Review of the paper "Understanding and Improving Zero-shot Multi-hop Reasoning in Generative Question Answering"[1]

Zunino Luca (337560)
CS552 - Modern natural language processing

I. Introduction

Large Language Models (LLMs) are often employed in generative Question Answering (QA), where the model must generate an appropriate answer to a given question in an end-to-end fashion. While most efforts in this area have focused on verifying LLMs performance in answering simple questions, some researchers have demonstrated that specific models can also answer more complex questions, which theoretically require multi-hop reasoning. The paper "Understanding and Improving Zero-shot Multi-hop Reasoning in Generative Question Answering"[1] by Jiang Z., Araki J., et al., delves into the reasoning capabilities of LLMs and suggests training pipelines to enhance these abilities. According to the authors, the mechanism through which LLMs answer complex questions remains inadequately understood. Although there have been demonstrations of LLMs' exceptional performance on multi-hop reasoning tasks, many researchers have argued that these models do not perform reasoning when answering questions but instead rely on pattern matching and memorization of data.

The authors' investigation of LLMs' multi-hop reasoning capabilities for QA tasks holds the potential to significantly contribute to the field. Indeed, question answering is a crucial task for LLMs, and its importance has grown with the emergence of chatbots like ChatGPT, where the model must answer complex user questions that often involve multi-hop reasoning. A deeper understanding of LLMs reasoning capabilities and, more generally, how a model generates answers from questions could enable the implementation of strategies to improve these types of models. To this end, the authors draw on their experimental findings to propose a training method that might enhance LLMs' multi-hop reasoning capabilities. This, in principle, could greatly advance the state of the art.

Throughout the paper, the authors assert that their analysis of generative QA models reveals that these models do not exhibit convincing multi-hop reasoning capabilities, but instead tend to take shortcuts when providing answers. They also claim that they significantly improved the multi-hop performance of open-book and closed-book QA models by finetuning them on concatenated single-hop questions and SPARQL queries.

While the authors' claims regarding their work are significant, not all aspects of the paper are equally persuasive. The presented analysis is rigorous, and although the metric used (exact match) might overly penalize partial answers that could still demonstrate multi-hop reasoning, it effectively allows for the conclusion that robust multi-hop reasoning does not naturally emerge in the analyzed models, necessitating finetuning. However, the method proposed to improve multi-hop performance using SPARQL queries is less convincing. This approach is primarily advocated by the authors to reduce the cost and complexity of finetuning (as SPARQL queries are easier to obtain than multi-hop question datasets), but a multi-hop questions dataset is used to generate the SPARQL queries employed for finetuning. Moreover, since the analysis focuses on two models (UnifiedQA and RAG), it is difficult to generalize to other models the conclusion that multi-hop reasoning does not naturally emerge without finetuning. Therefore, the work's main contribution lies in its ability to provide novel empirical insights into a known problem through a meticulous analysis of datasets and QA models: the conclusion that generative QA models do not exhibit convincing multi-hop reasoning capabilities (and that specific finetuning is required for these to emerge) is sound and offers new insights that could benefit the field. The suggested extension regarding the training pipeline is a valuable complement and could serve as an interesting starting point, but in its current form, it does not appear to be a groundbreaking contribution to the field.

In the subsequent sections of this review, these aspects will be examined in greater detail. Specifically, the experimental setup will be considered, and the experimental results will be presented. Then, building on the authors' analysis, an evaluation of the paper will be conducted, particularly highlighting its merits and potential areas for improvement.

II. Experimental setup

In order to investigate the reasoning capabilities of LLMs, the authors began by formulating three research questions, which serve as the foundation for the analysis conducted in the paper.

- Research question 1: The authors aimed to understand the relationship between a model's answers to multi-hop questions and its answers to the single-hop questions that compose the multi-hop questions. Specifically, they sought to determine whether the

correctness of the answers to single-hop questions is necessary and/or sufficient for the correctness of the complete, multi-hop question, and whether the answers are consistent in both cases.

- Research question 2: The authors aimed to ascertain whether models trained on single-hop questions could generalize to multi-hop questions (zero-shot generalization).
- Research question 3: Considering that models often do not exhibit true and robust multi-hop reasoning abilities, the authors sought to determine whether training on approximations of multi-hop questions could improve these abilities.

