The Knowledge Alignment Problem: Bridging Human and External Knowledge for Large Language Models

Anonymous ACL submission

Abstract

Large language models often necessitate grounding on external knowledge to generate faithful and reliable answers. Yet even with the correct groundings in the reference, they can ignore them and rely on wrong groundings or their inherent biases to hallucinate when users, being largely unaware of the specifics of the stored information, pose questions that might not directly correlate with the retrieved groundings. In this work, we formulate this knowledge alignment problem and introduce MixAlign, a framework that interacts with both the human user and the knowledge base to obtain and integrate clarifications on how the user question relates to the stored information. MixAlign employs a language model to achieve automatic knowledge alignment and, if necessary, further enhances this alignment through human user clarifications. Experimental results highlight the crucial role of knowledge alignment in boosting model performance and mitigating hallucination, with improvements noted up to 22.2% and 27.1% respectively. We also demonstrate the effectiveness of MixAlign in improving knowledge alignment by producing high-quality, user-centered clarifications.

1 Introduction

Despite the recent advances of large language models (LLMs), they still struggle in unfamiliar scenarios not covered during pre-training (Bubeck et al., 2023). A common approach to mitigate this issue involves retrieving and incorporating supporting evidence from an external knowledge base (Guu et al., 2020; Shuster et al., 2021). While the method indeed often improves the end-task performance, it still suffers from issues such as generating text that includes extraneous information not present in the retrieved knowledge (Dziri et al., 2022), ignoring the knowledge entirely (Krishna et al., 2021), or even contradicting the knowledge (Longpre et al., 2021). These erroneous behaviors are interpreted as a passive defense mechanism against poor retrievals (Gao et al., 2022).

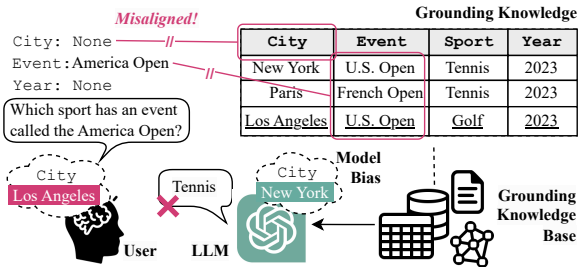

Figure 1: Knowledge Misalignment. Even if the user knows about the city constraint, he/she may not put it in the question being unaware that "city" is needed to filter noisy candidates. For the same reason, the user may not give a precise event name. Due to the misalignment between the user question and the grounding knowledge, the LLM fails to correlate the question with the correct grounding (underlined) and relies on its own biased knowledge (presuming New York as the intended city reference) to generate an incorrect answer.

In this work, we argue that the primary cause of the error cases stems from the misalignment between human and grounding knowledge. This misalignment is quite common, as users are often unfamiliar with the information contained in the external database. When framing their questions, they might unintentionally phrase them in ways that either inconsistently state or even overlook the conditions and constraints of the retrieved groundings (refer to Fig. 1). Facing this, the language model may follow spurious correlations and incorporate biased model knowledge to generate biased, misleading, or unsupported content.

To address this issue, we study the *Knowledge Alignment* problem considering various alignment types as depicted in Table 1. Unlike recent works on value alignment which aim to ensure LLM generation follows human values, ethics, and goals (Ouyang et al., 2022; Pyatkin et al., 2022), knowledge alignment seeks to bridge human and

grounding knowledge for LLM, thereby enhancing its ability to utilize grounding knowledge for faithful decision-making and issue resolution.

Towards solving the knowledge alignment problem, we propose MixAlign, a framework that interacts with both the user and the knowledge base to acquire clarifications on how the user's question relates to the stored grounding knowledge. MixAlign initiates the process with *model-based knowledge alignment*, where the LLM is employed to map the conditions and constraints of the user question to corresponding ones within the knowledge base. In cases the mapping process yields uncertainties or the evidence remains unclear, MixAlign generates a question seeking further clarification from the user, a step we refer to as *human-assisted knowledge alignment*. The clarifications from these steps are incorporated to generate the final answer.

In summary, our major contributions are:

- We study the Knowledge Alignment problem, a prevalent yet critical issue that influences the efficacy of LLMs when interacting with external databases.
- We introduce MixAlign, a mixed-initiative clarifying framework designed to improve knowledge alignment.
- Comprehensive evaluations highlight the importance of knowledge alignment and demonstrate the effectiveness of MixAlign in generating high-quality clarifications.

2 Related Work

Alignment in Large Language Models. Recent efforts have been made to ensure that AI systems pursue goals that match human values or interests rather than unintended and undesirable goals (Ngo, 2022; Wolf et al., 2023). This issue, known as the alignment problem in LLMs, has been addressed in several ways. Reinforcement Learning from Human Feedback (RLHF) is one such approach, which fine-tunes the LLM according to the reward signals adhering to human evaluators' preferences (Ouyang et al., 2022; Bai et al., 2022). Another strategy involves in-context learning using textual prompts that are helpful, honest, and harmless (Askell et al., 2021; Rae et al., 2021). The development of interpretability techniques to scrutinize the concepts learned by networks is yet another crucial approach, with the long-term aim of detecting and rectifying misaligned goals prior to deployment (Meng et al., 2022; Burns et al., 2022). This work can be seen as a special case of interpretability methods. Unlike existing works that emphasize aligning human values with LLM behavior, we aim to align human knowledge with external domain knowledge. This knowledge alignment enhances semantic and logical consistency between human expression and the stored evidence, thereby enabling LLMs to engage in more effective reasoning and problem-solving.

