Ragability Benchmark: A Dataset and Library to Test LLMs on Inter-context Conflicts

Stephanie Gross, Johann Petrak, Brigitte Krenn
Austrian Research Institute for Artificial Intelligence
Freyung 6/6, 1010 Vienna
stephanie.gross@ofai.at, brigitte.krenn@ofai.at, johann.petrak@ofai.at

Abstract

Knowledge conflicts are a challenging issue when applying retrieval augmented generation (RAG) systems. In this paper, we propose a benchmark to test LLMs on how they deal with inter-context knowledge conflicts where implicit reasoning is required to solve the conflict. Based on actual empirical examples, real entities are replaced by fantasy entities to make sure the model's internal knowledge does not influence how the model deals with external conflicting information. The proposed benchmark can be used to assess current up-to-date LLMs, but it can also flexibly be adapted for in-depth evaluation of a specific RAG system on selected aspects of conflict identification. We also present an experiment where we apply the benchmark to test 7 current LLMs from different model families. The results show that LLMs are able to identify conflicting contexts (*Is there a contradiction, yes or no?*), while they struggle with answering content related queries. Adding a hint that there might be a contradiction in the provided contexts increases the performance of conflict identification for contradictory context, while it significantly decreases the performance for non-contradictory contexts.

Keywords: benchmark, inter-context conflicts, LLMs

1. Introduction

In this paper, a benchmark (dataset and code) is presented to evaluate how LLMs deal with knowledge conflicts or missing knowledge in the context of Retrieval-Augmented Generation (RAG) systems, i.e., when context information for the LLM is provided from sources external to the LLM. For companies, RAG systems provide a means to combine company-internal data with the power of LLMs without the need for fine-tuning or re-training an existing LLM. I.e., the system searches a company's knowledge base (documents, databases, APIs, intranets, etc.) for relevant information, feeds the retrieved context into an LLM so that the LLM generates a response grounded in the data retrieved from the company's own knowledge sources. However, the querying of the knowledge sources can lead to the retrieval of documents containing conflicting information, which are then fed as context to the LLM. This may impact the performance of the LLM, leading to inconsistent and erroneous outputs (Gokul et al., 2025). Therefore, it is important to rigorously test LLMs with respect to their ability to handle contradictions.

1.1. Related Work

There are three main types of knowledge conflicts related to LLMs (Xu et al., 2024): (1) context–memory conflict, i.e., conflicts between the model's internal knowledge and external knowledge (Kasai et al., 2023); (2) inter-context conflict, i.e., conflicts among multiple input documents from the external knowledge, e.g., within a RAG system (Su et al., 2024; Lee et al., 2025); and (3) intra-memory conflict, i.e., conflicts reflecting variation in an LLM's training data (Chang and Bergen, 2024). In this paper, we focus on inter-context conflicts. The causes of knowledge conflicts also vary, including semantic conflicts caused by ambiguities and multiple meanings of words, temporal conflicts arising from knowledge/facts changing over time, and misinformation conflicts resulting from incorrect or misleading information. Su et al. (2024) generated ConflictBank, a benchmark based on 2.86M claims extracted from Wikidata to systematically evaluate the effects of knowledge conflict in retrieved knowledge, embedded knowledge, and their interactions. Their results showed, among others, that LLMs are more sensitive to temporal and semantic conflicts than to misinformation conflicts, highlighting the importance of that challenge. In our approach, we focus on misinformation conflicts stemming from categorical, numerical and temporal conflicts in the contexts.

MMKC-Bench (Jia et al., 2025) is a multimodal knowledge conflict benchmark aimed at evaluating factual knowledge conflicts of parametric knowledge within the LLM as well as conflicts within the external knowledge. It includes 1,573 textual knowledge instances and 3,381 images, collected through automated pipelines with human verification. Their results show that while current LLMs are capable of recognizing knowledge conflicts, they tend to favor internal parametric knowledge over external evidence.

Testing the ability of LLMs in detecting external knowledge conflicts is particularly important for RAG systems. Accordingly, several benchmarks

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 2323–2333
11–16 May 2026.

have been proposed addressing conflicts in context documents. These benchmarks differ in various aspects including how the conflicting contexts are created and based on which data. WikiContradict (Hou et al., 2024), for instance, comprises manually annotated conflicting documents from Wikipedia. ECON (Jiayang et al., 2024) is based on two QA datasets NaturalQuestions (Lee et al., 2019) and ComplexWebQuestions (Talmor and Berant, 2018). Whereby for each QA pair, a set of alternative answers is generated and for each answer related evidence is generated, using an LLM (llama3-70b-instruct). Conflicting contexts are constructed by selecting conflicting evidence pairs such that they support conflicting answers to the same question. The evidences are checked utilizing (i) natural language inference (NLI) checking and (ii) LLM reasoning checks. Pieces of evidence that fail the two checking steps are filtered out. Gokul et al. (2025) base their dataset on HotpotQA (Yang et al., 2018), a dataset requiring multi-hop reasoning for question answering, and use claude-3-sonnet for generating conflicting documents. Another approach is pursued for the creation of MAGIC (Lee et al., 2025) where the Wikidat5M knowledge graph (Wang et al., 2021) is the knowledge source and an LLM with strong reasoning capability is used for knowledge conflict generation. The proposed conflict datasets also differ in which types of conflicts they represent. Gokul et al. (2025) distinguish between conflicts within a single document, between a pair of documents and conditional conflicts where the presence of document C leads to a conflict between documents A and B. MAGIC addresses single- and multi-hop conflicts and distinguishes between the number of conflicts (1-4) that must be identified. WikiContradict comprises data with clearly stated (explicit) contradictions and implicit contradictions where reasoning is required to identify the conflict. ECON distinguishes between answer and factoid conflicts, i.e., whether two pieces of evidence lead to conflicting, however, true answers, or whether the facts in the contexts are contradictory (for examples see Table 1 in Jiayang et al., 2024).