In their thorough examination of LLMs' multi-hop reasoning capabilities, the authors considered two main classes of generative QA models: closed-book and open-book. The primary distinction between the two classes is that closed-book models must answer questions without access to external resources (relying solely on model parameters, which necessitates proper storage of all required information), while open-book models can retrieve relevant context from external sources. The authors chose to investigate both types of models in their analysis, as the difference in received input could potentially lead to distinct multi-hop reasoning mechanisms. In the analysis, closed-book models are formalized as sequence-to-sequence models that take a question $\mathbf{q}$ as input and calculate the probability of an answer $\mathbf{a}$ based on θ, the model parameters: $P(\mathbf{a}|\mathbf{q};\theta) = \prod_{i=1}^{|\mathbf{a}|} P(a_i|\mathbf{q}, \mathbf{a_{<i}};\theta)$. As mentioned earlier, the key difference for open-book models is their ability to retrieve relevant context $(\mathbf{c})$ from external sources and utilize this context to generate answers: $P(\mathbf{a}|\mathbf{c},\mathbf{q};\theta) = \prod_{i=1}^{|\mathbf{a}|} P(a_i|\mathbf{c},\mathbf{q}, \mathbf{a_{<i}};\theta)$. In the analysis, the authors selected the UnifiedQA model as an example of a closed-book model. This model, based on the T5 model, is further fine-tuned on numerous QA datasets (using a procedure that converts various QA formats into a sequence-to-sequence format). As a representative open-book model, the authors chose the RAG model, which comprises a retriever (based on the dense passage retrieval model) that searches for relevant context and a generator (based on BART) that generates answers based on the question and retrieved context.

The proposed method to evaluate the multi-hop reasoning capabilities of LLMs in an open-book setting involves starting with multi-hop questions $\mathbf{q}$ containing T hops (in the analysis, T is fixed at 2) and decomposing these into T single-hop questions $(\mathbf{q_t}, t \in \{1, ..., T\})$. For each single-hop question $\mathbf{q_t}$, $\mathbf{a_t}$ represents the corresponding answer, $\hat{\mathbf{a_t}}$ denotes the answer predicted by the model, and $\mathbf{c_t}$ refers to the corresponding retrieved context (the same letters without the subscript indicate the question, answer, predicted answer, and context for the multi-hop question). In the setting considered, the model is queried using all the single-hop questions $\mathbf{q_t}$ and multi-hop questions $\mathbf{q}$ concatenated with the corresponding context ($\mathbf{c_t}$ or $\mathbf{c}$); the predicted single-hop and multi-hop answers are generated using greedy decoding:

$$\begin{matrix}\hat{\mathbf{a}} \\ \hat{\mathbf{a_t}}\end{matrix} = \arg\max_y P\left(y \left| \begin{matrix}[c,]\mathbf{q} \\ [c_t,]\mathbf{q_t}\end{matrix} ;\theta\right)\right.\ .$$

Before the analysis, both the open-book and closed-book models were finetuned on single-hop and multi-hop QA pairs from the ComplexWebQuestions training set. Conversely, predictions were made starting from the test sets of both ComplexWebQuestions and HotpotQA.

The analysis focused on two main aspects. First, the correlation between the correctness of the decomposed, single-hop questions and the correctness of the multi-hop questions was investigated. Next, the consistency between the answers provided by the model to multi-hop questions and those given to the corresponding chain of single-hop questions was examined.

Regarding the first aspect, correctness confusion matrices were produced, in which the dataset entries were grouped into four bins based on the correctness of the two single-hop questions ($s_1 s_2 = \{00, 01, 10, 11\}$, where s_1 and s_2 represent the correctness of the predictions generated from the first and second single-hop questions, respectively). For each of the bins, the percentage of correct multi-hop answers associated with that particular configuration ($s = \{1, 0\}$, where s represents the correctness of the prediction generated from the multi-hop question) was computed. Subsequently, the conditional success rate on multi-hop questions $P(s = 1|s_1 s_2)$ was calculated. This parameter indicates the likelihood of correctly answering multi-hop questions based on the correctness of the answers to single-hop, decomposed questions; for robust models, $P(s = 1|s_1 s_2 = 11)$ should be close to one, and $P(s = 1|s_1 s_2 = \{00, 01, 10\})$ should be close to zero.