Clarification Question Generation. The study of asking clarifying questions spans a wide range of tasks, including information retrieval and open-domain question answering (Rao and Daumé III, 2018; Majumder et al., 2021; Kuhn et al., 2022; Pyatkin et al., 2022). The effectiveness of these questions is often determined by information-theoretic measures such as relevance, informativeness, or utility (Rao and Daumé III, 2018, 2019; White et al., 2021). Rule-based methods have been proposed for generating clarification questions by filling manually defined templates (Wang and Li, 2021) or applying a set of syntactic transformations on ambiguous questions (Dhole, 2020). In addition to rule-based methods, neural network-based approaches have been proposed to generate more coherent questions by training text generation models (Rao and Daumé III, 2018, 2019) or utilizing state-of-the-art pre-trained large language models (Krasheninnikov et al., 2022; Kuhn et al., 2022). Most of the existing works focus on resolving ambiguities within user queries, whereas we seek clarifications on how the user question is related to the stored knowledge. Instead of requesting the user to provide more context aimlessly, we direct them on how to offer such information by concentrating on a particular constraint.

3 The Knowledge Alignment Problem

We study how aligning human knowledge with grounding knowledge, represented by conditions and constraints found in the user question (Q) and retrieved evidence (K), impacts the LLM's ability to utilize evidence for answering questions. Specifically, we address knowledge misalignments by acquiring clarifications (C), refining our prompt to: $\text{LLM}(A|Q, K) \rightarrow \text{LLM}(A|Q, K, C)$.

We commence our study with a straightforward setting and consider questions that inquire about a single, specific subject, allowing us to express

Table 1: Knowledge misalignment types. We evaluate 1,173 valid examples from our FuzzyQA dataset with an evaluation protocol based on GPT-4. The overall proportion of samples with knowledge misalignment is 79.54%. "Percentage" denotes the ratio of examples with a certain type to those with any misalignment.

Type	Explanation	Example	Percentage%
Semantic	The user might use an ambiguous term that, while ideally should map to a single item in the database, in reality can correspond to multiple attributes or values.	For "What is the best burger?", when you say "best burger", are you referring to taste, nutritional value, price, or a combination of these attributes?	41.48
Contextual	The user may have implicitly established some conditions without explicitly expressing them.	For "What is the 15th most populous city in the United States?", the statistics may vary with time, which year are you considering?	32.04
Structural	The user might have stated some conditions that are not addressed in the database.	For "Find me an American writer.", the question can not be answered when the database does not include the nationality of the writers.	56.70
Logical	The user can ask complex questions where certain conditions need to be determined before other conditions can be clarified, while the database only supports basic logits such as "and", "or" and "not".	For "Fine me for a movie directed by the singer who has won a Grammy', the question answering requires first identifying singer who have won a Grammy, and then finding the films directed by the identified singers.	5.57

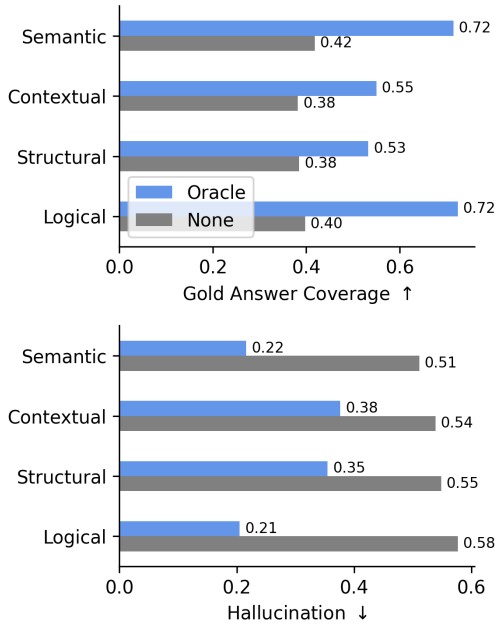

Figure 2: Oracle clarification results regarding different knowledge alignment types.

conditions and constraints in the form of attribute-value pairs related to that subject. For the representation of evidence, we opt for tabular databases due to their inherent clarity and structured nature. Each row in these databases encapsulates well-defined constraints. While other knowledge formats, such as triplets in knowledge graphs or textual paragraphs, can also be organized in this manner, we leave them for future research.

As shown in Table 1, we consider four misalignment types that correspond to conditions with different expressions (semantic), conditions absent in either the user question (contextual) or domain knowledge base (structural), and complex conditions composed of multiple simple conditions (logical), respectively.

To evaluate the importance of knowledge alignment in enhancing language model performance, we conducted experiments using oracle clarifications on different alignment types (refer to Section 4 for detailed settings). As depicted in Fig. 2, we observe a notable difference in performance across different knowledge alignment types, with a marginal gap ranging from 15% to 32% in gold answer coverage and 16% to 37% in hallucination.

Among the alignment types, we find that semantic and logical alignment exhibit a larger performance gap compared to the other two types. The semantic and logical alignment share a common characteristic: they prioritize the analysis of existing conditions and constraints rather than requesting the integration of additional, unmentioned information. This distinction is primarily driven by the nature of our dataset, where questions are typically answerable within the question itself. In practical scenarios, however, it is common that individuals who are unfamiliar with the domain may require additional contextual information, while those who are familiar with the domain may not.