1.2. Our Approach

For an in-depth analysis of an LLM's abilities to handle contradictory contexts, different combinations of contexts (such as no context, contradictory or non-contradictory contexts) and queries are presented to the LLM in various combinations and orders of context presentation. The LLM responses are evaluated in two different ways: (i) 'Yes'/'no' responses to whether specific contexts are contradictory are directly converted to calculate accuracy scores for different metrics. (ii) More verbose responses to context related queries are evaluated by a *checker LLM* and then converted to calculate accuracy scores. The software allows for flexibly connecting to different LLMs and to flexibly define which LLMs are used as *checker LLMs*.[1]

The implemented metrics focus on whether (i) the LLM refuses to answer a query if there is not enough context information provided or the provided information is contradictory; (ii) the LLM provides the correct answer if a query is answerable, i.e., the context contains enough and non-contradictory information to answer the question; (iii) the LLM is able to identify if there is a knowledge conflict in the provided context; (iv) the LLM in general provides a correct answer, which means it either refuses to answer if there is not enough information, answers correctly if there is no knowledge conflict, or identifies a knowledge conflict.

The whole pipeline for evaluating the performance of LLMs in different, contradictory context situations is implemented in a flexible way, so that, apart from the currently implemented test situations, it can be adapted to specific test requirements. This is particularly important when the benchmark is used in the application contexts of individual companies. Our main contributions:

- We present the Ragability Corpus, a handcrafted dataset for a systematic in-depth analysis of misinformation conflicts within external knowledge, as occurring in RAG systems. The instances are derived from an empirical basis (the WikiContradict dataset), whereby real entities are replaced by fantasy entities, to make sure that a query about these fantasy entities can not be answered with the model's parametric memory. The corpus is constructed in a modular way, and can be adapted to different domains and knowledge conflicts.
- We introduce the Ragability Library to conduct experiments based on the Ragability Corpus. From a small number of entries in the dataset a large number of test cases can be automatically generated. *Testee-* and *checker LLMs* can be flexibly loaded, and prompt strategies can be flexibly applied.
- We conducted an experiment to test 7 LLMs on their ability to deal with conflicting information, and present our analysis of the results.
- We release both the Ragability corpus[2] and the Ragability library[3] and provide instructions and ideas how both the corpus and the library can be extended for further application areas.

[1] https://github.com/OFAI/python-llms-wrapper

[2] https://huggingface.co/datasets/ofai/RagabilityCorpus

[3] https://github.com/OFAI/python-ragability

The paper is organized as follows: The Ragability Corpus, a dataset containing knowledge conflicts, is presented in Section 2. The further building blocks required to conduct Ragability experiments are described in Section 3. Experiments are presented in Section 4. The conclusion in Section 5 summarizes our approach and the conducted experiments. Furthermore, experiences from employing the benchmark at companies are briefly summarized.

2. The Ragability Corpus

In this section, we introduce WikiContradict, the empirical basis for the Ragability Corpus, describe the Ragability Corpus, and explain how the dataset can be extended and how a new one can be created.

2.1. The Empirical Basis: WikiContradict

WikiContradict (Hou et al., 2024) is a benchmark for evaluating LLMs on real-world knowledge conflicts from Wikipedia[4]. It consists of 253 human annotated instances that cover different types of real-world knowledge conflicts. An instance in the dataset covers a query, context1, context2, the answer to the query based on context1 (answer1), the answer to the query based on context2 (answer2), the contradiction type of the two contexts, i.e., if the contradiction is explicit or if reasoning is required to identify the contradiction (implicit contradiction), as well as additional metadata on the Wikipedia article. We chose WikiContradict as basis, because it contains (i) actual real-world knowledge conflicts, and (ii) instances where implicit reasoning is required.

In WikiContradict, 92 instances were annotated as requiring implicit reasoning. However, out of these 92 instances, there are only 51 distinct context pairs, as 41 instances of the dataset were duplicates of context pairs with additional queries. From these 51, we removed another eight context pairs, because the contradiction was not obvious to the authors of this paper.

The remaining 43 context pairs were chosen as empirical basis, because they represent real-world knowledge conflicts for the identification of which implicit reasoning is required. Thus, providing realistic challenges LLMs need to deal with.

2.2. The Ragability Corpus

A unique feature of the Ragability Corpus is its assurance of novelty of data instances. In order to make sure that the LLM is only able to answer a question if additional context is provided, e.g., via RAG, and that the data instance is not already part of the LLM's training data, the entities in the data taken from WikiContradict are replaced by fantasy entities that do not exist, dates and numbers are changed.

