

Who’s Who: Large Language Models Meet Knowledge Conflicts in Practice

Quang Hieu Pham^{1*}, Hoang Ngo^{1*}, Anh Tuan Luu², Dat Quoc Nguyen¹

¹VinAI Research, Vietnam; ²Nanyang Technological University, Singapore

¹{v.hieupq1, v.hoangnv49, v.datnq9}@vinai.io; ²anhtuan.luu@ntu.edu.sg

Abstract

Retrieval-augmented generation (RAG) methods are viable solutions for addressing the static memory limits of pre-trained language models. Nevertheless, encountering conflicting sources of information within the retrieval context is an inevitable practical challenge. In such situations, the language models are recommended to transparently inform users about the conflicts rather than autonomously deciding what to present based on their inherent biases. To analyze how current large language models (LLMs) align with our recommendation, we introduce WhoQA, a public benchmark dataset to examine model’s behavior in knowledge conflict situations. We induce conflicts by asking about a common property among entities having the same name, resulting in questions with up to 8 distinctive answers. WhoQA evaluation set includes 5K questions across 13 Wikidata property types and 150K Wikipedia entities. Our experiments show that despite the simplicity of WhoQA questions, knowledge conflicts significantly degrades LLMs’ performance in RAG settings.

1 Introduction

Recent large-scale pretrained language models (LLMs) excel in tasks requiring natural language understanding (Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2023; OpenAI et al., 2023). However, they often "hallucinate" plausible but incorrect content due to outdated or incorrect pre-training information (Parikh et al., 2020; Wang, 2019; Dhingra et al., 2022; Luu et al., 2022). Retrieval augmented generation (RAG) methods provide contextual knowledge to address this, but conflicts within retrieval contexts are still inevitable (Chen et al., 2024; Ge et al., 2024).

According to Xie et al. (2024), under certain knowledge conflict circumstances, large lan-

*The first two authors contributed equally to this work.

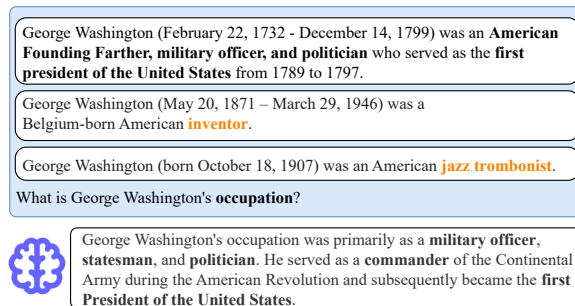


Figure 1: An example of when an LLM inherently biases an entity over another entity.

guage models (LLMs) may either ignore contextual knowledge and rely solely on their parametric memory or, more concerning, prefer certain contextual knowledge based on their popularity or specific input ordering. Figure 1 illustrates a conflict where an LLM prioritizes the popular facts about George Washington, the U.S. president, over contexts about an inventor and a jazz trombonist with the same name. This action can lead to information loss and bias, particularly when users have limited knowledge about the subject and are conducting an information-seeking query. Since eliciting LLMs parametric knowledge to handle specific cases is costly (Xie et al., 2024), we recommend that LLMs should inform their users about conflicting information, provide proper citations, and allow users to make informed decisions based on the presented evidence.

To examine if current LLMs handle knowledge conflicts transparently and accurately, we construct WhoQA, a dataset derived from Wikipedia articles and their linked Wikidata entities (Vrandečić and Krötzsch, 2014). In WhoQA, we intentionally induce conflicts by making questions asking about shared properties of Wikipedia entities with the same name. Previous studies have explored LLMs’ behavior in knowledge conflicts, either by substituting named entities (Longpre et al., 2021; Neeman

et al., 2023) or using another LLM for counterfactuals (Xie et al., 2024). These methods can generate non-existent facts, which LLMs, trained for factual consistency, may avoid. Our work mends the gap by sourcing all supporting documents from Wikipedia, a widely used and trusted source for many NLP tasks. With 2-8 different answers per question, compared to a maximum of two in previous works (Longpre et al., 2021; Neeman et al., 2023; Xie et al., 2024), WhoQA demonstrates the complication of knowledge conflicts, which are not only natural but also prevalent in practice.