4 Methodology

In this section, we introduce MixAlign, a method designed to enhance knowledge alignment in grounded generation tasks. MixAlign utilizes the

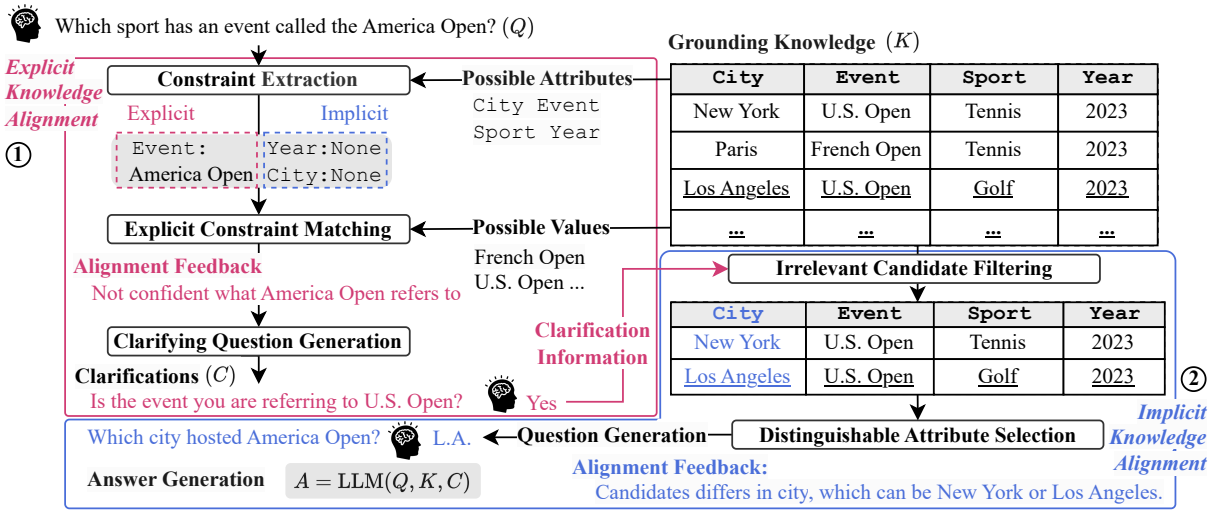

Figure 3: Diagram of MixAlign. MixAlign aims to identify knowledge misalignments and obtain clarifications regarding them automatically. It first handles explicit constraints in the user's question for semantic and logical alignment (Explicit Knowledge Alignment), then tackles implicit or missing constraints for contextual and structural alignment (Implicit Knowledge Alignment). Given the user question, MixAlign utilizes LLM to extract and correlate the constraints within the question referring to the grounding knowledge. If the model cannot confidently establish alignment, a clarifying question is generated to seek assistance from the user. The alignment information is then incorporated to filter candidate knowledge groundings. If confusion persists, the LLM is employed to select an attribute that can distinguish the remaining groundings and seek further clarification from the user.

LLM to align user expressions with the grounding knowledge and, if necessary, further enhances this alignment through human clarification. Fig. 3 depict how MixAlign tackles knowledge alignment in a two-stage manner. We illustrate the explicit knowledge alignment in Section 4.1 and implicit knowledge alignment in Section 4.2.

4.1 Explicit Knowledge Alignment

In this stage, we align constraints that are explicitly stated in the user question with those from grounding knowledge and obtain clarifications. As detailed in Section 3, constraints are represented as attribute-value pairs. Considering the potentially vast number of grounding knowledge constraints, we employ a two-step approach that first extracts values from user questions based on attributes from the grounding knowledge, and then matches values for those valid attributes. In cases where the model is uncertain about the correlation of a particular constraint, we engage the user by posing a question to confirm the alignment. This interactive step ensures the accuracy and reliability of the alignment process. Specifically, we have:

Step 1: *Constraint Extraction*. Given attributes from the grounding knowledge, extract the corresponding values from the question.

Step 2: *Explicit Constraint Matching*. Find values in the grounding knowledge that are correlated or coreference with the extracted constraints with valid values.

Step 3: *Clarification Question Generation*. Generate a question to clarify any misunderstanding the model couldn't resolve.

Note that the extracted constraint for an attribute can be a phrase, e.g., hometown: the 15th most populous city in the United States. In this case, the constraint extraction module can be seen as a question decomposer, and constraint alignment as a subquestion solver.

All steps are implemented by prompting the LLM. For step 1, we prompt the LLM with the following instruction:

> *Extract any phrases that act as conditions or constraints relating to each attribute. If you are not confident that there's an applicable phrase, signify this with 'None'.*
> *Attributes: City, Event, Sport, Year*
> *Question: Which sport has event called America Open?*

To address the issue of attributes with less semantic names, such as "Name-1", we employ the LLM to describe the attribute using its possible values before utilizing it. The prompt is shown in Appendix B.1. This approach helps provide more context and understanding to both the model and the user.

We proceed by verifying the value references in the grounding knowledge (Step 2). For each explicit constraint, we prompt the LLM as follows:

> *Question: Which sport has event called America Open?*
> *For "America Open" in the question, identify the corresponding option it refers to. If there is ambiguity or uncertainty, where multiple options seem equally probable, or no options clearly match, respond with "None".*
> *Options: French Open, U.S. Open.*

We collect the results as alignment feedback. For matches deemed successful, we categorize them as explicit clarifications, e.g., "For 'event', America Open refers to U.S. Open." In cases the model is uncertain about the correlation, we engage the user by posing a question to seek alignment (Step 3):

> *Question: Which sport has event called America Open?*
> *The constraint "America Open" in the user question is unclear. Ask a clarifying question to make the user confirm the corresponding value of the constraint. The constraint can refer to values in this list: French Open, U.S. Open*

The clarifying question and the user response obtained from this interaction serve as explicit clarification, e.g., "Question: Is the event you are referring to U.S. Open? Answer: Yes."