Out of the 43 context pairs extracted from WikiContradict, 53 Ragability Corpus entries were manually created: For some cases, the query which needs to be answered based on the contexts was varied. E.g., one query for one context pair focuses on the contradiction of how many died in a specific battle and the other focuses on how many were wounded. For some context pairs, the contexts were slightly varied by reinforcing the knowledge conflict. In the Ragability Corpus, each context 1 and context 2 pair has a unique ID. In case there are two different queries for the same context pair, they have the same ID. Each entry also contains the ID from the WikiContradict dataset which inspired the new examples. In WikiContradict, the context in some cases contains a long paragraph, while in the Ragability Corpus, a context contains 1-2 sentences (see Table 1 for sample contexts). In WikiContradict the same text snippet was used for both context 1 & context 2, if the contradiction was within a sentence. In the Ragability Corpus, however, each context can be viewed as a different snippet retrieved from a RAG system. Note that the dataset can still be extended to add further contexts to the instances. Please also note that although the corpus contains a low number of manually created examples, via the systematic combination of different contexts and queries, the LLMs to be tested need to respond to a much higher number of prompts, depending on the evaluation goal. In the experiments presented in this paper, each of the LLMs responded to 1060 individual test instances. In the following, the features representing each entry in the dataset are presented.

2.2.1. Contexts

Another unique feature of the Ragability Corpus is that its number of contexts can be extended in a systematic way, to allow for a more in-depth analysis of how well LLMs are able to deal with knowledge conflicts. In WikiContradict, there are two contexts per instance. If the contradiction occurs in the same sentence, both contexts contain the same text. In the Ragability Corpus, the number of contexts which is considered by the Ragability library is currently 4. This allows different kinds of context to be flexibly created.

context 1, context 2: these 2 contexts are derived from the WikiContradict dataset. The individual contexts are non-contradictory, but both contexts together are contradictory. In the Ragability experiments, there is a strong focus on these two contexts. To identify the contradiction, implicit reasoning is

[4] https://huggingface.co/datasets/ibm-research/Wikipedia_contradict_benchmark

Feature	Example 1	Example 2
contradiction_ID	13	22
WikiContradict_ID	42	82
reasoning_required_c1c2	categorical	temporal.numerical
c1xq	qeu	qeu
c2xq	qeu	qiu
context_1	A Cap Squirrel is a suricate.	Corale Fenger started learning to ski when she was 12 years old.
context_2	A Cap Squirrel is a squirrel.	Corale Fenger was born in 1993. She started learning to ski in 2003.
context_3_nc1_c2	A Cap squirrel is a small mongoose.	Corale Fenger was born in 1993. She started learning to ski in 2005.
context_4_nc1_nc2_nc3	A Cap squirrel is an animal.	Corale Fenger had learned how to ski.
query_text	What kind of animal is a Cap Squirrel?	How old was Corale Fenger when she started learning to ski?
answer_context1	a suricate	12 years
answer_context1_long	A Cap Squirrel is a suricate.	Corale Fenger was 12 years old when she started learning to ski.
answer_context2	a squirrel	10 years
answer_context2_long	A Cap Squirrel is a squirrel.	Corale Fenger was 10 years old when she started learning to ski.

Table 1: Two sample instances of the dataset.

required in all these cases.

context 3: this context is not contradictory to context 1 but contradictory to context 2. Context 3 is a reformulation of context 2 so that it is not contradictory to context 1 anymore, still keeping its wording and/or syntactic structure. The contradiction between context 2 and 3 is lexically overt, no implicit reasoning is required.

context 4: provides additional context and is not contradictory to any other context. It relates to the same semantic content but is superficial enough so that is does not raise a conflict.

2.2.2. Answers to the Query

The answers to the query include a knowledge conflict for context 1 and context 2 throughout the dataset. For both contexts, there is a short and a long answer to the query in the dataset. The reason why the data comprise both a short and a long answer is that a *checker LLM* is able to identify the minimal semantic content and the maximal semantic content of the response. Examples for query, answer_context1, answer_context1_long, answer_context2, answer_context2_long are presented in Table 1.

2.2.3. Tags

Via different tags, the relation between different contexts or a context and the query are made explicit in the dataset. In the column **reasoning_required**, the different types of reasoning required to compare context 1 and context 2 are annotated: **categorical, numerical, temporal.numerical** and **temporal.relational**, whereby temporal.numerical refers to date/time expressions (e.g., *2013, 21 of May, 1989/12/24*) and temporal.relational to verbal expressions such as *earlier, later, before, after*, etc.

In addition, the dataset contains tags for the relations between context 1 and query and context 2 and query, allowing an even more detailed analysis of the results. However, this analysis is still future work and will not be part of this paper.

2.3. Adapting the Dataset

It is also possible to customize the dataset depending on the types of knowledge conflicts to be assessed with the Ragability library.

One approach is to keep the structure of the dataset (the types of columns) as it is and either extend the corpus with additional examples or adapt existing examples. For instance, you can

- **focus on a specific type of reasoning** which is needed for the LLMs to identify knowledge conflicts: If the test should focus, e.g., on numerical knowledge conflicts, only the instances where numerical reasoning is required are used and additional examples can be added.
- **adapt the contexts to a specific domain**: The dataset can be adapted to contain real-world data, e.g., to focus on potential information conflicts in company-specific documents such as fact sheets, manuals, compliance documents, etc. To do so, replace the instances of the current corpus with comparable examples containing contradictory contexts from the real-world data in question. By testing different *testee-LLMs*, the LLM can be identified which is best suited to handle company-relevant conflicting information.