Similar to solving knowledge conflicts, AmbigQA (Min et al., 2020) rewrites ambiguous user queries so that each version has only one answer. However, sometimes there can be no information to resolve the ambiguity. For example, the statements "George Washington is a farmer" and "George Washington is the first president of the United States" both refer to the same person, but can seem contradictory, if found in two separate documents without context. SituatedQA (Zhang and Choi, 2021) creates knowledge conflicts by considering similar questions in different temporal or geographical contexts. In WhoQA, where many entities share the same name, a wider range of properties can induce knowledge conflicts.

In summary, our contributions include:

- We highlight the prevalence of knowledge conflicts in practical settings and manually construct an evaluation set of 5,152 questions with supporting evidence, serving as a gold-standard benchmark.
- We perform extensive experiments using strong LLMs and show that knowledge conflicts pose a significant challenge, potentially misleading or biasing LLM users.
- We publicly release WhoQA to foster future research of knowledge conflicts in LLMs. Our WhoQA is available at: <https://github.com/VinAIResearch/WhoQA>.

2 Our WhoQA Dataset

From the set of named entities in the English Wikipedia and Wikidata dump, we group all entities that have the same name together, collect question-answer pairs as well as supporting contexts, and manually revise the evaluation set.

Take an example from Figure 1 with a set S containing three people whose names are

George Washington: one is the first President of the US, who was also a military officer; another one is a Belgium-born American inventor; and the last one is an American jazz trombonist. Given a question asking about a property $p = \text{"occupation"}$ of George Washington— $q = \text{"What is George Washington's occupation?"}$ and a set C of supporting contexts from all the three George Washington’s Wikipedia documents, a model is expected to give all answers within a set $A = \{\{\text{President of the US, military officer}\}, \{\text{inventor}\}, \{\text{jazz trombonist}\}\}$.

Formally, WhoQA is a multi-answer question answering dataset in which each question can be described as a quadruplet (q, A, S, C) , where q is a question asking about a property p of a named-entity s ; $A = \{A_i\}_{i=1}^m$ is a set of m possible distinctive answer sets to q ; $S = \{s_i\}_{i=1}^n$ is a set of n named entities that share the same name mentioned in q ; and $C = \{c_i\}_{i=1}^n$ where c_i is the corresponding supporting context for entity s_i . In a real-world RAG setup, C is obtained from a retrieval system. In this work, we focus on studying knowledge conflicts, hence each c_i is a textual context extracted from the corresponding Wikipedia document of s_i .

Question-Answer pairs collection: Starting with a set S containing named entities s_i that share the same name, we induce knowledge conflicts by creating questions about all shared properties (e.g., occupation, date of birth, and the like) among the entities. We only consider the top 60 most popular properties reported by Wikipedia. For each property, we manually construct 5 question templates and substitute the shared name among entities in S to create specific questions. An example of a template for the property $p = \text{"occupation"}$ is *"What is <shared_entity_name>'s occupation?"*. See details of these question templates in Appendix A.

For each entity s_i in the set S , the answer to a question q , created for the property p , is the corresponding set A_i in the Wikidata property triplet (s_i, p, A_i) of s_i . We only consider properties p in which the corresponding answer set A_i contains Wikidata entities, plain texts or a date-time string. If A_i is a Wikipedia entity, we take all the alternative names annotated by Wikidata. For A_i that are date-time strings, we follow the Wikipedia Manual of Style instructions to cover all of their recommended written forms. We left A_i which are plain texts in their original form. This step generates about 293K questions, not counting different

template variations, highlighting the prevalence of knowledge conflicts in practice.

Supporting context collection: For the corresponding Wikipedia document of each entity s_i in S , we search for all occurrences of elements within the corresponding set A_i and take contexts of 256 tokens around the most frequently occurring element as candidate supporting contexts. Then, we concatenate the question q with the most frequently occurring element, and use Contriever (Izacard et al., 2022) to calculate candidate contexts’ relevant score. The final score for each context is averaged over the 5 questions generated from our 5 question templates. We keep only the most relevant context, if its score is above 0.45. Otherwise, the Wikipedia document is removed from our dataset. Finally, we filter the set A_i to include only elements that appear together with the most frequent element in the top-1 context. Our simple matching process typically results in supporting contexts where the answers are directly stated.

Final Benchmark: After the filtering steps above, 76,487 questions remain, spanning 13 property types and involving 145,710 different Wikidata entities. To study the effect of property types and the number of conflicting answers on LLM performance, we undersample properties with many questions and prioritize those with many conflicting answers. For each property, we randomly select up to 500 questions, using weighted sampling where questions with more conflicting answers have a higher probability of being selected.