Regarding the second aspect, the authors sought to examine the prediction consistency between multi-hop questions and chains of decompositions. Specifically, they considered two predictions from multi-hop questions ($\hat{a_1}*$ and $\hat{a}$, where $\hat{a_1}*$ is the intermediate answer, and $\hat{a}$ is the answer to the multi-hop question) and two predictions from decomposed, single-hop questions in sequence ($\hat{a_1}$ and $\hat{a_2}*$, where $\hat{a_1}$ is the prediction for the first-hop question and $\hat{a_2}*$ is the prediction for the second-hop question, in which the predicted answer to the first hop $\hat{a_1}$ has been substituted) and compared the results to measure their consistency.

After analyzing the experimental results (presented in detail in the next section), the authors demonstrated that the LLMs under analysis have a limited zero-shot capacity for multi-hop reasoning if they are not trained on multi-hop questions, as the performance significantly degraded in such a setting. Since datasets containing multi-hop questions are expensive to obtain, the authors investigated the possibility of improving multi-hop reasoning ability with

datasets containing only single-hop questions or no Natural Language (NL) questions at all. They, therefore, followed two approaches: the first one involves concatenating decomposed single-hop questions to approximate a multi-hop question and finetuning the model on this concatenation, while the second one aims at teaching models multi-hop reasoning without relying on NL questions by executing logical forms, SPARQL in particular (standard query language over knowledge bases). The second approach was proposed since SPARQL queries are easier and cheaper to obtain than multi-hop questions. To test their idea, they associated each multi-hop question contained in the ComplexWebQuestions dataset with a SPARQL query and used each single-hop and multi-hop query as a pseudo input question. Then, to verify how the networks' performances vary according to the finetuning performed, they considered different settings (listed in table I).

Name	Description
Default	Original model without finetuning
SM-NL	Model trained on single-hop and multi-hop NL questions (upper bound of the zero-shot performance)
S-NL	Model trained on decomposed single-hop NL questions
SM-SPARQL	Model trained on single-hop and multi-hop SPARQL queries
S-NL+Concat	Model trained on decomposed single-hop NL questions, also concatenating them
S-NL+Concat+SM-SPARQL	Model trained on decomposed single-hop NL questions, also concatenating them, and on single-hop and multi-hop SPARQL queries

Table I
DIFFERENT MODEL SETTINGS CONSIDERED BY THE AUTHORS IN THE FINETUNING ANALYSIS.

Some aspects that need to be analyzed in more detail to understand the experimental results are the datasets used, the evaluation metric considered, and the idealization of the retriever component in the open-book model.

A. Datasets

To perform their analysis, the authors queried the models on both multi-hop questions and their decompositions into single-hop questions. They decided to start from a dataset containing multi-hop questions and obtained the corresponding single-hop questions by applying heuristics. In particular, two (already existing) datasets have been used in the analysis. The first dataset, ComplexWebQuestions, is used as the main testbed. This dataset has been obtained by combining simpler, single-hop questions coming from the WebQuestionsSP dataset into multi-hop questions by using various heuristics. Specifically, this dataset contains four different types of questions (composition, conjunction, superlative, and comparative), which allow a fine-grained analysis of the various multi-hop question types. An example of a multi-hop question of type "composition" (from the ComplexWebQuestions dataset) can be found in table II. By using the same heuristics used by the creators of the dataset in reverse order, the authors could reconstruct the single-hop questions. To make an example, we can consider the composition type: the dataset has been generated by using questions from WebQuestionsSP as the second (and last hop) and substituting an entity in these questions with a relational phrase. This process is reverted to obtain a chain of single-hop questions. This dataset contains 27,639 questions in the training set and 3,519 in the development set. The second dataset, a subset of multi-hop questions from HotpotQA annotated manually with decompositions, has been used to test the generality of the model. This dataset only contains the composition type of questions and has 1,000 questions in the development set. It should be noted that all the questions from the considered datasets have two hops.

Type	Text of question	Answer
Hop1	Return the country where Limonese Creole is spoken.	Costa Rica
Hop2	Which continent is Costa Rica located?	North America
Multi-hop	On which continent is Limonese Creole spoken?	North America

Table II
EXAMPLE OF A MULTI-HOP QUESTION OF TYPE "COMPOSITION" FROM THE COMPLEXWEBQUESTIONS DATASET.