4.2 Implicit Knowledge Alignment

At this stage, we assess the need to address implicit constraints not stated in the user's question. If needed, we identify an attribute that optimally distinguishes candidates and pose a question to assist the user in resolving any potential ambiguities or inconsistencies. This stage comprises three steps:

Step 1 *Irrelevant Candidate Filtering*. Filter candidate groundings with previously obtained clarifications.

Step 2 *Distinguishable Attribute Selection*. Identify the optimal attribute to differentiate knowledge groundings.

Step 3 *Clarifying Question Generation*. If a valid attribute is found, generate a clarifying question regarding it.

We begin by determining the necessity to address implicit constraints. As elaborated in Section 3, we focus on questions that pertain to a singular, distinct subject. Given that the accurate grounding knowledge should be unique, we filter candidates and seek further clarification if multiple candidates remain. We set aside more complex scenarios for future exploration. Specifically, we prompt the LLM to filter the candidates (Step 1):

> *In the context of the given question and its clarifying information, filter the list of candidates. The aim is to select only those candidates that adhere to the conditions or constraints provided.*
> *Candidates:*
> *1. City: New York; Event: U.S. Open; Year: 2023;*
> *2. City: Paris; Event: French Open; Year: 2023;*
> *3. City: Los Angeles; Event: U.S. Open; Year: 2023;*
> *4. . . .*
> *Clarifying information:*
> *Question: Is the event you are referring to U.S. Open?*
> *Answer: Yes.*

In Step 2, we select the attribute by taking into account two aspects: (1) Distinguishability: We aim to eliminate noisy candidates as much as possible after clarification. (2) Answerability: We avoid asking the user about unfamiliar attributes such as names and ID numbers. For simplicity, we merge Step 2 and 3 and prompt LLM with:

> *Given the following candidates, your task is to formulate a clarifying question to filter out irrelevant candidates. This clarifying question should aim to ascertain the value of an attribute to best differentiate among candidates. Ensure that the attribute relates to general knowledge rather than specialized knowledge.*
> *Candidates:*
> *1. City: New Yrok; Event: U.S. Open; Year: 2023;*
> *2. City: Los Angeles; Event: U.S. Open; Year: 2023;*

The clarifying question and user response act as implicit clarifications, e.g., "Question: Which city hosted America Open? Answer: L.A."

4.3 Answer Generation

The final answer is generated by including explicit and implicit (if any) clarifications (C) in the prompt, i.e., $\text{LLM}(A|Q, K, C)$.

4.4 A Casual Look at MixAlign

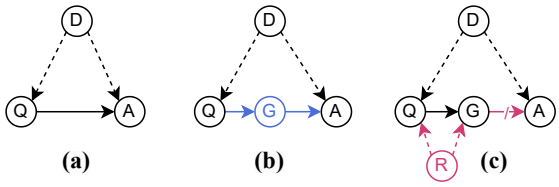

Figure 4: Knowledge grounding effectively boosts LLM performance (through front-door adjustment) only when the knowledge is causally retrieved and can causally induce the answer. That is, the retrieval method itself should be trustworthy enough to not introduce statistical co-occurrence information (i.e., a nurse must be a woman), and the retrieved knowledge must be aligned with the question in order to be utilized for further deducing the answer.

To uncover the cause-effect relationships in retrieval-augmented generation, we have developed

a Structural Causal Model (SCM) (Peters et al., 2017). SCM is a directed acyclic graph that represents causal connections within a system.

As shown in Fig. 4(a), the pre-trained knowledge (D) in LLM introduces confounding factors into the system. For example, the model may assume that a nurse must be a woman, resulting in biased correlations and ultimately harm model performance. As illustrated in Fig. 4(b), the Retrieval-augmented Language Model mitigates biased correlations through the front-door adjustment (Pearl, 2009), which employs a mediator (G, retrieved knowledge groundings) to block all directed paths from the cause (Q) to the effect (A). However, as depicted in Fig. 4(c), the front-door adjustment can easily fail when the groundings are statistically retrieved using the nearest neighbors search based on co-occurrence information. To address the aforementioned issue, MixAlign offers clear explanations on *why the question and knowledge are related*, thereby promoting front-door adjustments and boost model performance.

5 Experiments

Evaluation Task: We focus on knowledge-augmented generation instead of evidence retrieval to explore the benefits of knowledge alignment. Specifically, we consider a controlled number of irrelevant knowledge groundings (database rows) with the primary grounding in the model's input context. This count of irrelevant groundings is denoted as 'Irrelevant Groundings (#)'.

Dataset: FuzzyQA is an evolution of the OTT-QA dataset (Chen et al., 2020). OTT-QA is an English dataset that contains open questions that require grounding on tables and text for answers. In FuzzyQA, we made two changes:

1. We shifted the focus solely to tables as the primary knowledge source (detailed in Section 3). This results in a filtered dataset comprising 1,173 (question, answer, table) triples, reserved solely for validation (our method does not require training).

2. We simplify each question with GPT-4 by dropping constraints but ensuring the answers remain unchanged, as detailed in Appendix B.2. This adjustment was made because OTT-QA questions were crafted by annotators who had prior access to the tables, a scenario that contrasts with real-world situations where users often frame their queries without detailed table knowledge. By simplifying the questions, we aim to simulate this real-world ambiguity. Note that this simplification makes the questions more challenging for LLMs, as the reduced detail introduces extra complexity.