To ensure the reliability of our evaluation set, we hire two annotators to independently verify whether the answer elements in A_i can be inferred from the automatically collected supporting contexts. Initially, the annotators have to check 18,967 contexts for 5,592 questions. However, we allow them to exclude a question, if there is at least one incorrectly labeled context. In total, the annotators exclude 1,602 contexts together with their corresponding questions and answers. Our post checking process finally returns an evaluation set containing 5,152 questions, with 3.25 conflicting answers per question on average and $18,967 - 1,602 = 17,365$ contexts.

Figure 2 and Figure 3 demonstrates the overall statistics of our dataset with respect to the number of conflicting answers and types of Wikidata property respectively.

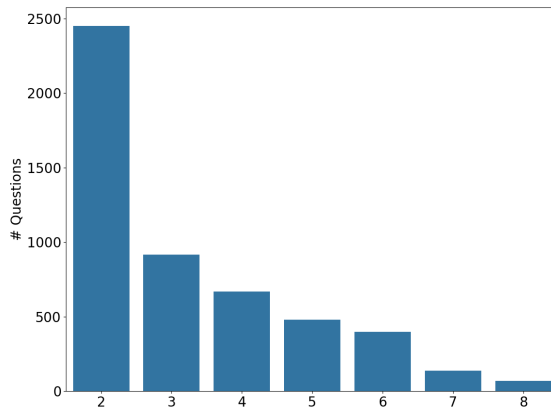


Figure 2: Number of questions distribution with respect to number of conflicting answers

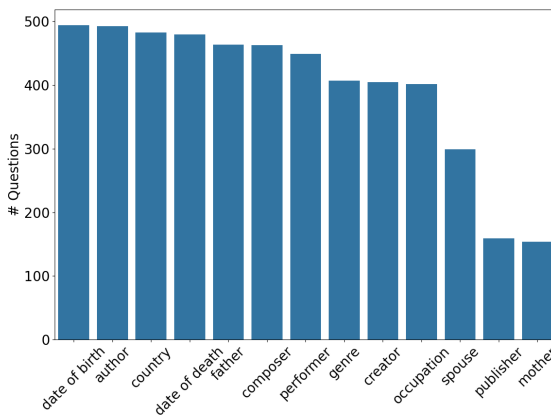


Figure 3: Number of question with respect to questions’ property types

3 Experiments

3.1 Experimental setup

We study LLMs’ behavior when dealing with knowledge conflicts in three different scenarios:

1. The first setting is a simple QA scenario where only a single context is presented to a model, hence no conflict.
2. In the second setting, we provide the model with conflicting contexts without informing it about the presence of conflicts.
3. Finally, in the third setting, we explicitly specify the presence of conflicts in the model’s input.

We evaluate 10 open source models: Gemma 7B (Team et al., 2024), Mistral 7B (Jiang et al., 2023), Mixtral 8x7B (Jiang et al., 2024), Qwen1.5 Chat (7B, 14B, 32B, 72B) (Bai et al., 2023), Command

Model	#Param	SimQA	W/oS	W/S
Gemma 1.1	7B	91.8	12.7	34.4
Mistral	7B	96.5	44.5	54.7
Qwen1.5 Chat	7B	91.9	22.0	50.2
Llama 3	8B	98.2	72.4	71.5
Qwen1.5 Chat	14B	95.5	9.90	51.9
Qwen1.5 Chat	32B	95.5	21.0	60.5
Command R	35B	92.5	69.9	-
Mixtral 8x7B	40B	97.1	53.7	64.2
Llama 3	70B	97.8	83.8	86.1
Qwen1.5 Chat	72B	95.9	28.2	66.8
gpt-3.5-turbo	-	97.9	36.0	58.9

Table 1: Macro average accuracy of all models over the number of conflicting answers. “SimQA”, “W/oS” and “W/S” denote the first setting of simple QA, the second setting of knowledge conflict without specification and the third setting of knowledge conflict with specification, respectively. Due to the chat template of Command R, it cannot be used for the third setting “W/S”.

R 35B (Cohere, 2024), and Llama 3 (8B, 70B) (Dubey et al., 2024). We also evaluate GPT-3.5 (Ouyang et al., 2022) behavior via OpenAI API. See our prompt templates for all these experimental models in Appendix B.

We serve all the open source LLMs in our experiments using vLLM (Kwon et al., 2023), with bfloat16 floating-point format on a single machine with 8 Nvidia A100 40GB GPUs. The inference process uses sampling with top_k = 50 tokens, top_p = 0.9, max_tokens = 512, and temperature = 0.9.