We can delve deeper into the details of each dataset to understand whether they are suitable for the analysis performed by the authors. ComplexWebQuestions is a large dataset containing complex questions in natural language, each associated with the corresponding SPARQL query to retrieve the answer. It was initially introduced by Talmor A. and Berant J. in the paper "The Web as a Knowledge-base for Answering Complex Questions"[2]. In the paper, the dataset was used to train a model able to decompose complex questions into simpler ones and interact with the web to obtain an answer. This dataset seems a reasonable choice for the analysis performed by the authors of the paper. First, it appears to be widespread in the literature, as the paper introducing the dataset currently has 338 citations, 73 of which are recognized as highly influential by Semantic Scholar[3]. Furthermore, the dataset size is reasonable for the finetuning task of the models considered in the analysis (27,639 questions in the training set and 3,519 questions in the development set), and it is well aligned with what the authors aim to investigate. It is also convenient as it already has the SPARQL queries associated with each natural language question, an aspect which has significantly sped up the creation of the SPARQL dataset introduced by

the authors.

Furthermore, to test the model's generality, the authors used a subset of multi-hop questions from HotpotQA (annotated manually with decompositions). HotpotQA was originally introduced by Yang Z., Qi P., et al. in the paper "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering"[4]. It is a QA dataset collected from the English version of Wikipedia and contains a total of about 113K questions. An interesting aspect is that each question in the dataset comes with two gold paragraphs necessary to answer the question. This dataset, too, seems a reasonable choice for the analysis performed by the authors, and it is even more widespread in the literature: the paper introducing the dataset currently has 1,002 citations, 284 of which are recognized as highly influential by Semantic Scholar[5]. In this case, the authors consider a much smaller subset (1,000 questions), where each multi-hop question has been manually annotated with decompositions.

B. Evaluation metric

The evaluation metric used to verify the correctness of the answers provided by the models is Exact Match (EM), corresponding to the percentage of predictions matching the ground truth exactly. When the generation has multiple answers, the various generated answers are split and matched against all the answers in the ground truth; the prediction is considered correct if all the ground-truth answers, and nothing else, are present in the prediction.

C. Retriever of the open-book model

An essential component of open-book models is the retriever, as it has been demonstrated that a better retrieval component can lead to better performance on QA. The authors decided to ablate out the influence of the retriever and directly provide context containing the answers to the QA model. In this way, a failure in question answering can be directly attributed to the generator instead of the retrieval component. Since gold context is not available in the ComplexWebQuestions dataset, the authors obtained a pseudo-gold context for each single-hop question $c_t = [p_t^{P-G}, p_t^N]$ where p_t^{P-G} is the pseudo-gold passage obtained by the DPR model (first passage among the top-100 generated corresponding to question q_t containing the answer a_t) and p_t^N is a negative passage (first passage retrieved by DPR not containing a_t, incorrect context added to avoid making the task too easy). An important aspect is that, to avoid leaking superficial signals to the model, the concatenation order between p_t^{P-G} and p_t^N is randomized. The context c considered for multi-hop questions is the concatenation of the contexts for single-hop questions: $c = [c_1, ..., c_T]$. It should be noted that the context is kept fixed for all the experiments. Due to the idealization of the context, the "open-book" model is therefore renamed into "oracle-book" by the authors of the paper.

III. Experimental results and analysis

The experimental results provided by the authors can be divided into three parts. The first two parts are related to the two different points of view taken into account in their analysis of generative QA models, while the third part refers to the experiments done on finetuning approaches which could improve the multi-step reasoning capabilities of the models considered.

A. Correlation of correctness

Regarding the first point of view, the correlation of correctness, the authors initially expected that the ability to answer decomposed, single-hop questions would be necessary and/or sufficient to answer multi-hop questions correctly. More specifically, the hypothesis was that if a model can answer multi-hop questions correctly, it should be able to follow the chain of decomposition internally (i.e., perform multi-hop reasoning). Therefore, they expected the performance on multi-hop questions to be significantly lower than the performance on all single-hop questions (due to error propagation). The observed results significantly diverged from this prediction, as on the ComplexWebQuestions dataset, the multi-hop performance was slightly higher than the performance on hop 1, and the gap between hop 2 and multi-hop performance was smaller than anticipated, especially for the oracle-book model. According to the authors, this is a sign that generative QA models manage to take shortcuts while answering multi-hop questions (i.e., they can answer the multi-hop question correctly even without answering correctly to the chain of single-hop questions).