If the dataset is adapted while the structure of the dataset is not altered, all parts of the Ragability library can be used as they are and do not need to be adapted. However, maintaining the structure of the database also requires preserving the relationships between columns, specifically, which columns are contradictory or non-contradictory to one another must remain unchanged.

Another approach is to change the structure of the dataset by removing or adding columns, e.g.: **adding more context columns** with additional (conflicting or non-conflicting) contexts to the dataset to investigate how the LLM answers a user query based on different textual fragments; **adding more specific labels** in an additional column to the dataset to allow for a specific in-depth analysis.

Note: the corpus conversion module of the Ragability library must be adjusted in the following cases: (i) if existing columns are deleted, or (ii) additional columns are added. There are two possible workarounds to address this issue, with the second being the recommended approach:

- Add the new columns in the tsv version of the dataset and adapt the Python module to convert the corpus (ragability_cc_wc1) by adding a function of how to convert the new or adapted columns for further processing.
- Directly adapt the already converted dataset by adding the new instances containing information about the contexts, the query, the relevant checks and metrics, and the relevant tags.

If new examples are added, take care to maintain the initial structure of the context columns, i.e., context 1 and context 2 must be contradictory, context 3 must be non-contradictory to context 1 and contradictory to context 2, and context 4 must be non-contradictory to any other context.

3. Building Blocks of a Ragability Experiment

The building blocks required to run a Ragability experiment are 1) a **dataset** containing knowledge conflicts, 2) **config files** to configure the LLMs to be evaluated (*testee-LLMs*) and the *checker LLM*, 3) **prompt files** to prompt the *testee-LLMs* and the *checker LLM*, and 4) the so-called **Ragability library** comprising a data conversion module (ragability_cc_wc1), an answer generation module (ragability_query), an answer checking module (ragability_check), a module for analyzing the results from the checks (ragability_eval), and a module to convert the output from ragability_check to tsv format.

In the following, the building blocks necessary to run the experiments are described in more detail, it is discussed how these building blocks work together, and how they can be adapted to specific requirements. Figure 1 gives an overview of how the library modules and the input files work together to test the ability of LLMs to handle contradictory contexts.

3.1. Config Files

For conducting the experiments, two config files are need: (i) a config file containing a list of *testee-LLMs* (at least 1 LLM), and (ii) a config file determining the *checker LLM* (1 LLM). The config files can be in json, hjson or yaml format and are used to configure LLMs and providers. Each user needs to create their own config files containing their respective API keys.

3.2. Prompt Files

There are two different files: The file to prompt the *testee-LLMs* to later analyze their responses on how they deal with knowledge conflicts, and the file for prompting a *checker LLM* to compare the responses from the *testee-LLMs* with the answers from the dataset.

Prompts for the *testee-LLM*: There are three different kinds of prompts: The one prompt contains the query without any context in order to check whether the LLM correctly denies to answer the query due to missing information. The other prompt contains n contexts (in our experiments 1-4 contexts) and a query. Another prompt contains n contexts (in our experiments 1-4 contexts) and the request to respond with 'yes' or 'no' if the context is contradictory. Depending on the experiments, the wording of the prompts can be adapted, hints that the contexts might contain conflicting information can be added, as well as examples.

Prompts for the *checker LLM*: There are two different prompts for the *checker LLM*: (i) to check whether the *testee-LLM* refused to answer or gave a concrete answer; (ii) if it gave a concrete answer, to check whether the response is correct, compared to the answers defined in the dataset. The *checker LLM* is prompted to respond with 'yes' or 'no' to the checker prompts. The prompt file can flexibly be adapted to what the *testee-LLMs* should be tested for, e.g., how it should deal with knowledge conflicts, and whether additional hints for the LLM are provided.

3.3. Ragability Library

ragability_cc_wc1 converts the dataset (tsv file) to a format for further processing. During this step, a number of prompts is generated with different context-query combinations, different hints for contradictory content etc. Input is the dataset in tsv format, and output is the converted dataset.

ragability_query prompts the *testee-LLM(s)* to respond to different prompts, comprising context(s), queries, and tags from the dataset. Input are the converted dataset, the prompt file and the config file containing a list of *testee-LLM(s)*. Output is a hjson file containing per prompt the information from the dataset, relevant tags and the response from the *testee-LLM(s)*.

ragability_check prompts a *checker LLM* to check the responses from the *testee-LLM(s)*. Input is the output from ragability_query, the prompt file and the config file containing the *checker LLM*. Output is a hjson file containing per prompt the evaluation of the *checker LLM*.

ragability_2tsv converts the output from ragability_check to tsv format for easier manual analysis. The tsv file contains columns relevant for a manual

