ChemLit-QA: A human evaluated dataset for chemistry RAG tasks

Geemi P. Wellawatte[a‡]
LIAC, EPFL

Huixuan Guo[b‡§]
NTU Singapore

Magdalena Lederbauer[d§]
ETH Zurich

Anna Borisova[a]
LIAC, EPFL

Matthew Hart[c§]
UNC Chapel Hill

Marta Brucka[a]
LIAC, EPFL

Philippe Schwaller[*a]
LIAC, EPFL
philippe.schwaller@epfl.ch

Abstract

Retrieval-Augmented Generation (RAG) is a widely used strategy in Large-Language Models (LLMs) to extrapolate beyond the inherent pre-trained knowledge. Hence, RAG is crucial when working in data-sparse fields such as Chemistry. The evaluation of RAG systems is commonly conducted using specialized datasets. However, existing datasets, typically in the form of scientific Question-Answer-Context (QAC) triplets or QA pairs, are often limited in size due to the labor-intensive nature of manual curation or require further quality assessment when generated through automated processes. This highlights a critical need for large, high-quality datasets tailored to scientific applications. We introduce ChemLit-QA, a comprehensive, expert-validated, open-source dataset comprising over 1,000 entries specifically designed for chemistry. Our approach involves the initial generation and filtering of a QAC dataset using an automated framework based on GPT-4 Turbo, followed by rigorous evaluation by chemistry experts. Additionally, we provide two supplementary datasets: ChemLit-QA-neg focused on negative data, and ChemLit-QA-multi focused on multihop reasoning tasks for LLMs, which complement the main dataset on hallucination detection and more reasoning-intensive tasks.

1 Introduction

Over the last few years, we have seen an increased interest in developing and using Large Language Models (LLMs) to accelerate scientific discovery.[1–4] While LLMs perform well in general knowledge domains, they often fall short in specialized, knowledge-intensive scientific fields such as chemistry.[5,6] This is because relevant scientific data may not be adequately represented in the training corpora of foundational models, let alone be accurate and up-to-date. As a result, LLMs are prone to hallucinations and may generate false information based on general knowledge given queries in scientific domains.[7–10]

A widely adopted strategy to address these limitations is Retrieval-Augmented Generation (RAG), which enhances LLMs by enabling them to retrieve information from external knowledge sources based on semantic similarity.[11] Many studies have demonstrated that RAG improves the reliability of answers when addressing domain-specific queries.[11–13] In addition to RAG, other common approaches to improve an LLM's performance for generating factually correct text are through fine-tuning[3,14,15], in-context learning[16,17] and, using Language-Interfaced Fine-Tuning (LIFT)[18,19]. However, there are currently only a few standard datasets available to evaluate these models effectively.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

The most common RAG-specific datasets are in the form of Question-Answer (QA) pairs or Question-Answer-Context (QAC) triplets. These datasets can be curated either manually or automatically using advanced LLMs. For example, the Stanford Question Answering Dataset (SQuAD) was curated with crowdsourcing[20] while HotpotQA[21] relied on the crowdsourcing website Amazon Mechanical Turk. While manually curated data is typically logically cohesive, comprehensible, and reasonable, the curation process is resource-intensive. This challenge is amplified when curating domain-specific, knowledge-intensive scientific datasets, as it requires domain experts to thoroughly understand literature and create QA pairs or QAC triplets. As a result, manually-curated scientific datasets tend to be much smaller. For example, LitQA[22] contains 50 MCQ-type questions across the domain of biology.

LLM-assisted, automated dataset curation thus becomes an increasingly attractive solution as exemplified by MultiHop-RAG[23] and RGB[24]. However, these datasets are based on general knowledge. Switching the sources to scientific data often results in LLMs being incapable of generating useful content.[8,25] For instance, long and difficult texts that contain chemical language, chemical property evaluations, and reaction protocols tend to cause LLMs to misunderstand, hallucinate and generate factually incorrect QA pairs or QAC triplets.[26] In addition to hallucinating[27], previous studies have shown that LLMs can propagate inaccurate and biased knowledge as learned from their pre-training data.[28,29]

Most automatically generated scientific datasets rely on LLM-independent algorithms to generate fixed-formed questions from a specific type of scientific information, such as protein sequence information, NMR spectrums, etc. A few examples are SeqQA and DbQA in LAB-Bench[30], ScienceQA[31], and the semi-programmatically curated questions in ChemBench[32]. Even with the simplified conditions, heavy human supervision is still required.