Accuracy evaluation metric: An answer is correct if it includes all references to the elements in the answer set supported by our collected contexts. As the Wikidata database already contains various name variations for its entities, this significantly reduces the chance of missing correct answers from the models’ responses.

3.2 Simple QA

We divide each question into n turns, each corresponding to one of the n conflicting contexts. As shown in Table 1, when conflicts are removed, most LLMs can accurately answer single-hop questions. This result highlights the ease of finding answers within the individual contexts of WhoQA, allowing our subsequent experiments to focus more on the issue of knowledge conflicts.

Model	#Param	≤ 4 Wo/S	> 4 Wo/S	≤ 4 W/S	> 4 W/S
Gemma 1.1	7B	16.5	9.80	41.7	29.0
Mistral	7B	53.0	38.1	59.9	50.7
Qwen1.5 Chat	7B	20.9	22.9	56.8	45.3
Llama 3	8B	70.6	73.7	76.2	67.9
Qwen1.5 Chat	14B	5.50	13.2	49.4	53.8
Qwen1.5 Chat	32B	11.7	28.0	61.8	59.6
Command R	35B	77.9	63.9	-	-
Mixtral 8x7B	40B	51.5	55.3	64.8	63.7
Llama 3	70B	83.8	83.8	87.9	84.7
Qwen1.5 Chat	72B	25.0	30.6	60.9	71.3
gpt-3.5-turbo	-	45.8	28.6	51.8	64.2

Table 2: Average model performance variation across different number of conflicting answers (≤ 4 or > 4) in the input context.

3.3 Knowledge conflict: without specification

We hypothesize that an ideal LLM should inherently recognize knowledge conflicts. To test this, we provide all conflicting contexts for each question and ask the LLM to answer without indicating the presence of conflicts.

Simple knowledge conflicts substantially impairs LLMs. As shown in Table 1, LLMs in our experiments exhibit significant performance drops when knowledge conflicts occur. Since finding answers from each single context in WhoQA is straightforward, this decline is attributable to knowledge conflicts. Therefore, the varying levels of performance drop indicate LLMs’ robustness against knowledge conflicts.

Without being noticed, LLMs are not sensitive to subtle knowledge conflicts. We examine how LLM performance changes as the number of conflicting answers increases. It is logical to assume that more conflicting information in the input context leads to higher information entropy. Table 2 shows that LLMs generally perform better on questions with more conflicting answers. This suggests that LLMs are less sensitive to subtle conflicts within their input contexts. Therefore, LLM practitioners should address knowledge conflicts in tasks where fine-grained answers are expected.

There is a tradeoff between helpfulness and accuracy when LLMs meet knowledge conflicts. We consider questions where LLMs miss all answers from all contexts as hard questions and manually review at most 100 hard question responses for each LLM. Table 3 shows the rate at which each model gives no answer to questions. Specifically, gpt-3.5-turbo often states that there are multiple individuals with the same name in the input contexts

Model	#Param	Rejection rate (%)
gpt-3.5-turbo	-	99
Qwen1.5-Chat	72B	8
LLama 3	70B	33
Mixtral 8x7B	40B	18
Gemma 1.1	7B	92

Table 3: Rejection to answer rate of models on their own hard evaluation set. See the full result table with all models in Appendix C.

and asks for clarification. While safe, this kind of response requires the user to read all contexts to clarify, providing no direct benefit. Gemma 7B, instead of acknowledging conflicts, simply states there is not enough information to answer, which can confuse the user. For models with low rejection rates like LLama3 70B, Qwen1.5 72B, and Mixtral 8x7B, handling conflicts is crucial. Reviewing Qwen1.5 72B responses, we find that Qwen often selects only one answer among many. This behavior is more concerning than refraining from answering, as it can cause misinformation and bias the users.

3.4 Knowledge conflict: with specification

We modify the prompt template to include few-shot examples, notifying LLMs of knowledge conflicts within their input context. Results in Table 1 show significant performance improvement for most models. The exceptions are the two LLama3 models, which show minimal benefit. A review of their responses reveals that these models are aware of conflicts and attempt to provide all available answers, regardless of whether conflicts are stated. This behavior supports our view that LLMs should be transparent and serve as tools to aid users’ decisions. Results from Table 2 suggest that questions with more conflicting answers are indeed more challenging for LLMs.