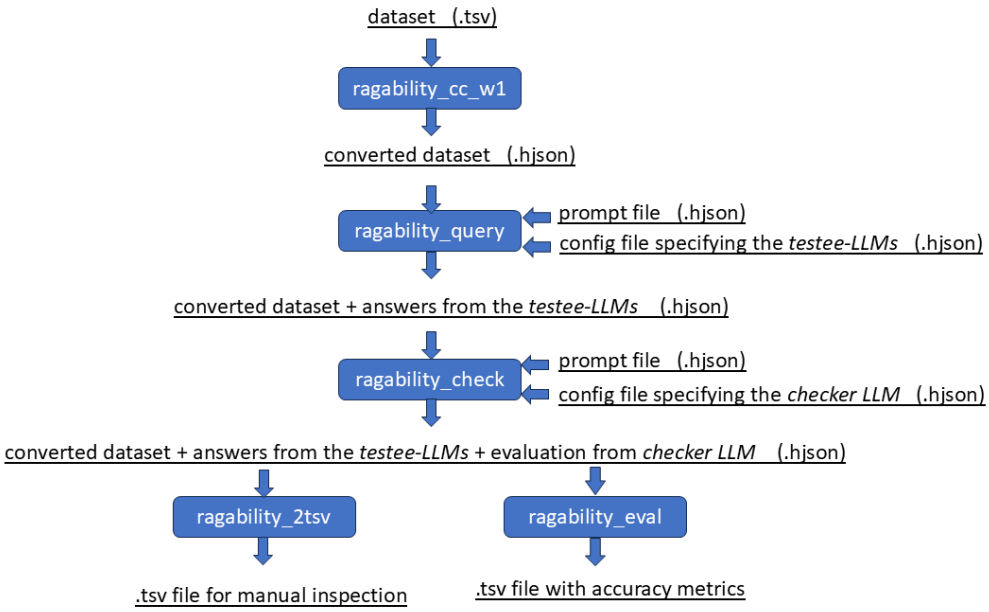

Figure 1: How to run a Ragability experiment. Modules of the Ragability library to run the experiment are indicated by blue boxes.

analysis of the prompts and the responses from the LLM(s).

ragability_eval analyses the results from ragability_check and outputs a tsv file containing the accuracy per metric, which can also be filtered by tags (e.g., 'q_two_contexts' to look at the results of prompts containing only a query and two contexts).

3.4. Metrics

Currently, there are four different metrics implemented:

- **correct_answer_all**: rate of the answer being correct for all instances (cf. the prompts for the *testee-LLM*).
- **correct_answer_answerable**: rate of the answers being correct among all instances where a correct answer can be given.
- **refusal_not_answerable**: rate of correctly refusing to answer a query that cannot be answered because of missing or contradictory context.
- **contradiction_identification**: rate of correctly identifying a contradiction when the *testee-LLM* is explicitly asked to respond with 'yes' or 'no'.

To calculate the accuracy for the four different metrics, the answers provided in ragability_query are checked, depending on the type of prompt. For different types of prompts different checks are made to identify, e.g., whether a query was answered correctly, informing the metrics 'correct_answer_all'. In case non-contradictory contexts are provided and a query relating to the semantic content is given, the answer is correct if it semantically reflects the answers provided in the dataset (see the answer contexts in Table 1). In the case of contradictory contexts, the response is considered to be correct if the LLM refuses to answer the query.

4. Experiment

4.1. Setup of the Experiment

The module ragability_cc_wc1 converts the dataset and generates a number of different context-query combinations. In the presented experiments, 20 different prompts are generated out of one instance of the Ragability corpus, resulting in 1060 test instances the LLMs need to respond to. Via the generated test instances, different research questions can be investigated. Our preliminary findings suggested that the sequence of texts in the prompt has an impact on the results. Therefore, for all combinations with two contexts, we generated separate instances for both sequences, e.g., context x + context y and context y + context x to smoothen that effect. However, the current version of the conversion module does not generate all possible permutations for more than 2 contexts, as this would significantly increase the total number of

instances. Also, via the generated test instances it can be further investigated whether it is easier for an LLM to identify a conflict / no conflict if it is prompted to answer a query on the semantic content in the provided context, or to respond with 'yes' or 'no' whether there is a contradiction. Moreover, responses to prompts providing additional hints that there might be a contradiction can be compared to responses without these hints in the prompts. Adding hints is inspired by (Hou et al., 2024) who showed that prompting LLMs to pay attention to conflicting information significantly improves their performance to correctly answer these questions.

Experiments were conducted with state-of-the-art LLMs of different size and architecture, including gemini-2.5-flash, gemini-2.5-pro, gpt-5-mini, gpt-5, gpt-4.1, claude-sonnet-4-5-20250929, llama3 (8B). With these LLMs we cover six proprietary models and one open-source model (llama3), large and small (llama3, gpt-5-mini), as well as thinking models (gpt-5, gpt-5-mini, gemini-2.5-pro, claude-sonnet-4-5-20250929). Where possible, the temperature of the applied models was set to 0 to keep the LLM output as deterministic as possible. For the gpt 5 models, it is not possible to manually set the temperature via the API. To assess variation, we queried gpt-5 and gpt-5-mini five times each. Gpt-5 reached an average accuracy of 0.7717 (std = 0.0042) and gpt-5-mini of 0.7506 (std = 0.0047). A Fleiss's κ of 0.88 for gpt-5 and 0.84 for gpt-5-mini suggests a high consistency over the 5 runs per model. Because of the strong alignment between runs, the following analysis focuses on a single run of gpt 5 models, similar to the approach taken with the other LLMs.

As *checker LLMs*, we applied gemini-2.5-flash and gpt-5. To manually verify the *checker LLMs*, we sampled all test-set IDs where a *checker LLM* was used for evaluation, creating approximately equal-sized sub-samples for each of the seven LLMs. We then manually annotated the *checker LLM's* evaluation on the same sub-sample, yielding an accuracy score of 0.916 for the Ragability pipeline including gemini-2.5-flash as *checker LLM* and 0.954 for the pipeline including gpt-5.