Considering the conditional success rate more in detail, the authors observed that the success rate on multi-hop questions is highest if both single-hop components are correctly answered ($P(s = 1|s_1 s_2 = 11) = 85\%/88\%$ considering closed-books and oracle-books respectively). However, even if both components are answered correctly, the multi-hop answer may diverge from the answer of the second hop, indicating that the models are not necessarily following the chain of components while answering multi-hop questions. Furthermore, the authors considered the success on the second-hop questions and discovered that it is more correlated with success on the multi-hop questions rather than success on first-hop questions ($P(s = 1|s_1 s_2 = 01) = 46\%/75\% > P(s = 1|s_1 s_2 = 10)$, where the first result refers to closed-books model and the second one to oracle-books model). According to the authors, this result indicates that the model can bypass the requirements of the first step and still answer correctly to multi-hop questions: one possible reason is the fact that the first-hop question often involves multiple answers, but some of them may not be necessary for the multi-hop question (i.e., the model can still answer the question correctly if it does not know a part of the answer to hop 1).

Furthermore, it has been systematically verified that the closed-book model is performing significantly worse than the oracle-book one (indicating that it is beneficial to integrate external knowledge) and that both models generalize poorly to the HotpotQA dataset (indicating that the multi-hop reasoning capabilities cannot generalize across different datasets).

B. Consistency between the answers

Regarding the second point of view, the authors observed that the consistency between the answers given by the model to multi-hop questions and to the corresponding chain of single-hop questions is relatively low for both models (although it is even lower for the closed-book over the oracle-book) and on both the first and the second hops (the consistency on the second hop is even lower). According to the authors, these results indicate that the QA models are not necessarily answering multi-hop questions in the same way as they answer the individual, decomposed questions. Furthermore, the lower consistency in closed-book models is due to the fact that implicitly navigating the parameters is harder than searching evidence in context, and the lower consistency in the second hop is because inconsistent intermediate predictions will propagate to the second hop, accumulating inconsistency.

C. Finetuning the models

Concerning their investigation of different ways of finetuning the models, the main experimental results presented by the authors are the following:

- The performance obtained finetuning on single-hop questions only (S-NL) drops by almost half with respect to multi-hop performance on both closed-book and oracle-book models, indicating that compositional generalization is not emerging naturally;
- By finetuning on the simple concatenation of single-hop questions, the multi-hop performance increases significantly with respect to S-NL, indicating that simple concatenation could be an effective approximation for multi-hop questions;
- The models trained only on SPARQL queries can generalize to NL questions to some degree, and their performance is significantly better than the one obtained without finetuning. This indicates that, even if SPARQL queries are much less flexible than NL questions (using pre-defined grammar and explicitly specifying the compositional structure), the ability learnt considering SPARQL queries can be reused by the model;
- Combining the concatenation of single-hop questions (more natural) and SPARQL queries (more explicit in the reasoning process) slightly improves the performance of the closed-book model and slightly decreases the performance of the oracle-book model compared to training on the two supervision types separately. According to the authors, this difference is because closed-book models are less constrained (since no additional context is present) and can benefit from additional supervision;
- All the zero-shot settings considered have significantly lower performance than the upper bound (SM-NL, finetuning on multi-hop questions).

IV. Evaluation

The primary strengths of the authors' analysis lie in its capacity to shed light on the multi-hop reasoning capabilities of generative QA models and outline strategies for their improvement. The analysis is rigorous and provides both a lower bound and an upper bound to the results, making it easier to relate and compare the different strategies for enhancing multi-hop reasoning capabilities via finetuning. Although the generality of the conclusions cannot be asserted, it is noteworthy how poorly the two default (i.e., not finetuned) models considered perform in multi-hop reasoning tasks. A significant strength of this paper is that it effectively highlights this issue and the necessity to find efficient strategies for enhancing these types of capabilities during training or finetuning. Indeed, multi-hop reasoning abilities are especially crucial for chatbots like ChatGPT, which have gained impressive popularity in recent months. This paper is a significant contribution, as the analysis conducted could be easily extended to other models and datasets, potentially leading to broader improvements in multi-hop reasoning across various NLP applications. Additionally, by exploring different finetuning strategies and their effects on model performance, the authors help to pave the way for more effective and targeted model training in the future.