Recent developments in prompt engineering techniques[33] and improved capabilities of LLMs[34,35] have facilitated progress in automated, LLM-based scientific dataset generation. For instance, the authors of SciQAG[36] successfully applied an automated LLM-based framework to scientific literature, which uses an expert-tuned prompt to generate 10 insightful questions from each paper. However, no context is provided with the QAs, making their credibility difficult to trace. Further, the information distribution in scientific literature is often uneven. For instance, several paragraphs might be dedicated to discussing the research results and insights, whilst a considerable proportion of text contains non-relevant information such as references and acknowledgments. As a result, valuable context might have been overlooked when generating the questions.

We propose ChemLit-QA, an open-source, open-ended, expert-validated, large dataset. This dataset was created using an end-to-end generation pipeline specifically for RAG and fine-tuning benchmark tasks in chemistry. As given below, we made multiple improvements to the automated QAC generation pipeline.

- We used a carefully selected, diverse sample from a corpus of published papers in ChemRxiv (`https://chemrxiv.org/`) and parsed each scientific paper into chunks of 2,000 characters. ChemRxiv corpus was selected as it is an open-source database. We adopted the fine-web approach[37] to source the most knowledge-rich partitions (referred to as chunks or context in this work).
- A novel QAC generation workflow was adopted which first identifies the most suitable reasoning types to generate, then invokes independent question generation chains accordingly, each based on a specific instruction-based prompt. This strategy results in high-quality and diverse generated questions.
- In addition to the QAC triplets, we provide "similar chunks" that have the highest Euclidean similarity to the original chunk's embedding vector which was used for generating a QA pair. These similar chunks mimic the results of a retriever in a RAG workflow, enabling us to simulate RAG evaluations efficiently.
- The generated QAC triplets were both automatically filtered using LLM-based and semantic similarity metrics and re-evaluated by 4 experts to curate the ChemLit-QA dataset. A sample of questions that were rejected by the experts was compiled into a challenging negative dataset named ChemLit-QA-neg. All questions in this dataset are consistent with their contexts; however, the answers are either unavailable or could not be inferred from the provided information.

- Based on ChemLit-QA, we further automated the generation of multi-hop questions with bridge entities and curated the ChemLit-QA-multi dataset. We show that these questions are more challenging for LLMs.

We conducted several downstream experiments (RAG and fine-tuning) with a substantial number of State-Of-The-Art (SOTA) LLMs. Our findings demonstrate that we successfully generated human-like QAC triplets that are both knowledge-intensive and context-specific using an automated approach. Fine-tuning LLMs on ChemLit-QA resulted in superior performance compared to their baseline counterparts in a RAG setting. Additionally, we observed that LLMs struggled more with answering questions in the more challenging ChemLit-QA-multi dataset compared to the standard ChemLit-QA dataset. However, no LLM performed satisfactorily on the negative identification task within ChemLit-QA-neg. Specifically, GPT-4o mini, the best-performing model, achieved only a mean Answer Correctness of approximately 60%. Our results, including the datasets, are publicly available at `https://github.com/geemi725/ChemLit-QA`.

2 Related datasets

We present a summary of relevant datasets to this work. The datasets are classified according to three metrics: (a) Is it curated automatically or manually? (b) Does it contain QAC triples to support RAG evaluation? (c) Is it curated from scientific information sources?

SQuAD[20] and Hotpot-QA[21] are both manually curated, non-scientific datasets with QAC triples. The answers are typically simple and limited to a few words. Multihop-RAG[23] and RGB[24] are automatically curated, non-scientific datasets with QAC triples. While both contain questions with simple free-form and null answers, Multihop-RAG includes additional {yes, no} questions. SciQ[38] is a manually curated Multiple-Choice Question set from scientific exams, where each question is supplied with a sentence of evidence. ChemBench[32] comprises a wide range of chemistry-related questions from exams and papers. Part of the questions are generated automatically, but contexts are not provided. PubMedQA[39] is a RAG-compatible dataset focusing on the biomedical field. The dataset is curated semi-automatically based on keywords, with answers fixed to {yes, no, maybe}. SciQAG[36] adopts an LLM-based automatic QA generation method, thus the answers are free-form sentences and short paragraphs. However, contexts are not provided with the answers.