In the experiment, we also varied the prompts and added two contradiction examples to the system prompt. However, the accuracy scores decreased for the prompts which included examples. In the following analysis, we concentrate on experiments without examples in the system prompt.

4.2. Analysis and Discussion

For the **contradiction identification task**, i.e., when presented with contradictory or non-contradictory contexts and the task was to respond 'yes' if the context is contradictory and 'no' if the context is non-contradictory, the LLMs showed high performance, see Table 2. No *checker LLM* was needed for this task. Gemini-2.5-pro achieved the highest accuracy score (0.9488), closely followed by gpt-5-mini (0.9434), gpt5 (0.9407), and gemini-2.5-flash (0.9380) (mean = 0.899, std = 0.075). The performance for the tasks to correctly respond to an answerable query where not-conflicting context is provided (metric correct_answer_answerable) and for the task to refuse to answer a semantic query due to missing or conflicting context (metric refusal_not_answerable) was significantly lower, see also Table 2 for the accuracy scores of the different metrics.

Looking at the **overall performance**, the LLMs responded to the 1060 test instances with an accuracy ranging from 0.7698 (gpt-5) to 0.5302 (llama3) (mean = 0.7133, std = 0.0783) with gpt-5 as *checker LLM* and an accuracy ranging from 0.7665 (gpt-5) to 0.5104 (llama3) (mean = 0.6739, std = 0.0806) with gemini-2.5-flash as *checker LLM*. The manual evaluation of the *checker LLMs* revealed that both models produced false negatives by missing correct responses, but neither generated false positives. Thus, the accuracy score for the different metrics in Table 2 also indicate **differences in *checker LLMs*:** gpt-5 performed better in identifying whether an answerable query was answered correctly, while gemini-2.5-flash could identify more correct answers, when the *testee-LLMs* were expected to refuse to answer a query. As in general, gpt-5 was able to identify more true positives than gemini-2.5-flash, the further analysis will focus on runs applying gpt-5 as *checker LLM*.

To gain more insights in the **differences of overall performance of 7 *testee-LLMs***, a multiple, pairwise comparison was conducted, using McNemar's test with continuity correction (see Table 3). A Bonferroni correction was not applied, as manual evaluation of the *checker LLM* showed that, in the extracted sample (689 instances per *checker LLM*), there were no false positives – only false negatives. With statistical significance defined as $p < 0.05$, gpt-5, gemini-2.5-flash, gpt-5-mini, and gemini-2.5-pro showed significant better performance than the other three models. Gpt-4.1 significantly outperformed claude-sonnet-4-5 and llama3, while claude-sonnet-4-5 significantly outperformed llama3.

To better understand LLM behavior in conflicting contexts, we compared responses to test instances with **two contradictory contexts, both with and without hints of a potential contradiction**. Without hints, the accuracy scores range between 0.1321 (gpt-5) and 0 (llama3) (mean = 0.0701, std = 0.0483). However, in accordance with (Hou et al., 2024), the accuracy scores increase significantly if a hint is added from 0.7925 (gpt-4.1)

metric	gemini-2.5-flash	gemini-2.5-pro	gpt-5-mini	gpt-5	gpt-4.1	claude-sonnet-4-5-20250929	llama3
contradiction_identification	0.9380	0.9488	0.9434	0.9407	0.9030	0.8949	0.7224
checker LLM: gpt 5							
correct_answer_all	0.7623	0.7528	0.7557	0.7698	0.7264	0.6962	0.5302
refusal_not_answerable	0.5472	0.5418	0.5903	0.5606	0.5013	0.5741	0.2830
correct_answer_answerable	0.8082	0.7704	0.7296	0.8145	0.7830	0.6069	0.5943
checker LLM: gemini-2.5-flash							
correct_answer_all	0.7	0.7292	0.7349	0.7665	0.6462	0.6302	0.5104
refusal_not_answerable	0.5876	0.6469	0.6280	0.5772	0.5121	0.6092	0.3019
correct_answer_answerable	0.5534	0.5691	0.6164	0.7830	0.5031	0.3459	0.5063

Table 2: The accuracy scores of the four metrics for the different LLMs. For the metrics, 'correct_answer_all', 'refusal_not_answerable' and 'correct_answer_answerable', two checker models were applied for a semantic comparison of the response of the *testee-LLM* and the answers provided in the dataset.

to 0.9434 (claude-sonnet-4-5-20250929) for the proprietary models and to 0.3585 for the smaller, open-source LLM llama3 (mean = 0.8046, std = 0.1897). Adding these hints per default still needs to be treated with caution, as for non-contradictory contexts, the accuracy decreased when the hint was added. For two non-contradictory contexts without a hint, the accuracy scores range from 0.7075 (claude-sonnet-4-5-20250929) to 0.8491 (gemini-2.5-flash) (mean = 0.779, std = 0.0531). Adding the hint that the contexts might be contradictory decreased the accuracy scores for all LLMs to a range from 0.2453 (llama3) to 0.7358 (gpt-5) (mean = 0.562, std = 0.167).