In their analysis, the authors examine multiple settings to address the research questions they propose. A significant strength of this analysis is that the settings encompass the entire spectrum of performance, making it easier to evaluate the proposed solution and compare the various finetuning strategies. The authors consider lower bounds, which serve as a baseline and represent the original, non-finetuned models (Default). As expected, these models exhibit poor performance on the multi-hop task (6.56 multi-hop EM for UnifiedQA, 7.62 multi-hop EM for RAG, with a maximum EM of 100). The upper bounds, on the other hand, consist of models finetuned on both single-hop and multi-hop questions (SM-NL), which demonstrate the best performance among the models examined (33.25 multi-hop EM for UnifiedQA, 60.32 multi-hop EM for RAG). All other configurations investigated by the authors yield scores that fall between these bounds. Given the substantial number of configurations analyzed, where models are trained on NL, concatenation, SPARQL, or a combination thereof, and considering the availability of an upper bound, it is straightforward to assess how the authors' proposed finetuning technique performs:

SM-SPARQL achieves a 24.84 multi-hop EM for UnifiedQA and a 51.60 multi-hop EM for RAG, while Combo records a 27.14 multi-hop EM for UnifiedQA and a 53.07 multi-hop EM for RAG. One of the merits of this paper is the clear presentation of the gap between the proposed finetuning strategy and the upper bound, highlighting the need for further research to enhance the effectiveness of finetuning using SPARQL queries to the level attained by multi-hop NL questions.

While the experimental design is overall convincing, and the conclusions drawn by the authors are solid and accurate, some aspects could have been designed more rigorously. First, the three main simplifications performed (exact match as an evaluation metric, shift from open-book to oracle-book model, and idealized construction of the SPARQL dataset) risk biasing, at least partially, the results that the authors present. In the next subsection, each simplification will be analyzed in more detail. Then, in the following four subsections, an issue concerning the impact of the different architectures will be considered, further areas of improvement will be described, ethical considerations will be presented, and reproducibility will be addressed.

A. Simplifications

1) Evaluation metric: The choice of Exact Match (EM) as the evaluation metric, and the strategy used to decide whether the model answered a given question correctly risk penalizing partial answers too much and partly invalidating the results. To clarify this, we can draw inspiration from an entry of the dataset ComplexWebQuestions: the multi-hop question is "What country with the smallest calling code does the Niger River flow through?" which can be divided into a hop 1 question ("What countries does the Niger River flow through?" [answer: Benin, Guinea, Mali, Niger, Nigeria]) and into a hop 2 question ("Which one of the following country calling codes is smallest: Benin, Guinea, Mali, Niger, Nigeria?" [answer: Mali]). Let us imagine a scenario in which the model was presented with the multi-hop question and reasoned in the following way: "The Niger River flows through Benin, Guinea, Mali, Niger [answer to hop 1: Benin, Guinea, Mali, Niger] $\rightarrow$ The calling code of Benin is +229, the calling code of Guinea is +224, the calling code of Mali is +223, the calling code of Niger is +227 $\rightarrow$ The smallest country calling code is the one of Mali [answer to multi-hop: Mali]". This chain of thought would show a significant degree of multi-hop reasoning, but since EM is used, and since all the entries of the gold answer must be contained in the prediction for the model answer to be correct, the answer to the hop 1 question would be considered wrong (as Nigeria is missing). Therefore, this dataset entry would be categorized in the $s_1 = 0, s_2 = 1, s = 1$ group and would be used as an example of the fact that the model is taking shortcuts instead of applying multi-hop reasoning. A more lenient metric (and more attention to whether the model was able to correctly predict the part of hop 1 answers necessary to continue the chain of reasoning) would have been beneficial and would have strengthened the contributions of the work. The authors seem to be aware of this aspect, as they state that the hop 2 performance is significantly higher than the hop 1 performance in the questions of types "conjunction", "comparative", and "superlative" since hop 1 questions usually have more answers than hop 2 questions, thus being harder. They also correctly state that this aspect does not invalidate their conclusion about correctness correlation since the conditional success rate is used. However, they did not discuss whether their decision on how to decide whether s_1, s_2 have values 0 or 1 is the best one. Furthermore, the authors indicate that "generative QA models manage to take shortcuts instead of performing real reasoning". They identify shortcuts as superficial signals or as the fact that "for multi-hop questions with multiple intermediate answers, generative QA models might not need to know all of them in order to answer the multi-hop question". The fallacy of this reasoning is that these two shortcuts that have been identified are profoundly different in their nature, as not knowing one of the answers to the intermediate question is not necessarily implying an inability to perform real reasoning.