ChemLit-QA (this work) is one of the first datasets containing fully LLM-generated, free-form QAC triples extracted from scientific, in specific, chemistry-related publications.

3 Method: Dataset curation

3.1 Pipeline for automatic generation of QACs

Data chunk preparation We first parsed the entire corpus of ChemRxiv (`https://chemrxiv.org/`) papers until March 2024. Next, we then cleaned the parsed documents in XML format by eliminating XML tags from the main text. Both reference tags and reference numbers were removed for in-text citations. A two-level hierarchy (see SI Fig. S1) was curated from expert knowledge and LLM-generated keywords. We then prompted the Mistral-7B model[40] to assign a first- and second-level label to each paper based on its title and abstract. Five papers were selected from each second-level category as the input to our pipeline to enforce diversity within the generated dataset.

The context of each paper was split recursively into chunks of a maximum length 2,000 characters using LangChain(`https://www.langchain.com/`), which were embedded and stored in a FAISS vector database (`https://faiss.ai/index.html`) for efficient manipulation. Each chunk was classified based on whether it contained useful scientific content. All in-line classifications in the pipeline were performed by GPT-4 Turbo[41].

Data entry generation We proposed seven reasoning types for questions – Explanatory, Comparative, Conditional, Causal, Predictive, Procedural, and Evaluative, to ensure their diversity and reasoning-intensiveness. Their definitions and the instruction-based prompts are available in the SI. For each chunk labeled as useful, we used an independent chain to identify all suitable reasoning types from the chunk. The respective chains were then invoked to generate the QAC triplet, which was collected with the corresponding reasoning type. We then matched the LLM-generated contexts to

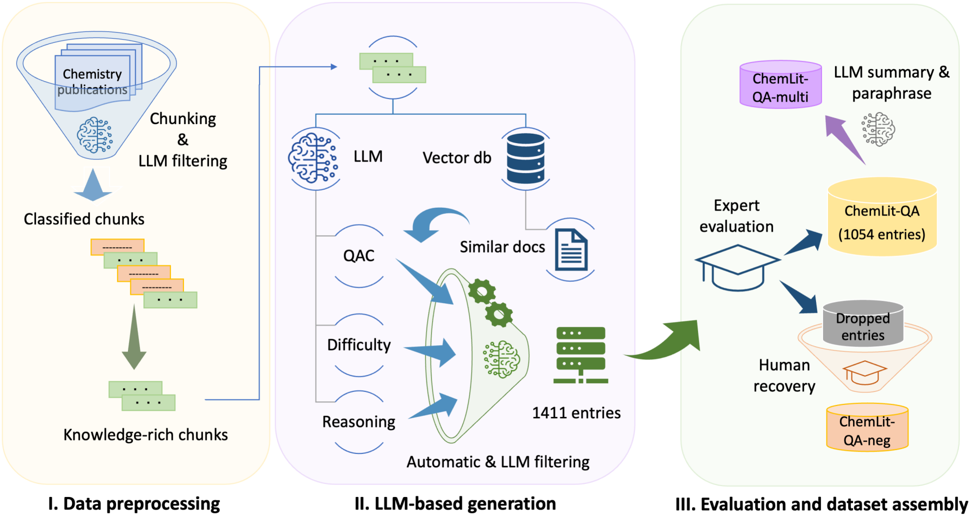


Figure 1: Dataset generation and evaluation pipeline used in this work.

the most similar sentences within the context to determine their start-end indices, which are valuable for downstream fine-tuning purposes. We used SpaCy(https://spacy.io/) for this task. Next, each generated QAC triplet was supplied to an independent chain tasked to label the question-answer difficulty as 'Easy', 'Medium', or 'Hard'. All prompts are available in the SI.