With regards to **different types of reasoning**, the LLMs to be tested did not show obvious tendencies. For categorical reasoning, the accuracy scores range from 0.5329 (llama3) to 0.7539 (gpt-5) (mean = 0.7024, sd = 0.0712) and for numerical reasoning from 0.4962 (llama3) to 0.7885 (gpt5) (mean = 0.7096, std = 0.0939). The accuracy scores for temporal.numerical reasoning range between 0.5643 (llama3) and 0.8286 (gpt-5) (mean = 0.7561, std = 0.0914), and for temporal.relational reasoning between 0.65 (llama3, gpt5) and 0.85 (gemini-2.5-flash) (mean = 0.7357, std = 0.0693).

Summarizing the observed effects:

- Especially the proprietary LLMs showed high performance for contradiction identification (*Is there a contradiction yes or no?*), while they struggled with correctly answering the content related query.
- Adding a hint that there might be contradictory information increased the performance of answering a query based on contradictory information significantly. However, it also significantly decreased the performance of responding to non-contradictory information. In the context of RAG systems, it is most likely that there are more non-contradictory contexts than contradictory ones. Therefore, it might be helpful to add a contradiction alarm functionality to the RAG where parallel to the answer generation prompt a check prompt for conflicting contexts (*Is there a contradiction yes or no?*) is given, instead of adding a hint to the answer generation prompt that there might be a contradiction.
- The LLMs did not show obvious tendencies with regards to the type of reasoning (categorical, numerical, temporal). Further investigations are needed on how the reasoning type affects an LLM's capability to handle conflicting information.
- The performance of the LLMs decreased when adding examples to the system prompts. This might be an artifact arising from prompt sensitivity (Huang et al., 2024), changing with the given examples. This needs to be further investigated by increasing the diversity of examples in the prompts.
- The proprietary LLMs showed higher performance in the different tasks as compared to the smaller, open-source model llama3.

5. Conclusion

The paper presents a benchmark (a dataset and a library for experimentation) to test the ability of LLMs to handle conflicting contexts. For their identification, the conflicts represented in the dataset are either lexically overt or require implicit reasoning. The dataset comprises instances with entity-related, numerical, or time-related (numerical or relational) conflicts. It is carefully handcrafted and the

Pairwise comparison	n_{10}	n_{01}	χ^2	p
gpt-5 vs gemini-2.5-flash	46	38	0.583	0.445
gpt-5 vs gpt-5-mini	50	35	2.306	0.129
gpt-5 vs gemini-2.5-pro	62	44	2.746	0.099
gpt-5 vs gpt-4.1	88	42	15.577	<0.001
gpt-5 vs claude-sonnet-4-5	124	46	34.876	<0.001
gpt-5 vs llama3	289	35	197.559	<0.001
gemini-2.5-flash vs gpt-5-mini	52	45	0.371	0.542
gemini-2.5-flash vs gemini-2.5-pro	55	45	0.81	0.368
gemini-2.5-flash vs gpt-4.1	83	45	10.695	0.001
gemini-2.5-flash vs claude-sonnet-4-5	116	46	29.389	<0.001
gemini-2.5-flash vs llama3	285	39	185.262	<0.001
gpt-5-mini vs gemini-2.5-pro	59	56	0.035	0.852
gpt-5-mini vs gpt-4.1	82	51	6.767	0.009
gpt-5-mini vs claude-sonnet-4-5	107	44	25.457	<0.001
gpt-5-mini vs llama3	279	40	177.567	<0.001
gemini-2.5-pro vs gpt-4.1	87	59	4.993	0.025
gemini-2.5-pro vs claude-sonnet-4-5	115	55	20.476	<0.001
gemini-2.5-pro vs llama3	288	52	162.426	<0.001
gpt-4.1 vs claude-sonnet-4-5	89	57	6.582	0.01
gpt-4.1 vs llama3	254	46	142.83	<0.001
claude-sonnet-4-5 vs llama3	232	56	106.337	<0.001

Table 3: Multiple pairwise comparisons were conducted to compare the performance of 7 LLMs on the Ragability testset, applying McNemar's test (with continuity correction). As *checker LLM*, gpt-5 was used. This table reports all pairwise comparisons, including the number of correct responses provided only by the first listed model (n_{10}), only by the second listed model (n_{01}), as well as χ^2 statistics and the corresponding p value.

instances are inspired by entries from WikiContradict, whereby real entities are replaced by fantasy entities. The purpose of the benchmark is twofold: first, current LLMs can be tested on their ability on how to deal with conflicting context information to gain insights in individual and overlapping difficulties regarding different aspects of contradictory information, e.g., (i) different tasks, such as answering context related queries or responding with 'yes' or 'no' when asked whether there is a knowledge conflict, (ii) prompting, such as adding examples or hints, or (iii) different types of contradictions. In the experiments presented in this paper, from the 52 handcrafted entries in the dataset, 1060 test cases were automatically generated. 7 state-of-the-art LLMs (6 proprietary, 1 open-source model) were prompted to respond to 1060 different test cases each, applying two of these LLMs to check those replies where semantic evaluation is required. In this respect, the benchmark is a valuable instrument for identifying which LLM(s) perform best under which contradiction identification setups.