Table 2: Oracle clarifying information for each knowledge alignment type. A and B denote human and domain knowledge, respectively.

Type	Clarifying Information
Semantic	The term 'A' in the question refers to 'B' in our database.
Contextual	The value for the missing contextual condition in the question is 'A'.
Structural	The value for the condition 'B' in the database is 'A'.
Logical	The complex condition 'A' in your question refers to the condition 'B' in the database.

Language Model and Baselines. While our framework is designed to be compatible with any LLM architecture, in our experiments, we employ the OpenAI Text-DaVinci-003 (176B) (Ouyang et al., 2022) for all the generation methods. We examine the impact of incorporating clarifications into the prompt, and consider existing clarifying baselines:

• *None*. No clarification included.

• *Oracle*. We reverse-generate the oracle clarifications from the ground-truth answer and knowledge considering the templates in Table 2 with GPT-4. The prompt is shown in Appendix B.3.

• *Direct-Ask* (Kuhn et al., 2022). Clarifying questions are asked based solely on the original question. Direct-Ask prepends the question with a prompt: "In order to answer this question, I have to ask the following clarifying question:".

• *Knowledge-Ask*. Building upon Direct-Ask, we incorporate candidate knowledge to generate clarifying questions and modify the instruction as "In order to answer this question with the context, candidate 1, candidate 2, . . . , I have to ask the following clarifying question:".

• *MixAlign (proposed)*. In addition to the previous settings, we introduce alignment feedback to enhance the process of posing clarifying questions.

Metrics. We adopt G-EVAL (Liu et al., 2023), a state-of-the-art ChatGPT-based evaluation framework, and consider the three metrics below. Please refer to Appendix C for details.

• *Gold Answer Coverage*. This binary metric evaluates whether the model's answer covers the gold answer, indicating how accurately the model captures the relevant information.

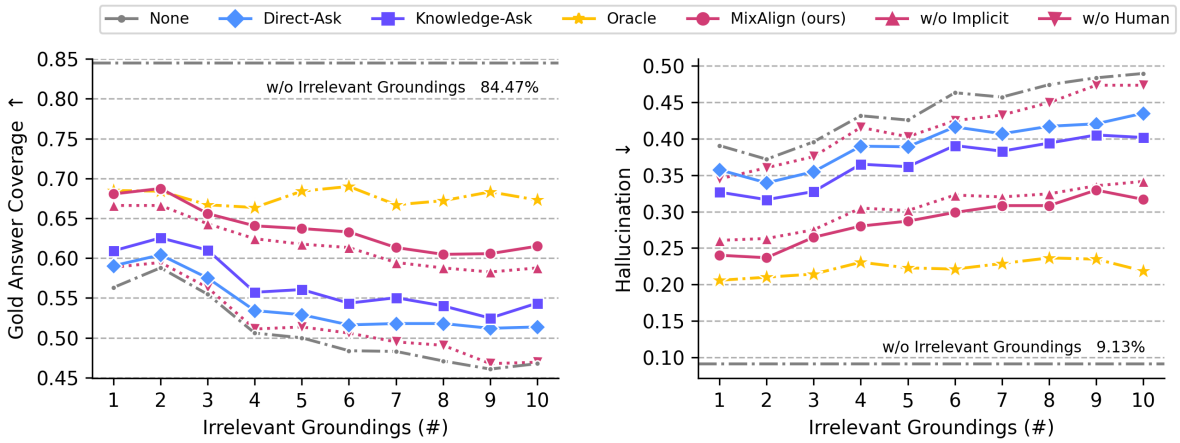

Figure 5: Overall evaluation results. We fused irrelevant knowledge groundings with ground-truth evidence for LLM generation. "w/o Implicit" and "w/o Human" represent two ablation variants of MixAlign (detailed in Sec. 5.2.) "w/o Irrelevant Groundings" denotes that the LLM is solely reliant on the ground-truth evidence, absent of noise. We report the mean from one run over the FuzzyQA dataset. OTT-QA results are in Appendix A.1.

• *Hallucination*. This binary indicator detects factual inconsistencies between the model's answer and the input context, highlighting instances where the model generates unsupported information.

• *Accepted*. This binary indicator checks whether the clarifying question posed by the model (partially) repeats the original question back to the user.

User Simulator. Following Kuhn et al. (2022), we implement the user simulator as an "oracle" language model that has access to attributions about the gold answer subject (detailed in Appendix B.4).

5.1 Overall Results

Negative impact of irrelevant groundings. Comparing "None" to "w/o irrelevant groundings" in Fig. 5 highlights the negative effect of including irrelevant groundings. Specifically, coverage decreases by 25.7% to 39.4%, and hallucination by 28.1% to 39.8%. Moreover, as more irrelevant groundings are introduced, performance worsens because they become increasingly difficult for the LLM to differentiate and associate.

Knowledge alignment significantly boosts the LLM performance. With oracle clarification, we observe a noticeable gap of 9.6% to 22.2% in coverage and 16.2% to 27.1% in hallucination. While the results are promising, a significant gap remains when compared to "w/o irrelevant groundings". One key reason for this is the complexity of the prompt given to the LLM. In the absence of irrelevant groundings, the LLM's prompt contains only a single ground-truth grounding. However, with clarification, the prompt is populated with multiple clarifying details and all available groundings. This increased complexity poses an inherent challenge for the LLM to process effectively.

Table 3: Average acceptance rate (%) for clarifying questions across varied irrelevant groundings (#).

