Knowledge-Enhanced Geospatial QA: Integrating Wikidata Fact Verification with LLMs

Sanaz Saki Norouzi, Pascal Hitzler

Kansas State University
Manhattan, KS USA
{sanazsn, hitzler}@ksu.edu

Abstract

Geospatial question answering (QA) challenges large language models (LLMs) in reasoning about geospatial relationships. This paper presents a hybrid framework that integrates LLMs with Wikidata for fact verification and retrieval, enhancing their geospatial reasoning capabilities. The framework generates facts, verifies them against Wikidata, and uses validated knowledge for use in a Retrieval-Augmented Generation (RAG) pipeline. Experimental results demonstrate that this approach outperforms zero-shot prompting for all tested models, including GPT-3.5-turbo-0125, Llama-3-8b, and Qwen-2.5-14b, showcasing its effectiveness in improving geospatial QA accuracy.

Introduction

The advent of large language models (LLMs) has led to widespread experimentation with these models across various tasks. These models exhibit several strengths, such as the ability to provide reasonable answers to a wide range of questions and applications (Sartori and Orrù 2023; Shojaee-Mend et al. 2024). However, they also have notable drawbacks. One significant issue is their tendency to generate false or misleading information, a phenomenon known as hallucination. Additionally, even when provided with accurate information, LLMs often struggle with logical reasoning, which can result in misleading conclusions and incorrect answers (Liu, Sheng, and Hu 2024; Chen and Shu 2023; Jiang et al. 2024).

To address these challenges, developing LLMs frameworks with reduced hallucination is crucial. One effective approach to achieving this is through rigorous fact verification. By ensuring the accuracy of the information generated, we can enhance the reliability of these models, making them more suitable for important applications. Moreover, end users require outputs that are not only accurate but also trustworthy. One possible way to address this issue is the implementation of strong fact verification mechanisms within LLMs.

Knowledge graphs (KG) are structured representations of knowledge that connect entities through meaningful relationships and one way to reduce hallucination is to integrate KGs with LLMs for fact verification (Ehrlinger and Wöß 2016). By mitigating hallucinations and ensuring that the generated information is fact-checked, we can significantly improve the utility and safety of LLMs in various domains. This approach will help in building user confidence and ensuring that these models can be relied upon for critical decision-making processes.

Hallucinations can be categorized into two main types: factuality hallucinations and faithfulness hallucinations. Each type includes several sub-types, such as factual inconsistency, factual fabrication, and logical inconsistency (Huang et al. 2023). These hallucinations pose a significant challenge to fact verification in LLMs, as they are prone to generating outdated facts. Moreover, LLMs often produce fluent and authoritative-sounding text, which makes it difficult to identify when hallucinations occur, and factually incorrect or fabricated information can blend seamlessly with accurate content (Huang et al. 2023; Tyen et al. 2023). The challenge is further worsened by the absence of robust automated mechanisms for real-time fact-checking; LLMs are not inherently equipped to cross-reference their outputs with reliable sources or databases. While human oversight can address accuracy issues, it is resource-intensive, costly, and impractical for large datasets or real-time interactions, making scalability a major obstacle (Bowman et al. 2022). Addressing these issues requires innovations in data integration and scalable fact-verification solutions that effectively combine AI-driven tools with human expertise. A critical component of such solutions is the integration of external knowledge sources for real-time fact-checking, which enables LLMs to provide accurate and up-to-date information.

Considering the challenges and methods discussed, understanding geospatial reasoning is crucial for many question-answering tasks, especially those related to locations, distances, or boundaries, which play an important role when a disaster strikes or in any situation requiring urgent intervention. Determining which states or regions are geographically closest to the affected area is essential. Despite their generative capabilities, current LLMs often fail to reason accurately about such spatial concepts, leading to errors or hallucinations in their outputs. To address this limitation, we propose a framework that integrates a fact verification approach with the retrieval-augmented generation (RAG) technique to enhance the accuracy and reliability of LLM outputs in the geospatial QA reasoning task. In geospatial

QA, LLMs often struggle to reason about latitude and longitude comparisons, even when they inherently possess relevant knowledge. To address this, we propose a method that involves the following steps: first, extract the facts generated by the LLM for answering a question, structured as triples. Next, identify the corresponding entity ID and property ID from Wikidata, and use these to create a SPARQL query to retrieve accurate information. Finally, compare the LLM-generated facts with the retrieved data from Wikidata. If they match, the facts are considered verified and stored in a vector database. These verified facts can then be utilized in an RAG pipeline to enhance geospatial reasoning in QA tasks.