2) Shift from open-book to oracle-book: The shift from open-book to oracle-book (explained in the "experimental setup" section) is arguably the most well-grounded simplification among the three considered. Indeed, this approach is taken as the authors aim to ablate the influence of the retrieval component so that the failure to answer a question cannot be attributed to a suboptimal retriever. Nevertheless, a significant part of the challenge of open-book models is retrieving meaningful context, and this idealized scenario in which a pseudo-gold context is given to the model can potentially change how the model performs multi-hop reasoning. Furthermore, even if the authors wanted to ablate the influence of the retriever, it would have probably been more convincing to consider a dataset containing gold contexts. In this way, the context passed to the model would have been controlled more tightly, and it would have been possible to guarantee (at least to a certain extent) that the retrieved contexts are not too easy nor too hard for the model. The authors justify the choice of considering pseudo-gold context by stating that the chosen dataset did not contain gold contexts.

3) Idealized construction of the SPARQL dataset: Although the improvement obtained over the baseline (models without any finetuning) when finetuning the models on SPARQL queries is significant and could pave the way for an effective finetuning on QA tasks, the approach followed by the authors, and the conclusions drawn, are not entirely convincing. First of all, the authors state that datasets containing multi-hop questions are difficult and expensive to obtain, while one of the advantages of SPARQL queries

is that they can be acquired more easily. Nevertheless, they create the SPARQL queries on which they finetune the model starting from the ComplexWebQuestions dataset, which is a multi-hop questions dataset. Therefore, the analysis does not effectively demonstrate whether SPARQL queries obtained "in the wild" could genuinely be helpful for the model but merely converts a multi-hop questions dataset into a SPARQL dataset. Although reasonable as a starting point, this approach seems a bit too idealized and does not eliminate the need for a multi-hop questions dataset. Furthermore, the obtained performance is worse than S-NL+Concat (model trained on decomposed single-hop NL questions, also concatenating them), and the authors, therefore, state: "converting the SPARQL queries into NL questions and training models on them can [...] further improve performance". The first criticism is that, at this point, the innovation would be a more cost-effective way of obtaining multi-hop questions and not a new way of teaching multi-hop reasoning capabilities by executing logical forms. Then, "in the wild" SPARQL queries would be necessary for this (otherwise, it would just mean going back to the starting point, that is, to multi-hop questions), but the authors failed to provide evidence of the effectiveness of this approach.

B. Impact of different architectures

During their analysis, the authors state that the oracle-book RAG model performed significantly better than the closed-book UnifiedQA models because of the possibility of gathering external evidence. While this conclusion is reasonable, the authors fail to consider the impact that the different architectures of the two models may have had on performances, and this aspect could potentially invalidate the conclusion, at least partly.

C. Further areas of improvements

One of the main aspects that seem to be missing in the analysis performed by the authors of the paper is a comparison of the performance obtained on the idealized, multi-hop SPARQL queries (obtained starting from a multi-hop questions dataset) and on existing query logs. Indeed, the authors claim that finetuning on SPARQL queries is cheaper and easier than tuning on multi-hop questions, but they use a multi-hop question dataset to generate SPARQL queries. Given that the performance obtained by training on SPARQL queries is worse than the one obtained on S-NL+Concat (model trained on decomposed single-hop NL questions, also concatenating them) in both closed-book and open-book settings, the main selling point would be the reduced cost of obtaining training data (and the possibility of combining the different approaches, as the "Combo." setting performed in a convincing way), but one such dataset is not considered in the analysis. This additional analysis would have strengthened the contributions of the work as it would have verified in a more transparent way whether "in the wild", already existing query logs are effective in increasing the multi-hop reasoning capabilities of QA models. Without this analysis, it is impossible to exclude that the performance boost comes from the underlying structure of the multi-hop questions (questions which would be non-trivial to obtain using existing query logs) and not from the SPARQL structure itself. The first part of the analysis, considering existing query logs without additional constraints on their characteristics, would have been relatively straightforward but not necessarily effective. The difficult part of this approach would have been identifying multi-hop queries in the query logs, but it would have been necessary to show the feasibility of such an approach (if it ends up being more costly and time-consuming than directly creating multi-hop questions, then it is probably better to consider these directly).