Direct-Ask	Knowledge-Ask	MixAlign
36.09 ± 0.95	27.67 ± 0.52	100

MixAlign outperforms baseline clarifying methods. When benchmarking MixAlign against Direct-Ask, we note a difference of 8.1% to 11.7% in coverage and 9% to 11.8% in hallucination. In comparison with Knowledge-Ask, the disparity is 4.6% to 8.9% in coverage and 6.3% to 9.2% in hallucination. The key driver behind this enhancement is MixAlign's feedback-driven clarifying question generation. This approach ensures MixAlign clarifies unaligned content instead of seeking generic clarifications and echoing (in part or whole) the initial question back to the user, which is typically unhelpful and unanswerable. This is further illustrated in Table 3, where the acceptance rate for baseline methods lags significantly behind MixAlign. Specifically, Direct-Ask tends to mirror the original question, while Knowledge-Ask tends to focus on the compositional gap of the original query and pose the subquestion back to the user. Please refer to Table 4 for more error cases.

Remarkably, the performance of MixAlign is nearly on par with that of oracle clarification when the number of irrelevant groundings is minimal, as a limited number of candidates makes it easier for

Table 4: Case clarifying questions. Baseline methods may: 1. inquire about irrelevant factors that do not aid problem-solving, 2. reflect the question back to the user, and 3. incorporate noisy knowledge into the question.

User question	What are the campaign notes of the 2012 Democratic candidate born on May 23, 1958?	Where did the Magic: The Gathering player known as "The Great One" win?
Direct-Ask	What specific information are you looking for regarding the 2012 Democratic candidate?	What format of Magic: The Gathering was "The Great One" playing?
Knowledge-Ask	Which candidate was born on May 23, 1958?	Which season did **Jon Finkel** win?
MixAlign (ours)	Is the candidate you are referring to Keith Judd?	Which season did "The Great One" win?

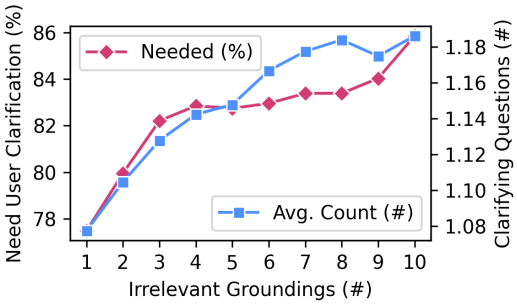

Figure 6: Efficiency analysis results of MixAlign. We target: 1. Needed (%): How often do we need to request user clarification after the initial LLM-based constraint matching? 2. Avg.Count (#): When needed, how many clarifying questions, on average, are posed?

the LLM to achieve precise constraint matching. But with more candidates, distinguishing between them becomes more challenging, causing the performance gap to widen to as much as 7.7% in coverage and 9.8% in hallucination. We will further investigate this in the following ablation study.

5.2 Ablation Study on MixAlign

We consider two ablation variants of MixAlign: 1. w/o Implicit: This version excludes implicit knowledge alignment and only addresses constraints that are explicitly stated in the user's question. 2. w/o Human: This version relies solely on LLM-based constraint matching and does not incorporate additional human-assisted clarification. The results are merged into Fig. 5. Our key findings are:

Human assistance is exceptionally vital. For the w/o Human version, we note a significant reduction in performance when solely depending on LLM-based constraint matching. As the number of irrelevant groundings increases, this reduction becomes more noticeable. This suggests that the LLM struggles to extract and match constraints with high confidence. Further evidence of this is seen in the increasing difference between the w/o Human and w/o Implicit versions, emphasizing that more constraints are not confidently matched, leading to the need for human clarification.

Implicit knowledge alignment is necessary. We see that removing implicit knowledge alignment consistently leads to reduced performance. Comparing w/o Implicit to MixAlign, we also observe that the performance gap remains largely constant, irrespective of the increase in irrelevant groundings. This is attributed to error propagation from the explicit knowledge alignment, as implicit alignment targets only the groundings remaining after explicit alignment. Future work could consider an end-to-end approach to mitigate this limitation.

5.3 Efficiency Analysis of MixAlign

In this section, we assess the user effort required by MixAlign for clarifying misalignments. As shown in Fig. 6, as the number of irrelevant groundings increases, there's a rise in both the percentage of examples requiring user input and the average number of questions asked. However, MixAlign successfully reduces the need for user clarifications by 14.2 to 22.5%. Furthermore, with an average question count spanning from 1.08 to 1.19, showing that usually just one single clarifying question is needed, which verifies the efficiency. Please refer to Appendix A.2 for full results.

6 Conclusion and Future Work

In this work, we introduce the *Knowledge Alignment* problem, which addresses mismatches between constraints present in user questions and the knowledge groundings referred to by LLMs, and we propose the MixAlign framework to bridge this gap by generating clarifications for any identified misalignments. Experimental results highlight the crucial role of knowledge alignment in improving model performance and faithfulness and demonstrate the efficacy of MixAlign in generating high-quality clarifications. Future studies could explore adapting MixAlign to various knowledge forms and modalities, thereby broadening its applicability.

7 Limitations

External knowledge extends beyond simple tabular databases or textual formats, often manifesting with more complex modularity which combines different elements together. Addressing these intricate configurations poses a formidable challenge, and we outline this as an area for future exploration.

A further limitation to consider is the increased computational load and time consumption associated with the additional clarification steps. We mitigate this in our study by involving model-based alignment and avoiding multi-turn dialogues for human-assisted alignment. Nevertheless, more efficient strategies for addressing this concern could be further investigated.