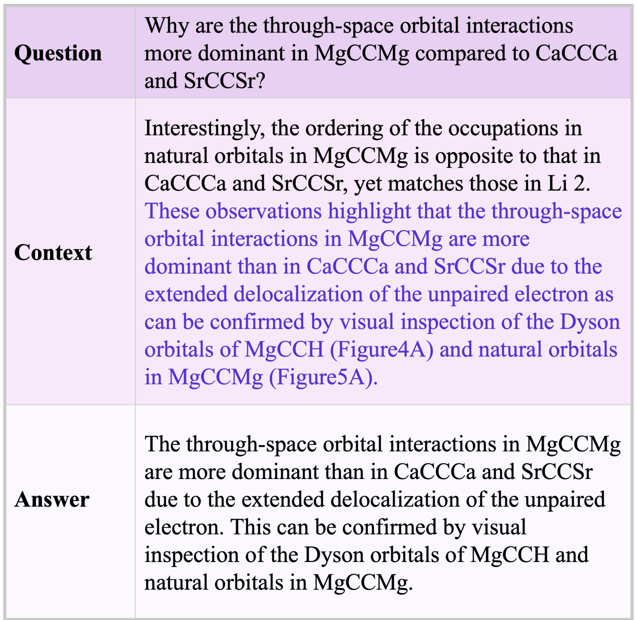
Additionally, we retrieved the top 4 text chunks within the same document that had the highest Euclidean (L2) similarity to the input chunk, using the `similarity_search_by_vector` method in FAISS. We ensured that no chunk was included more than once in the retrieved set. By augmenting a prompt with a context from a QAC triplet and its most similar counterparts, we simulate the retrieval process of a basic retrieval model. This approach streamlines the RAG pipeline evaluation, as demonstrated in Section 3.4. We expect the inclusion of a "similar chunks" section to enhance the benchmarking capabilities of the ChemLit-QA dataset.

Automatic filtering of generated data After running the QAC generation pipeline, we randomly sampled 2,000 entries to keep the workload reasonable for the subsequent expert evaluation process. Then we used 4 LLM-based metrics to evaluate the output: answer relevancy, answer faithfulness, hallucination, and question faithfulness (customized with G-Eval) all in the range $\{0, 1\}$, using DeepEval (https://github.com/confident-ai/deepeval) framework along with GPT-4o. Furthermore, we computed Semantic Entropy (SE)[8] to detect the severeness of LLM confabulation by evaluating how semantically different candidate answers are to a fixed question. In order to better adapt to our task, we added a penalization term which increases with the proportion of contradictory answers to better spot ill-formed questions asking for non-existent entities or comparisons.

Finally, we removed the QAC triplets according to the following conditions. Considering the correlation between difficulty and pSE, a separate filter was applied to each difficulty category.

- Question/answer faithfulness and answer relevancy scores $< \mu - 0.5 \times \sigma$ where μ: mean and σ: standard deviation.
- Hallucination score > 0.1
- pSE $> Q_3 + 1.5 \times I_{QR}$, where Q_3: 75^{th} percentile in the distribution of the respective difficulty, and I_{QR}: difference between the 75^{th} and 25^{th} percentile.

After dropping 589 entries, the dataset contained 1,411 entries after this filtering step.



Question	Why are the through-space orbital interactions more dominant in MgCCMg compared to CaCCCa and SrCCSr?
Context	Interestingly, the ordering of the occupations in natural orbitals in MgCCMg is opposite to that in CaCCCa and SrCCSr, yet matches those in Li 2. These observations highlight that the through-space orbital interactions in MgCCMg are more dominant than in CaCCCa and SrCCSr due to the extended delocalization of the unpaired electron as can be confirmed by visual inspection of the Dyson orbitals of MgCCH (Figure4A) and natural orbitals in MgCCMg (Figure5A).
Answer	The through-space orbital interactions in MgCCMg are more dominant than in CaCCCa and SrCCSr due to the extended delocalization of the unpaired electron. This can be confirmed by visual inspection of the Dyson orbitals of MgCCH and natural orbitals in MgCCMg.

Figure 2: An example entry from ChemLit-QA. The portion of the context that directly addresses the question is highlighted in purple.

3.2 Expert evaluation of the dataset

The LLM-filtered dataset was randomly and equally split among 4 evaluators. A Streamlit interface(https://discuss.streamlit.io/ was built and used for ease of evaluation. A snapshot of the interface is provided in the SI. We dropped 357 entries that were assigned as "drop" by the experts, then corrected the answer, reasoning type, difficulty, and context according to their responses. The final ChemLit-QA dataset contained 1,054 entries. Figure 2 shows an example entry of the final dataset.