4 Conclusions

We introduce WhoQA, a high-quality benchmark dataset designed to examine language model behaviors in knowledge conflict situations. WhoQA bridges the gap with previous work by inducing knowledge conflicts without generating counterfactuals, demonstrating that conflicts can arise not only from misinformation but also naturally due to ambiguity in retrieved contexts, user questions, or similarities among entities. Through our ex-

periments, we also show that many LLMs are not sensitive to subtle conflicts within their input, and thus only simple conflicts can significantly impair LLM performance in RAG settings.

Limitations

The simplicity of our matching process to collect supporting contexts results in the fact that questions in WhoQA are mostly single-hop. We think that it could be noteworthy to propose a way to induce knowledge conflicts for multi-hop questions so as to challenge future LLMs. We still leave the question on how to make LLMs’ responses more informative in case on dealing with knowledge conflict. Possibly, looking at fine-tuning methods which control how LLMs deal with unfamiliar examples (Kang et al., 2024; Ge et al., 2024) is a promising direction.

References

- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Sheng-guang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint*, arXiv:2309.16609.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.
- Cohere. 2024. *Command r: Retrieval-augmented generation at production scale*.

- Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-Aware Language Models as Temporal Knowledge Bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, et al. 2024. The Llama 3 Herd of Models. *arXiv preprint*, arXiv:2407.21783.
- Xiou Ge, Ali Mousavi, Edouard Grave, Armand Joulin, Kun Qian, Benjamin Han, Mostafa Arefiyan, and Yunyao Li. 2024. Time Sensitive Knowledge Editing through Efficient Finetuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 583–593.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. [Unsupervised Dense Information Retrieval with Contrastive Learning](#). *Transactions on Machine Learning Research*.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint*, arXiv:2310.06825.
- Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of Experts. *arXiv preprint*, arXiv:2401.04088.
- Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar Finetuning Examples Control How Language Models Hallucinate. *arXiv preprint*, arXiv:2403.05612.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, pages 611–626.
- 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.
- Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A. Smith. 2022. Time Waits for No One! Analysis and Challenges of Temporal Misalignment. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5944–5958.
- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering Ambiguous Open-domain Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 5783–5797.
- Ella Neeman, Roei Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. 2023. DisentQA: Disentangling Parametric and Contextual Knowledge with Counterfactual Question Answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10056–10070.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, et al. 2023. GPT-4 Technical Report. *arXiv preprint*, arXiv:2303.08774.
- 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.
- Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqi, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A Controlled Table-To-Text Generation Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1186.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint*, arXiv:2403.08295.
- Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledge Base. *Communications of the ACM*, 57:78–85.
- Hongmin Wang. 2019. Revisiting Challenges in Data-to-Text Generation with Fact Grounding. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 311–322.
- Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. [Adaptive Chameleon or Stubborn Sloth: Revealing the Behavior of Large Language Models in Knowledge Conflicts](#). In *Proceedings of the 12th International Conference on Learning Representations*.
- Michael Zhang and Eunsol Choi. 2021. SituatedQA: Incorporating Extra-Linguistic Contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387.

A Question templates

We include in Table 4 all the 65 question templates for the final 13 Wikidata properties in our final dataset.

B Prompt templates

We report the prompt templates used in our experiments in Table 5.

C Models rejection rate

We consider questions where LLMs miss all answers from all contexts as hard questions and manually review 100 hard question responses for each LLM. Table 6 shows the rate at which each model gives no answer to questions.