The remainder of this paper is organized as follows: Section 2 reviews related work on hallucination reduction and fact verification in LLMs. Section 3 details our proposed methodology, including the integration of Wikidata, the verification pipeline, and the use of RAG. Section 4 presents experimental results. Finally, Section 5 concludes the paper and outlines future directions.

Related Work

Recent research to enhance hallucination mitigation in QA systems has emphasized the integration of knowledge graphs to ensure accurate and reliable outputs. In (Liu et al. 2024), the GraphEval framework is proposed for evaluating the factuality of LLMs using declarative fact sentences from DBpedia. Their method involves generating templates for fact triples, creating negative samples, and querying LLMs for each fact to classify it as "True", "False", or "I don't know", and then, they train a judge model to evaluate the factuality of the entire knowledge graph (KG). Kim et al. proposes KG-GPT, a framework integrating KGs with LLMs for tasks like fact verification and question answering. It segments claims into triples, retrieves relevant subgraphs, and infers logical conclusions (Kim et al. 2023). Adam and Kliegr introduces a method for statement verification in KGs, to do so, they proposed a zero-shot prompting method using GPT to validate triples against text snippets from grounding documents (Adam and Kliegr 2024). Wang et al. proposes a fine-grained framework for verifying LLM-generated content, focusing on open and closed questions. It involves splitting responses into sentences, decontextualizing them, and identifying check-worthy claims, and for each claim, it generates search queries, retrieves the top five evidence snippets via Google, and determines the stance of the evidence based on the majority votes, then correct claims are merged and compared with references (Wang et al. 2023). On the other hand, RAG techniques are also used for reducing hallucination. For example, Li, Yuan, and Zhang explores improving the factual accuracy of an LLM for domain-specific and time-sensitive queries using a RAG system (Li, Yuan, and Zhang 2024). LLM-AUGMENTER, a framework to improve the factual accuracy of LLMs is proposed based on integrating external knowledge (Peng et al. 2023). The authors in (Vu et al. 2023) introduce a few-shot in-context learning method that retrieves and integrates relevant information into the LLM prompts which reduces hallucination in the QA task. Zhao et al. Zhao et al. propose a method to improve the factual accuracy of LLMs by editing their reasoning chains using external knowledge (Zhao et al. 2023). There are some works related to geographical QA with LLMs and explore the integration of Large Language Models (LLMs) with geospatial tasks (Manvi et al. 2023; Zhang et al. 2023). In (Feng, Ding, and Xiao 2023), the authors introduced GeoQAMap, a system that converts natural language questions into SPARQL queries to retrieve geospatial information from Wikidata. The retrieved data is then used to create interactive maps, providing visual answers to the questions. Mooney et al. provided an evaluation of ChatGPT performance in geospatial and GIS skills (Mooney et al. 2023).

Method

This study proposes an approach to enhance geospatial question answering by integrating large language models with knowledge from Wikidata. The method is designed to address the limitations of LLMs in reasoning by introducing a fact-verification mechanism and leveraging Wikidata. The proposed method is illustrated in Figure 1. This approach uses three different LLMs that are efficient with minimal resources while still performing well on general tasks. It begins with generating facts for each input in the form of triples, which are then linked to Wikidata by assigning entity and property IDs using GPT-4o. A predefined SPARQL query is then used to retrieve relevant information from Wikidata. Next, the retrieved information is verified against the generated facts, a step that is also handled by GPT-4o. Finally, verified facts are incorporated into a RAG pipeline to assess the performance of selected LLMs. Using RAG helps by retrieving location coordinates (longitude and latitude) within the LLM's context, improving accuracy. GPT-4o is specifically used to map entities/properties and compare retrieved and generated facts because it is different from the models being evaluated while still delivering reliable performance.

The following subsections elaborate on the key components of the proposed framework. To illustrate, we will go through each step of the process and provide the corresponding output at each stage using a running example.