Since there was no overlap in the workloads of each evaluator, we conducted an additional test to assess the level of inter-rater agreement. All 4 experts were asked to evaluate the same subset of 60 entries, evenly drawn from each judge's evaluation task. We defined that the judges a) *Completely Agree* with each other when all provide the same label to a given entry, b) *Almost agree* when 3 out of 4 judges provide the same entry, c) *Partially agree* when the agreement is a 50-50 split, and d) *Disagree* for all other cases.

3.3 Additional datasets

ChemLit-QA-neg We noticed that a large proportion of the dropped questions after human evaluations contained coherent questions and were relevant to the contexts, but accurate answers could not be generated based on the given contexts only. We recovered such QAC triplets into a small negative dataset of 139 entries and rewrote all answers to 'Answer not available from the given context'.

ChemLit-QA-multi We constructed an additional automatic pipeline to produce multi-hop questions based on the cleaned dataset. First, we collected all QACs from the same clusters. Then, we extracted and lemmatized all named entities from the questions and contexts using SpaCy(https://spacy.io/). If there are more than two sentences in the contexts besides the original context from the QAC triplet describing a given question entity, we replaced it with an LLM-generated summary and appended the additional sentences to the context. After that, the Tanimoto similarity between original and rephrased questions based on the proportion of concurrent words was calculated. Rephrased questions with similarity above 0.55 were dropped.

3.4 Downstream experiments

To assess the capabilities of our ChemLit-QA dataset, we designed 2 downstream tasks as shown below. Note that, we created 2 datasets for training and testing with 843 and 211 entries, respectively. These datasets are also available in our GitHub repository.

RAG The goal of this task is to demonstrate that our ChemLit-QA dataset can easily be used for benchmarking RAG models. Here, we comprehensively benchmarked 18 LLMs in an RAG pipeline and compared it with their baseline performances. We compared 2 GPT models[42], 3 Claude models[43], 1 Mistral model[40], 6 Llama models[34,44], 2 Phi-3 models[45,46], 2 Gemini models[47], and 2 Gemma models[48] in this experiment. For the RAG case study, we prompted the LLM with the amplified context (original context + top 4 similar chunks) and the question. In the baseline task, the LLM was only prompted with the question and was asked to generate the answer. The test dataset with 211 entries was used in this experiment.

Fine-tuning In addition to RAG tasks, we demonstrate that ChemLit-QA dataset can be extended to fine-tuning tasks as well. With HuggingFace causal LM (`https://huggingface.co`), we fine-tuned Llama2-7B[44] and Mistral-7B-v0.1[40] models. Additionally, we fine-tuned OpenAI's GPT-4o mini[49] model using the OpenAI Platform (`https://platform.openai.com/docs/overview`).

The models were trained and tested on the previously created splits. However, for fine-tuning and inference, we had to reframe the entries with special prompts and create 4 new datasets (2 for train/test Llama2 & Mistral, 2 for train/test GPT-4o mini). The prompted datasets and scripts can be found in our GitHub repository. We repeated the same procedure and fine-tuned 3 more models (same pre-trained LLMs) with ChemLit-QA-multi dataset as well.

4 Results & Discussion

4.1 Dataset evaluations

Diversity of questions in ChemLit-QA One of the main objectives of this dataset was to ensure that the QAC triplets are diverse in terms of difficulty, reasoning, topics, etc. Figure 3 panel a) illustrates the distribution of difficulty within each reasoning type. We also find that ChemLit-QA remains diverse in topics, covering 7 high-level categories in chemistry, and 12 low-level topics (see SI Fig. S3). The highest percentage of 32.25% of entries belongs to the topic *quantum and theoretical chemistry*.

