KBs, which opens up a new challenge of integrating structured KBs into LLMs for explainable multi-hop QA in Japanese.

[12] http://shinra-project.info/

8. Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP20269633 and 19K20332. The authors would like to thank the anonymous reviewers for their insightful feedback.

9. Bibliographical References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity.

Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. 2022. KQA pro: A dataset with explicit compositional programs for complex question answering over knowledge base. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6101–6119, Dublin, Ireland. Association for Computational Linguistics.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7358–7370, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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. Association for Computational Linguistics.

Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. 2022. JGLUE: Japanese general language understanding evaluation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2957–2966, Marseille, France. European Language Resources Association.

Masatoshi Suzuki, Jun Suzuki, Koji Matsuda, Kyosuke Nishida, and Naoya Inoue. 2020. JAQKET: Construction of a Japanese QA dataset on the subject of quizzes. In *Proceedings of the 26th Conference on Natural Language Processing in Japan (NLP 2020)*.

Norio Takahashi, Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. 2019. Machine comprehension improves domain-specific Japanese predicate-argument structure analysis. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 98–104, Hong Kong, China. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

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. Association for Computational Linguistics.

Zijun Yao, Yantao Liu, Xin Lv, Shulin Cao, Jifan Yu, Juanzi Li, and Lei Hou. 2023. KoRC: Knowledge oriented reading comprehension benchmark for deep text understanding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11689–11707, Toronto, Canada. Association for Computational Linguistics.

10. Language Resource References

Satoshi Sekine. 2008. Extended named entity ontology with attribute information. In *Proceed-*

ings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco. European Language Resources Association (ELRA).

Satoshi Sekine, Maya Ando, Akio Kobayashi, and Aska Sumida. 2020. Updated Extended Named Entity Definitions and Japanese Wikipedia Classification Data 2019. In *Proceedings of the 26th Conference on Natural Language Processing in Japan (NLP 2020)*.

Satoshi Sekine, Akio Kobayashi, and Kouta Nakayama. 2019. Shinra: Structuring wikipedia by collaborative contribution. In *Conference on Automated Knowledge Base Construction*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.