Question: Which of these states is the farthest north?
Choices: [West Virginia, Louisiana, Arizona, Oklahoma]

Fact Generation by LLMs

This step is marked as (1) in Figure 1, three LLMs are used: GPT-3.5-turbo-0125, Llama-3-8b, and Qwen 2.5-14b. Each model is tasked with two objectives: 1- Provide an answer to the question using zero-shot prompting, which is chosen as the simplest and initial approach. 2- Generate the necessary facts to answer the question, formatted as triples in the format of Predicate(Subject, Object). The prompts used in this step are outlined below.

Mapping Entities and Properties to Wikidata

In this step, marked as (2) in the Figure 1, the GPT-4o model is specifically chosen because it differs from the models used

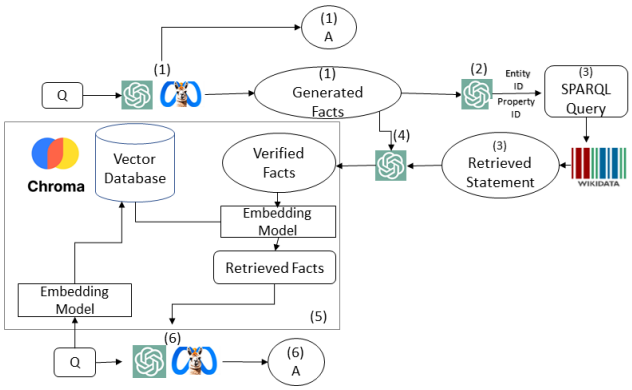

Figure 1: Proposed Method

Prompt for Getting the Answer
Provide the answer to the question based on the position of the correct choice. The answer should be given as 0, 1, 2, or 3. For example, if the choices are [A, B, C, D], then if A is the answer you should provide 0, if B is the answer provide 1, if C is the answer provide 2, and then if D is the answer provide 3 as the answer. Provide the answer in the form of Answer: index number of the choice. Now provide the answer for: {question}, choices: {choices}
The sample output for this prompt is: "The correct answer is 2. Arizona is the state farthest north among the given options."

Prompt for Generating Supporting Facts to Answer the Question
Generate facts for each choice in the list of choices that help answer this question in the form of triples like Predicate(Subject, Object) like hasBirthPlace(Bill, Paris):{question}, choices:{choices}

The sample output for this prompt is: `"facts": "hasLatitude(Oklahoma, 35.5000) hasLatitude(Arizona, 33.2000) hasLatitude(Louisiana, 30.0000) hasLatitude(West Virginia, 39.2000)"`

for evaluation while still delivering reasonable performance compared to other LLMs. In this step, for each fact generated by the LLM, GPT-4o is prompted to provide the corresponding entity ID and property ID of the fact in Wikidata. The prompt used for this task is as follows:

Given the following triple, identify a suitable EntityID and PropertyID from Wikidata for this triple. The format should be like EntityID: Q100, and PropertyID: P65. {triple}

SPARQL Query over Wikidata and Execution

In this step, marked as (3) in Figure 1, a function is implemented to execute pre-defined SPARQL query over Wikidata (via its query service), which is a multilingual, collaborative knowledge base offering structured data for humans and machines. In the semantic web, Wikidata serves as a vast, community-driven graph of interconnected entities, providing a free, shared source of data that is easily reusable and integrable. Each item in this knowledge base is assigned a unique identifier called a QID and represents a topic, concept, or object, with properties and values linking them to form statements. Thus, this function is called with the input generated from the previous step (Entity ID and Property ID). The query output provides the actual information retrieved from Wikidata, which is then utilized in the next step. The SPARQL query template is illustrated below:

The SPARQL Query:

```
 SELECT ?propertyLabel ?valueLabel
WHERE {
    wd:{entity_id} wdt:{property_id}
?value.
    SERVICE wikibase:label {
bd:serviceParam wikibase:language
"[AUTO_LANGUAGE],en". }
}
```

When making a query to Wikidata, the result will depend on the query's outcome:
1- **If the query retrieves information:**
The output displays the retrieved information. For instance, if the triple is "hasLatitude(Ohio, 40.02)", querying with the provided property ID and entity ID yields a result like: "Point(-82.5 40.5)".
2- **If the query does not retrieve any information:**
The output displays "No results found for property {Property_ID} and entity {Entity_ID}".

Fact Verification

This step, marked as (4) in Figure 1, involves using GPT-4o to verify the alignment between the facts generated by the LLMs and the information retrieved from Wikidata. GPT-4o checks whether the two sources point to the same information, determining if the generated fact can be considered verified. The prompt used for this verification process is as follows:

If the output of this step is True, the fact is considered verified; otherwise, it is considered unverified.