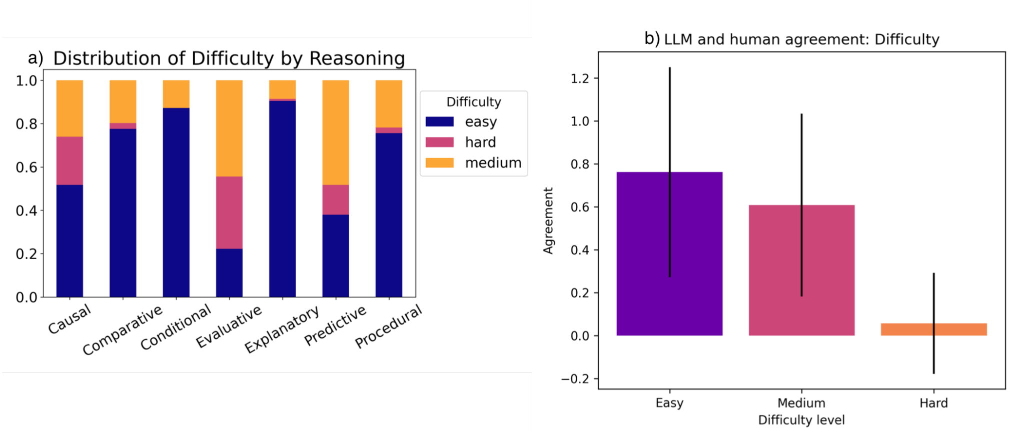


Figure 3: (a) Distribution of reasoning types in the ChemLit-QA dataset categorized by difficulty. The y-axis is normalized for clarity. (b) Agreement between humans and LLM on question difficulty.

The diversity of ChemLit-QA is also featured in the wide range of question keywords and phrases, as plotted in the SI Fig. S4. Besides questions starting with "why" and "what", a considerable proportion of questions contain context-specific keywords that appear less than 4 times in the entire dataset, and are classified under "OTHERS".

We found the distribution of easy, medium, and hard questions in the final dataset to be 74%, 18% and 8% respectively. The top 3 reasoning types in the dataset are Explanatory, Causal, and Comparative with 420, 265, and 187 entries, respectively. Panel a) in Figure 3) demonstrates the spread of difficulty categorized by the reasoning. Specifically, with 39.8%, explanatory questions to make up a majority of the dataset, as answers are explicitly mentioned in the context and thus easier to find. Meanwhile, causal or evaluative questions, which in total constitute 28.5% of the dataset, tend to be harder, as reasoning and inference are required to answer them.

Human and model agreements The agreement test statistics between human experts are available in the SI Tab. S1 and SI Fig. S5. We see that 84%-95% times humans either completely or substantially agree with each other. With such a high proportion of mutual agreement, we can be confident in the credibility of the human evaluation results.

We subsequently compared expert evaluations on the full ChemLit-QA dataset with the LLM-generated results. As seen in Figure 3 panel b), we found that the LLM and humans agree on 76%, 61%, and 6% for easy, medium, and hard questions, respectively. Furthermore, we observe that while LLMs and humans agree on most reasoning types, a significant percentage of disagreements can be found in the explanatory classification. See SI Fig. S6 for a more detailed comparison. Often, LLMs tend to misclassify such questions as Predictive, Conditional, or Causal. These disagreements highlight that expert validation remains crucial in curating a high-quality, knowledge-intensive QA dataset.

4.2 Downstream experiments

RAG: benchmarking LLMs on ChemLit-QA We compared the Answer Correctness score of all models in both RAG and baseline settings. Gemma-2b and Phi3 were removed from the comparison as they consistently failed to generate answers in the required format, leaving 16 models. We adopt the same nomenclature as Olama. (`https://ollama.com/`) As shown in Figure 4, our results agree with the previous findings that a RAG approach is better at generating factual answers than relying on the model's internal knowledge.[11,12,50,51]

All models demonstrated a significant performance increase when provided with contextual information (RAG setting). Notably, among the top 5 models, 3 were Claude-3[43] pre-trained models, and the other 2 were GPT-4o mini[49] and Gemini-1.5 Flash[47]. However, none of these models ranked among the top performers in the baseline tasks. A closer examination of the generated outputs revealed that these models are more adept at avoiding answers when they are uncertain, thereby reducing the likelihood of producing incorrect responses.