Second, individual RAG systems can be benchmarked on their ability to deal with contradictory facts. Both the data set and the library can be adapted to individual requirements with regard to the RAG system. Currently, the presented benchmark is used in practice as part of an AI governance system.[5] It has been applied and tested within Ragability's operational framework, helping customers validate their RAG system implementations and identify weaknesses in knowledge retrieval and contextual reasoning. Important assets of the benchmark in practice are that the experiments can be easily tailored to selected aspects of testing without the necessity of running the full benchmark and that the dataset can be flexibly customized and extended.

In summary, running the benchmark on a broad range of state-of-the-art LLMs provides insights into which kinds of conflicting contexts are still hard for LLMs to deal with and how prompting influences the outcome. This, in turn, provides valuable guidelines for special purpose testing of individual companies' RAG systems.

Future work includes a more thorough analysis of the results based on the different tags in the dataset of how well individual LLMs manage to identify different types of conflicts (explicit vs. implicit conflicts, ambiguous vs. unequivocal interpretations). Moreover, prompt sensitivity needs to be further investigated by increasing the diversity of prompts, e.g., by adding different examples of contradictions to the system prompts.

[5] See Mäntymäki et al. (2022) for some background on AI governance systems.

5.1. Ethical Considerations and Limitations

Ethical considerations: The dataset and benchmark help to assess the veracity of LLMs by focusing on how different LLMs handle contexts that might contain conflicting information. This is particularly important for Retrieval Augmented Generation systems, as such systems are widely used especially in industry and the public sector to combine one's own (in many cases dynamically growing and possibly changing) data sources with the power of LLMs. These are constellations in which the LLM is most likely to be presented with conflicting contextual information. Thus, concise testing of an employed LLM's ability to identify and handle conflicting information is indispensable for the use of RAG-based applications. Nevertheless, it is important for the users of such a benchmark to be aware that the provided benchmark can only give a first impression on an LLM's conflict handling potential, and that it might be necessary to adapt the dataset in a way such that it covers those conflicts which are most relevant for one's own data and applications. In addition, high scores on contradiction benchmarks can create a false sense of security regarding an LLM's factual accuracy. Whereby the accuracy of the *checker LLM* influences the evaluation result and one needs to be aware of checking the *checker LLM* as well.

This brings us to the discussion of restrictions of the current benchmark, in which the number of contexts per entry is restricted to 4 with 2 different reasoning-related types of conflicts (implicit, lexically overt), and contradictions at entity, numerical, or temporal level. Adding more contexts and more conflict types requires manual intervention, including the extension of the tag set, adaptation of prompts for the *testee*- and the *checker LLMs*, and extension of the library code. We use gemini-2.5-flash and gpt-5 as *checker LLMs* because they showed good results based on random manual inspections. However, also *checker LLMs* need to undergo systematic tests to ensure the quality of their checking results. Furthermore, the temperature for the two gpt-5 LLMs cannot be manually set to 0 via the API. Although Fleiss's κ shows a high consistency over the 5 runs per model, a direct comparison between the gpt-5 models and the other LLMs with temperature set to 0 needs to be treated with caution.

6. Acknowledgments

Part of this work was supported by the company Daiki GMBH (dai.ki), and by the Austrian Research Promotion Agency (www.ffg.at) within the project Chatlyn+.

7. References

Tyler A Chang and Benjamin K Bergen. 2024. Language model behavior: A comprehensive survey. *Computational Linguistics*, 50(1):293–350.

Vignesh Gokul, Srikanth Tenneti, and Alwarappan Nakkiran. 2025. Contradiction detection in rag systems: Evaluating llms as context validators for improved information consistency. *arXiv preprint arXiv:2504.00180*.

Yufang Hou, Alessandra Pascale, Javier Carnerero-Cano, Tigran Tchrakian, Radu Marinescu, Elizabeth Daly, Inkit Padhi, and Prasanna Sattigeri. 2024. Wikicontradict: A benchmark for evaluating llms on real-world knowledge conflicts from wikipedia. *Advances in Neural Information Processing Systems*, 37:109701–109747.

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, et al. 2024. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*.

Yifan Jia, Kailin Jiang, Yuyang Liang, Qihan Ren, Yi Xin, Rui Yang, Fenze Feng, Mingcai Chen, Hengyang Lu, Haozhe Wang, et al. 2025. Benchmarking multimodal knowledge conflict for large multimodal models. *arXiv preprint arXiv:2505.19509*.

Cheng Jiayang, Chunkit Chan, Qianqian Zhuang, Lin Qiu, Tianhang Zhang, Tengxiao Liu, Yangqiu Song, Yue Zhang, Pengfei Liu, and Zheng Zhang. 2024. Econ: On the detection and resolution of evidence conflicts. *arXiv preprint arXiv:2410.04068*.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. 2023. Realtime qa: What's the answer right now? *Advances in neural information processing systems*, 36:49025–49043.

Jungyeon Lee, Kangmin Lee, and Taeuk Kim. 2025. Magic: A multi-hop and graph-based benchmark for inter-context conflicts in retrieval-augmented generation. *arXiv preprint arXiv:2507.21544*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Matti Mäntymäki, Matti Minkkinen, Teemu Birkstedt, and Mika Viljanen. 2022. Putting ai ethics into practice: The hourglass model of organizational ai governance. *arXiv preprint arXiv:2206.00335*.

Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. 2024. Conflictbank: A benchmark for evaluating the influence of knowledge conflicts in llms. In *NeurIPS*.

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

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for llms: A survey. *arXiv preprint arXiv:2403.08319*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.