RAG for QA

This step, marked as (5) in Figure 1, represents the retrieval component. Verified facts from the previous step are stored in a vector database, with ChromaDB[1] used for this purpose. LangChain[2] is employed as the framework to manage retrieval and augmentation processes. Embeddings for both documents and queries are generated using Hugging Face's pre-trained model "all-MiniLM-L6-v2"[3].

[1]https://www.trychroma.com/
[2]https://www.langchain.com/
[3]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

”We have two pieces of information, and we want to determine if they refer to the same thing. For example, hasLatitude(West Virginia, 39.0) and Point(-79.0, 37.5) could represent the same concept, as the latitude of West Virginia covers a range that includes both 37.5 and 39.0. The provided triple should read as West Virginia hasLatitude 39.0. Generate the output as either True or False: first information: {generated fact by the LLM} second information: {retrieved info by Wikidata}
The sample output for this prompt is: The first piece of information is ”hasLatitude(Oklahoma, 35.5000)”, which means the latitude 35.5 is within the range of Oklahoma’s latitude. - The second piece of information is ”Point(-98.0 35.5)”, which indicates a point with a longitude of -98.0 and a latitude of 35.5. Since: - The latitude 35.5 is within Oklahoma’s latitude range. Both pieces of information are consistent with a location within Oklahoma. Therefore, the output is: True

QA in the RAG Pipeline

This step, marked as (6) in Figure 1, involves performing zero-shot QA within the RAG pipeline, similar to Step 1. The same prompt used for generating answers in Step 1 is applied here as well. This step is crucial for evaluating the performance of the framework. The LLMs used, as in Step (1), are GPT-3.5, Llama-3-8b, and Qwen-2.5-14b.

The sample output for this prompt is: answer: West Virginia explanation: The latitude of West Virginia is 39.2000, which is the highest among the given options.

Experimental Settings

The details of the framework were described in the previous section. This section focuses on the dataset and the general experimental setup used for the experiments.

Data

The dataset used in this study is a subset of questions from the ScienceQA dataset (Lu et al. 2022), available on HuggingFace[4]. We specifically selected questions related to geospatial reasoning, focusing on comparisons of states’ latitudes and longitudes.

After filtering for geospatial reasoning questions, we identified a total of 400 questions for this purpose. Geospatial reasoning tasks are particularly challenging for LLMs as they often struggle with logical reasoning in such contexts.

Although the ScienceQA dataset is multimodal, this work focuses on text-based geospatial reasoning, disregarding the images.

In the below box, two sample outputs generated by the Llama-3-8b model for the above question are shown, demonstrating its struggles with logical reasoning.

[4]https://huggingface.co/datasets/derek-thomas/ScienceQA

First Output:
Arizona is the farthest north. Arizona is located at a latitude of approximately **33.5°N**, which makes it the farthest north among the given options. **West Virginia** is located at a latitude of approximately **38.4°N**, **Louisiana** at approximately **30.5°N**, and **Oklahoma** at approximately **35.5°N**.

Second Output:
The state that is the farthest north is Oklahoma. It is located at approximately **36.5°N** latitude, which is the **highest latitude** among the four options. **West Virginia** is located at approximately **38.5°N** latitude, **Louisiana** is located at approximately **30.5°N** latitude, and **Arizona** is located at approximately **34.5°N** latitude. Therefore, Oklahoma is the state that is farthest north.

Experimental Setup

The experiments with open-source LLMs were conducted on a single A100 GPU, with the temperature set to zero for all runs. For open-source LLMs, we used half-precision floating-point computations to improve memory efficiency.

Results

This section presents the experimental results to evaluate the effectiveness of the proposed method. We compare the performance of the hybrid approach, which integrates LLMs with Wikidata, against baseline zero-shot prompting for geospatial question answering. The results highlight the impact of fact verification and retrieval on improving accuracy across all tested models, including GPT-3.5, Llama-3-8b, and Qwen-2.5-14b. The performance of the proposed framework is evaluated using the accuracy metric, which measures the percentage of correctly answered questions, and the results are shown in Table 1.