D. Ethical considerations

Like many works in natural language processing, the finetuned models can be affected by ethical issues. In particular, the authors are considering two models in their analysis, UnifiedQA (based on T5) and RAG. Since biases have been consistently reported for T5 (see, for example, [6]), it is reasonable to expect that these biases are translated to UnifiedQA as well, and as a consequence also to the finetuned model. Furthermore, there is no guarantee that the questions contained in the datasets used for the analysis and the finetuning are free from stereotypes and toxic behaviour, which could further affect the resulting models. Another potential issue for these generative QA models is the leakage of private information, as while answering a question (single-hop or multi-hop), the model could expose private and sensitive information present in the training data.

While the authors' main contribution is the analysis of multi-hop capabilities in models finetuned using different strategies rather than the models themselves, no ethics issues are explicitly addressed or presented in the paper. This could be because the authors are considering well-established models and datasets, which have been studied multiple times (at least T5) concerning ethical issues, and these analyses could have been taken for granted. In a certain sense, it is also possible to relate the main research question to ethical considerations: knowing whether a model can perform sound multi-hop reasoning to answer a question or if it is just inferring the answer from the context (or even more, using biases contained in the dataset) could be a significant way of addressing ethical concerns.

Considering, for example, a question like: "Which was the religion of the terrorist involved in the attack of [...]?", the model could exploit a racist bias (see [6], in which T5 completes "All terrorists are ..." with "Muslims") to directly answer the question without decomposing the question in a first hop ("Who was the terrorist involved in the attack of [...]?") and in a second hop ("Which was

the religion of [...]?"). This highlights the importance of understanding multi-hop reasoning capabilities and their potential impact on the ethical dimensions of QA systems. To ensure the responsible use and development of these models, researchers should be mindful of the biases present and work towards mitigating them through various debiasing techniques, improving the overall fairness and ethicality of the models.

E. Reproducibility

The reproducibility of the core part of the analysis conducted by the authors is ensured through the availability of the code used for the analysis and finetuning of the models, which can be found at the corresponding GitHub repository[7]. Additionally, the datasets employed are clearly identified, and the code for processing them is also provided within the repository. Furthermore, the hyperparameters used while finetuning the models have been clearly identified:

- To finetune the UnifiedQA model, the default hyperparameters have been considered. The procedure lasted for 100K steps and considered a batch size of 16 on a single TPU;
- To finetune the RAG model, the default hyperparameters have been considered. The procedure lasted for 10 epochs and considered a batch size of 4 on a single V100 GPU.

However, supplementary material offering further explanation on the analysis carried out and elaborating on the key design choices would have facilitated a better understanding and replication of the study. Nonetheless, a thorough examination of the code should suffice in validating most of the authors' design choices, making the analysis and the work convincingly reproducible.

REFERENCES

[1] Z. Jiang, J. Araki, H. Ding, and G. Neubig, "Understanding and improving zero-shot multi-hop reasoning in generative question answering," *Proceedings of the 29th International Conference on Computational Linguistics*, p. 1765–1775, 2022.

[2] A. Talmor and J. Berant, "The web as a knowledge-base for answering complex questions," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, p. 641–651, 2018.

[3] The web as a knowledge-base for answering complex questions. Accessed on May 7, 2023. [Online]. Available: https://www.semanticscholar.org/paper/The-Web-as-a-Knowledge-Base-for-Answering-Complex-Talmor-Berant/c8725f13be7434b69738491c66b45c9225258253

[4] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

[5] Hotpotqa: A dataset for diverse, explainable multi-hop question answering. Accessed on May 7, 2023. [Online]. Available: https://www.semanticscholar.org/paper/HotpotQA%3A-A-Dataset-for-Diverse%2C-Explainable-Yang-Qi/22655979df781d222eaf812b0d325fa9adf11594

[6] T. Schick, S. Udupa, and H. Schütze, "Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp," *Transactions of the Association for Computational Linguistics*, no. 9, pp. 1408–1424, 2021.

[7] Z. Jiang, J. Araki, H. Ding, and G. Neubig. Github repository for the paper "understanding and improving zero-shot multi-hop reasoning in generative question answering". Accessed on May 7, 2023. [Online]. Available: https://github.com/jzbjyb/multihop