References

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Wang, and William W Cohen. 2020. Open question answering over tables and text. *arXiv preprint arXiv:2010.10439*.

Kaustubh D Dhole. 2020. Resolving intent ambiguities by retrieving discriminative clarifying questions. *ArXiv preprint*, abs/2008.07559.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2022. Rarr: Researching and revising what language models say, using language models. *ArXiv preprint*, abs/2210.08726.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Dmitrii Krasheninnikov, Egor Krasheninnikov, and David Krueger. 2022. Assistance with large language models. In *NeurIPS ML Safety Workshop*.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4940–4957, Online. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Clam: Selective clarification for ambiguous questions with large language models. *ArXiv preprint*, abs/2212.07769.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bodhisattwa Prasad Majumder, Sudha Rao, Michel Galley, and Julian McAuley. 2021. Ask what's missing and what's useful: Improving clarification question generation using global knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4300–4312, Online. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Richard Ngo. 2022. The alignment problem from a deep learning perspective. *arXiv preprint arXiv:2209.00626*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang,

Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Judea Pearl. 2009. *Causality*. Cambridge university press.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

Valentina Pyatkin, Jena D Hwang, Vivek Srikumar, Ximing Lu, Liwei Jiang, Yejin Choi, and Chandra Bhagavatula. 2022. Reinforced clarification question generation with defeasibility rewards for disambiguating social and moral situations. *ArXiv preprint*, abs/2212.10409.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2737–2746, Melbourne, Australia. Association for Computational Linguistics.

Sudha Rao and Hal Daumé III. 2019. Answer-based Adversarial Training for Generating Clarification Questions. 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 143–155, Minneapolis, Minnesota. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jian Wang and Wenjie Li. 2021. Template-guided clarifying question generation for web search clarification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3468–3472.

Julia White, Gabriel Poesia, Robert Hawkins, Dorsa Sadigh, and Noah Goodman. 2021. Open-domain clarification question generation without question examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 563–570, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yotam Wolf, Noam Wies, Yoav Levine, and Amnon Shashua. 2023. Fundamental limitations of alignment in large language models. *arXiv preprint arXiv:2304.11082*.

A Additional Experimental Results

A.1 Results on the OTT-QA dataset

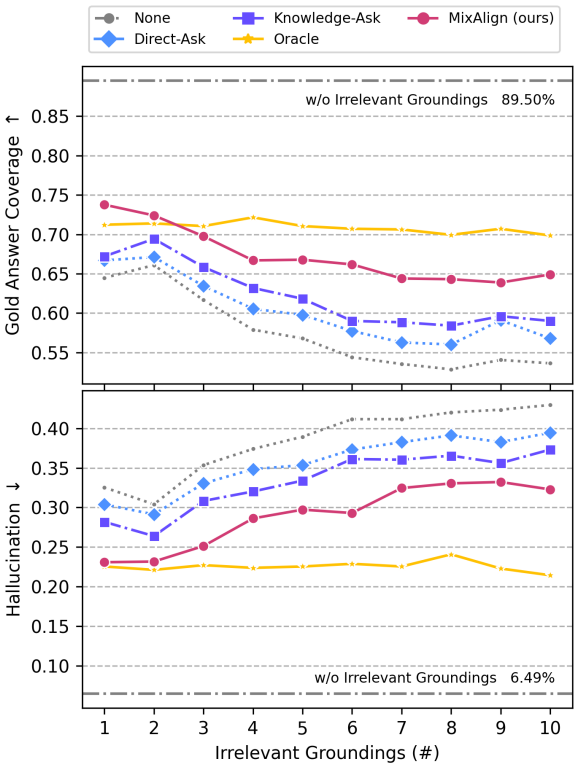

Figure 7: Overall results on OTT-QA. We report the mean over the 1,173 instances corresponding to FuzzyQA examples.

To study the impact of question simplification on performance, we further conduct experiments on the original OTT-QA questions. As depicted in Figure 7, the findings in Section 5.1 still holds, indicating that question simplification isn't the primary contributor to the observed performance discrepancy. We noticed a improvement of 5% in terms of coverage and hallucination relative to the FuzzyQA results shown in Figure 5. This supports the aforementioned explaination that question simplifications indeed pose a greater challenge for LLMs. Specifically, this question simplification particularly influencing contextual knowledge misalignment where users may omit constraints.

Suprisingly, MixAlign outperforms the oracle in terms of coverage when the number of irrelevant groundings is minimal. This phenomenon might be attributed to our decision to reuse oracle clarifications from FuzzyQA in this evaluation. While we

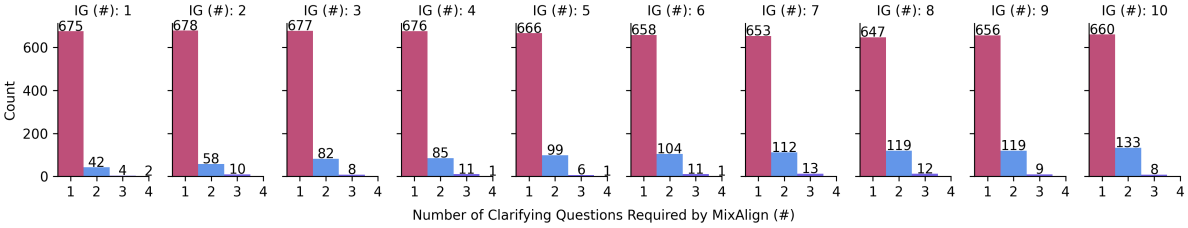

Figure 8: Distribution of clarifying questions based on the count of irrelevant groundings, denoted as IG(#).

assumed that the clarifications in OTT-QA would fall within those from FuzzyQA, the results hint at possible inconsistencies in the oracle data.