Model	Accuracy
GPT-3.5-zero shot	76%
GPT-3.5-proposed	**91%**
Llama-3-8b-zero shot	57%
Llama-3-8b-proposed	**76.5%**
Qwen-2.5-14b-zero shot	89%
Qwen-2.5-14b-proposed	**91.5%**

Table 1: Comparison of accuracy between baseline and proposed method for different models.

In a separate evaluation, we assess each LLM to determine how many of its generated facts are verifiable using Wikidata. Some examples of the verified and unverified facts are shown in Table 2. For this analysis, we create a set of unique generated facts by removing duplicates, and the findings are summarized in Table 3. Based on the results in Table 3, 58.7% of the facts generated by GPT-3.5 are verifiable with Wikidata, and this percentage is 51.5% for Llama-3-8b and 47% for Qwen-2.5-14b. Some facts are marked as unverified not because they are false, but because no suitable property was found for them during the mapping step. This means that, in some cases, the triples generated by LLMs

Model	Generated Fact	Retrieved Info by Wikidata	Verified/Unverified
GPT-3.5	isSouthOf(West Virginia, Louisiana)	None	Unverified
GPT-3.5	hasLongitude(Connecticut, -72.68)	Point(-72.7 41.6)	Verified
Llama-3-8b	isFarthestSouth(Wyoming, yes)	None	Unverified
Llama-3-8b	hasLatitude(South Carolina, 34.0)	Point(33.8 -81.1)	Verified
Qwen-2.5-14b	hasEasternMostBorder(Montana, Idaho)	North Dakota	Unverified
Qwen-2.5-14b	hasCapitalCity(Louisiana, Baton_Rouge)	Baton Rouge	Verified

Table 2: Examples of facts generated by LLMs and statements retrieved from Wikidata

Model	# Verified Facts	# Unverified Facts
GPT-3.5	88	62
Llama-3-8b	69	65
Qwen-2.5-14b	127	143

Table 3: The number of verified and unverified facts for each model

have predicates that don't match any existing property of Wikidata.

Considering all the results provided in the tables, it can be concluded that the proposed method is effective across all models, as its accuracy consistently improves the baseline accuracy for each model. This shows that the proposed approach improves task performance regardless of the underlying model architecture, emphasizing its general effectiveness. The degree of improvement in accuracy varies across the models, which can be referred to as architectural differences, such as the number of layers, parameter counts, and training strategies. For example, Qwen-2.5-14b, being a significantly larger model than Llama-3.8b in terms of layers, parameters, etc. achieves higher accuracy, although it already performs well in the baseline setup due to its inherent capacity. Also, the number of unverified facts produced by GPT-3.5 is notably lower than that of Qwen and Llama, showing that GPT-3.5 generates outputs that are more aligned with the requested format and are easier to verify.

Conclusion and Future Work

In this paper, we proposed a hybrid framework to improve geospatial question answering by integrating LLMs with structured knowledge from Wikidata. The framework combines fact generation, verification, and retrieval to enhance the accuracy and reliability of LLMs in reasoning about geospatial relationships. By leveraging Wikidata for fact verification and a RAG pipeline, the approach addresses key limitations of LLMs in handling geospatial reasoning tasks. Experimental results demonstrated that the proposed framework outperforms zero-shot prompting for all tested models, including GPT-3.5, Llama-3-8b, and Qwen-2.5-14b, underscoring the effectiveness of integrating LLMs with external knowledge sources in the zero-shot manner. While the proposed framework demonstrates significant improvements in geospatial QA accuracy, it is the first step towards improving zero-shot geospatial QA, and further exploration of complementary approaches such as few-shot prompting, chain-of-thought reasoning (CoT), fine-tuning remains unexplored. Besides, there are still several promising directions for future work, such as focusing on developing methods tailored to handle complex, multi-step geospatial reasoning tasks that go beyond simple fact retrieval, integrating more knowledge graphs or domain-specific databases, and optimizing the retrieval and verification processes, to accommodate larger datasets and better models. In other words, future research could explore broader types of geospatial reasoning beyond simple coordinate comparisons and incorporate diverse external sources to enhance fact retrieval.

Acknowledgments

This work is supported by Kansas State University through the Game-changing Research Initiation Program (GRIP), as part of the 'Towards a Global Food Systems Data Hub: Seeding the Center for Sustainable Wheat Production' project.

References

Adam, D.; and Kliegr, T. 2024. Traceable LLM-based validation of statements in knowledge graphs. *arXiv preprint arXiv:2409.07507*.