Property	Question template
date of birth	<ol style="list-style-type: none"> 1. What is the date of birth of [subject]? 2. When was [subject] born? 3. What is the birthdate of [subject]? 4. What is the date of [subject]'s birth? 5. When did [subject] come into this world?
author	<ol style="list-style-type: none"> 1. What is the author of [subject]? 2. Who created [subject]? 3. What is the name of the main creator of [subject]? 4. Who wrote [subject]? 5. Can you provide the authorship of [subject]?
country	<ol style="list-style-type: none"> 1. What is the country of origin for [subject]? 2. Which country does [subject] reside in? 3. Can you tell me the country where [subject] is located? 4. What is the sovereign state where [subject] is situated? 5. What country does [subject] belong to?
date of death	<ol style="list-style-type: none"> 1. What is the date of death of [subject]? 2. When did [subject] pass away? 3. What was the date of [subject]'s demise? 4. When did [subject] die? 5. Can you tell me the date of [subject]'s passing?
farther	<ol style="list-style-type: none"> 1. Can you tell me who is the farther of [subject]? 2. Who is the father of [subject]? 3. What is [subject]'s farther name? 4. Could you give me the name of [subject]'s farther? 5. Tell me who is [subject]'s farther?
composer	<ol style="list-style-type: none"> 1. Who is the composer of [subject]'s music? 2. Can you tell me who wrote the music for [subject]? 3. Who are the individuals responsible for composing [subject]'s music? 4. Could you specify the composer of [subject]? 5. Who compose the music for [subject]?
performer	<ol style="list-style-type: none"> 1. Who is the performer associated with [subject]? 2. Could you please identify the individual or group acting as the performer for [subject]? 3. Who takes on the role of performer of [subject]? 4. Who is the performer who perform [subject]? 5. Who performs [subject]?
genre	<ol style="list-style-type: none"> 1. What genre does [subject] belong to? 2. Could you specify the genre of [subject]? 3. What genre would [subject] be classified as? 4. Which genre would you classify [subject] into? 5. Which genre would [subject] be classified into?
creator	<ol style="list-style-type: none"> 1. Who is attributed as the creator of the [subject]? 2. By whom was the [subject] created? 3. Who is responsible for making the [subject]? 4. Who can be identified as the creator of the [subject]? 5. What individual or entity is credited with creating the [subject]?
occupation	<ol style="list-style-type: none"> 1. What is the [subject]'s occupation? 2. Could you tell me what [subject] does for a living? 3. What profession or job is associated with [subject]? 4. What profession does [subject] work in? 5. What role or position does [subject] hold professionally?
spouse	<ol style="list-style-type: none"> 1. Can you tell me who [subject]'s spouse is? 2. Who is married to [subject]? 3. What is the name of [subject]'s spouse? 4. Who is [subject] spouse? 5. Who is the spouse of [subject]?
publisher	<ol style="list-style-type: none"> 1. Who is the publisher of the [subject]? 2. Which organization or person is responsible for publishing the [subject]? 3. Who is responsible for bringing the [subject] to the public? 4. Can you name the publisher of the [subject]? 5. Is there any information on who published the [subject]?
mother	<ol style="list-style-type: none"> 1. Can you tell me who is the mother of [subject]? 2. Who is the mother of [subject]? 3. What is [subject]'s mother name? 4. Could you give me the name of [subject]'s mother? 5. Tell me who is [subject]'s mother?

Table 4: Question templates for properties in WhoQA, sorted descending based on the number of question per each property in the dataset. Here [subject] denotes the <shared_entity_name> within a set of entities S .

Setting	Prompt template
Conflict without specification	<p>Given the following documents, answer the question under the ### QUESTION section. Give short answer only.</p> <p>### DOCUMENTS: {docs}</p> <p>### QUESTION: {question}</p> <p>### RESPONSE:</p>
Conflict with specification	<p>### INSTRUCTIONS</p> <ol style="list-style-type: none"> 1. You are given some DOCUMENTS that are relevant to answer a QUESTION 2. Read the DOCUMENTS, think step-by-step and give your final answer(s) to the question. 3. Give your answer(s) as a list with each item starting with "-". Do not include any other formatting. In your answer(s), give only the necessary information in a concise format. Here are some examples for you to learn from. <p>Example 1:</p> <p>### DOCUMENTS: Doc #0: Foo was written by Bar</p> <p>Doc #1: Foo was written by Boo</p> <p>### QUESTION: Who is the author of Foo?</p> <p>### RESPONSE: - Bar (Doc #0) - Boo (Doc #1)</p> <p>Example 2:</p> <p>### DOCUMENTS: Doc #0: Bar met Foo in [year1]</p> <p>Doc #1: Foo met Bar in [year2]</p> <p>### QUESTION: When did Foo meet Bar?</p> <p>### RESPONSE: - [year1] (Doc #0) - [year2] (Doc #1)</p> <p>Your task:</p> <p>### DOCUMENTS: {docs}</p> <p>### QUESTION: {question}</p> <p>### RESPONSE: 10150</p>

Table 5: Prompt templates.

Model	#Param	Rejection rate (%)
gpt-3.5-turbo	-	99
Qwen1.5-Chat	72B	8
LLama 3	70B	33
Mixtral 8x7B	40B	18
Qwen1.5-Chat	32B	2
Qwen1.5-Chat	14B	0
Qwen1.5-Chat	7B	0
Mistral	7B	47
Llama 3	8B	23
Gemma 1.1	7B	92

Table 6: Rejection to answer rate of models on their own hard evaluation set.