A.2 Distribution of Clarifying Questions in MixAlign

In this section, we illustrate the comprehensive distribution of clarifying questions necessitated by MixAlign, as depicted in Fig. 8. We see that the maximum number of clarifying questions posed is 4. With an increase in irrelevant groundings, there is a corresponding increment in the number of clarifying questions. Nevertheless, the distribution predominantly hovers around a single question. It is noteworthy that the surge is primarily in instances requiring two questions; however, occurrences necessitating three or four questions do not exhibit a substantial increase.

B Supplementary Details on Prompts

This section includes a presentation of various prompts referred to in the paper. GPT-3 in the title denotes that we employ Text-DaVinci-003 for generation. GPT-4 denotes the OpenAI GPT-4, the current most powerful LLM.

B.1 Attribute Description (GPT-3)

Column names along with potential values:
Note: 2021, 2022, 2023
City: New York, Los Angeles, Paris
Generate a concise phrase that accurately describes each column name. If the column names lack sufficient semantic clarity or descriptiveness, furnish them with additional context or explanations.

B.2 Question Simplification (GPT-4)

Question: Which 2010 Regional League Division 2 Southern Region team plays at the stadium with the largest capacity?
Simplify this question by dropping attributes and conditions such as time and place, make sure that the answer to the simplified question is the same as the original question.

B.3 Oracle Clarification Generation (GPT-4)

Given the question and its gold database knowledge detect if there exist misalignments regarding each type below.

Semantic Misalignment: The user might use an ambiguous term in the question that, while ideally should map to a single item in the database, in reality can correspond to multiple columns or values, leading to uncertainty about the specific item the term refers to. For instance, if a user asks "What is the best burger?", the term "best burger" could refer to different columns such as taste and price. In another case, if a user mentions "Paris", it's ambiguous whether it refers to "Paris, France" or "Paris, Texas".

Contextual Misalignment: The user may omit certain conditions in the question, making it hard to relates with the knowledge. For example, for the question "What is the 15th most populous city in the United States?", the statistics might change over time, so without specifying the year in the question, there's a misalignment with the context.

Structural Misalignment: The user could state conditions that are not covered in the database structure. For example, if a user asks "Find me an American writer", this question cannot be answered if the database does not include nationality information for writers.

Logical Misalignment: The user's question might contain intricate conditions where certain aspects need to be resolved before others can be clarified. This often occurs when a single condition in the question encapsulates other sub-conditions or questions that need to be addressed first. For example, in the query "Find me a movie directed by the singer who has won a Grammy", the identification of the Grammy-winning singer is a prerequisite before the movies directed by this person can be determined.
Question: Which sport has an event called America Open?
Knowledge: City: Los Angeles; Event: U.S. Open; Year: 2023;",

B.4 User Simulator (GPT-4)

You are a user of a QA system. You know: City: Los

Angeles; Event: U.S. Open; Year: 2023;.
You just asked 'Which sport has an event called America Open?' and the system throws back a clarifying question 'Is the America Open referring to U.S. Open?'.
Please answer the clarifying question precisely. Do not respond with anything else besides the primary subject asked by the clarifying question.

C Details on Evaluation

Please note that when we refer to GPT-4, we are specifically talking about the version gpt-4-0613, which is a snapshot taken on June 13th, 2023. This version features enhanced function calling support and has a token limit of 8,192. It includes data up until September 2021.

C.1 Coverage and Hallucination

We follow the G-EVAL framework and employ OpenAI GPT-4 for evaluation. To enhance efficiency, we also implement rule-based matching to determine whether the gold standard answer is contained within the generated response prior to the LLM-based evaluation. For hallucination, we follow the prompt in G-EVAL:

Human Evaluation of Question Answering Systems:
Factual Consistency: Does the system answer untruthful or misleading facts that are not supported by the question, context, and gold answer?
Question:Which sport has an event called America Open?
Context:City: New York; Event: U.S. Open; Year: 2023; Sport: Tennis
City: Los Angeles; Event: U.S. Open; Year: 2023; Sport: Golf.
Gold Answer: Golf
Model answer: The sport you are asking for is golf.
Does the model answer contain factual inconsistency?

For gold answer coverage, we modify the template above into the following prompt:

Human Evaluation of Question Answering Systems:
Coverage: Is the model answer consistent with the gold answer?
Gold Answer:Golf
Model answer: The sport you are asking for is golf.
Does the model answer cover the gold answer?

C.2 Accepted

A clarifying question is considered unacceptable if it either 1. echoes the original user question or a subquestion back to the user, or 2. elicits negative user responses such as "I don't know."

We identify these unsatisfactory outcomes by examining the user response. Specifically, we have compiled a list of potential answers to user questions and subquestions, along with expressions of uncertainty or lack of knowledge, such as "I don't know" or "sorry." Through rule-based matching, we assess whether a user's answer contains these phrases; if so, we categorize the corresponding clarifying question as unacceptable.

D Licensing and Terms for Datasets

In this study, we developed the FuzzyQA dataset, based on OTT-QA. Like OTT-QA, FuzzyQA adheres to the MIT License, reflecting our commitment to legal compliance and respecting OTT-QA's original terms. This licensing approach ensures transparency and aligns with legal and ethical usage standards. Our enhancements to OTT-QA for FuzzyQA align with the original dataset's intended research and academic applications