Bowman, S. R.; Hyun, J.; Perez, E.; Chen, E.; Pettit, C.; Heiner, S.; Lukošiūtė, K.; Askell, A.; Jones, A.; Chen, A.; et al. 2022. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*.

Chen, C.; and Shu, K. 2023. Can LLM=-generated misinformation be detected? *arXiv preprint arXiv:2309.13788*.

Ehrlinger, L.; and Wöß, W. 2016. Towards a definition of knowledge graphs. In *SEMANTiCS (Posters, Demos, SuCCESS)*, volume 48, 1–4.

Feng, Y.; Ding, L.; and Xiao, G. 2023. Geoqamap-geographic question answering with maps leveraging LLM and open knowledge base (short paper). In *12th International Conference on Geographic Information Science (GIScience 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Jiang, B.; Xie, Y.; Hao, Z.; Wang, X.; Mallick, T.; Su, W. J.; Taylor, C. J.; and Roth, D. 2024. A Peek into Token Bias: Large Language Models Are Not Yet Genuine Reasoners. *arXiv preprint arXiv:2406.11050*.

Kim, J.; Kwon, Y.; Jo, Y.; and Choi, E. 2023. Kg-gpt: A general framework for reasoning on knowledge graphs using large language models. *arXiv preprint arXiv:2310.11220*.

Li, J.; Yuan, Y.; and Zhang, Z. 2024. Enhancing LLM factual accuracy with rag to counter hallucinations: A case study on domain-specific queries in private knowledge-bases. *arXiv preprint arXiv:2403.10446*.

Liu, A.; Sheng, Q.; and Hu, X. 2024. Preventing and detecting misinformation generated by large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 3001–3004.

Liu, X.; Wu, F.; Xu, T.; Chen, Z.; Zhang, Y.; Wang, X.; and Gao, J. 2024. Evaluating the Factuality of Large Language Models using Large-Scale Knowledge Graphs. *arXiv preprint arXiv:2404.00942*.

Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Manvi, R.; Khanna, S.; Mai, G.; Burke, M.; Lobell, D.; and Ermon, S. 2023. GeoLLM: Extracting geospatial knowledge from large language models. *arXiv preprint arXiv:2310.06213*.

Mooney, P.; Cui, W.; Guan, B.; and Juhász, L. 2023. Towards understanding the geospatial skills of ChatGPT: Taking a geographic information systems (GIS) exam. In *Proceedings of the 6th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, 85–94.

Peng, B.; Galley, M.; He, P.; Cheng, H.; Xie, Y.; Hu, Y.; Huang, Q.; Liden, L.; Yu, Z.; Chen, W.; et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Sartori, G.; and Orrù, G. 2023. Language models and psychological sciences. *Frontiers in Psychology*, 14: 1279317.

Shojaee-Mend, H.; Mohebbati, R.; Amiri, M.; and Atarodi, A. 2024. Evaluating the strengths and weaknesses of large language models in answering neurophysiology questions. *Scientific Reports*, 14(1): 10785.

Tyen, G.; Mansoor, H.; Chen, P.; Mak, T.; and Cărbune, V. 2023. LLMs cannot find reasoning errors, but can correct them! *arXiv preprint arXiv:2311.08516*.

Vu, T.; Iyyer, M.; Wang, X.; Constant, N.; Wei, J.; Wei, J.; Tar, C.; Sung, Y.-H.; Zhou, D.; Le, Q.; et al. 2023. FreshLLMs: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214*.

Wang, Y.; Reddy, R. G.; Mujahid, Z. M.; Arora, A.; Rubashevskii, A.; Geng, J.; Afzal, O. M.; Pan, L.; Borenstein, N.; Pillai, A.; et al. 2023. Factcheck-GPT: End-to-End Fine-Grained Document-Level Fact-Checking and Correction of LLM Output. *arXiv preprint arXiv:2311.09000*.

Zhang, Y.; Wei, C.; Wu, S.; He, Z.; and Yu, W. 2023. GeoGPT: understanding and processing geospatial tasks through an autonomous GPT. *arXiv preprint arXiv:2307.07930*.

Zhao, R.; Li, X.; Joty, S.; Qin, C.; and Bing, L. 2023. Verify-and-Edit: A Knowledge-Enhanced Chain-of-Thought Framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5823–5840.