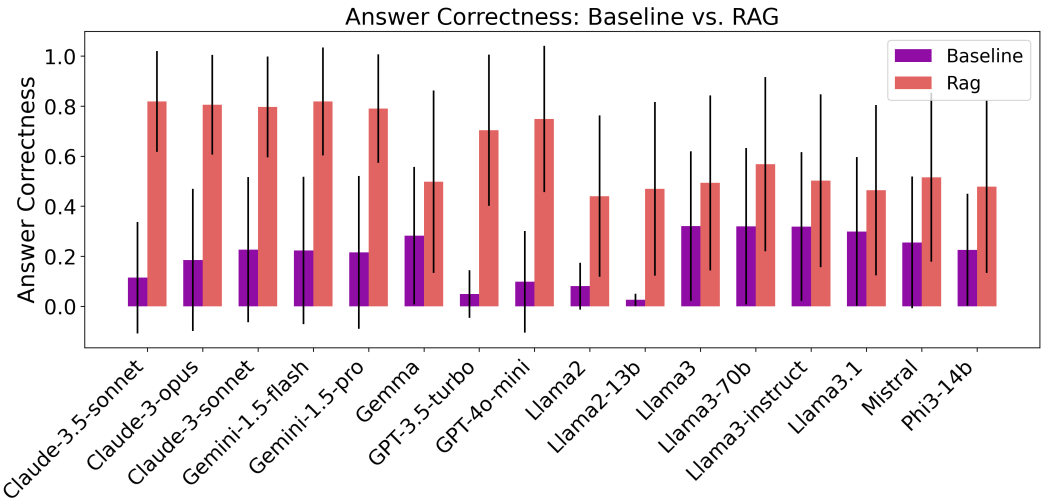


Figure 4: Mean Answer correctness scores of RAG and baseline models. Evaluated on a test dataset with 211 entries. Error bars demonstrate the standard deviation.

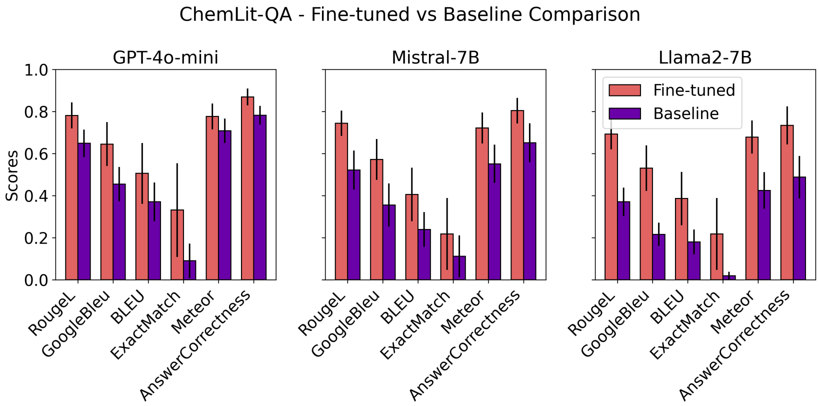


Figure 5: Comparison between baseline and fine-tuned performance on the test dataset for GPT-4o mini, Mistral-7B, and Llama2-7B. The error bars in the figure refer to the standard deviation across the dataset.

Fine-tuning on ChemLit-QA We used the ChemLit-QA dataset to fine-tune 3 LLM models (Llama2-7B[44], Mistral-7B[40] and GPT-4o mini[49]) as discussed in section 3.4. Figure 5 illustrates the results of the models fine-tuned on ChemLit-QA. As found in previous studies[52,53], we demonstrate that fine-tuning improves the inherent performance of an LLM. Specifically, we see a significant increase in performance for Llama2 compared to the other two models, with the mean Answer Correctness increasing from 0.49 ± 0.10 to 0.73 ± 0.10.

Results of the models fine-tuned on the ChemLit-QA-multi dataset can be found in the SI Fig. S8. Despite retaining the same order in performance, all models showed a decrease across all metrics, which proves that ChemLit-QA-multi is more challenging than the main dataset. Interestingly, fine-tuning GPT-4o mini on this dataset did not significantly improve its performance. In contrast, Mistral-7B showed a significant improvement, bringing its final performance comparable to GPT-4o mini.

Hallucination detection on ChemLit-QA-neg One of the limitations of LLMs is hallucinating answers, specifically when they cannot be deduced based on the given contexts. To evaluate an LLM's ability to successfully avoid hallucinations, we performed a test case with our ChemLit-QA-neg dataset. We expect the LLMs to identify negative questions by either responding "Answer not available from the context" as prompted or arriving to the same conclusion through reasoning. We used the LLM-based Answer Correctness metric to capture this expected behavior. A threshold of 0.7 was applied, where all scores below 0.7 were assumed to represent hallucinated answers and were set to 0.

As shown in Figure 6, we discovered that only GPT-4o mini[49], Claude-3.5-Sonnet[43], and Cladue-3-Opus[43] significantly outperformed most open-source models, with GPT-4o mini achieving the highest mean Answer Correctness of 0.58 ± 0.41. Interestingly, Llama3.1 and Llama3[34] are among the worst performers, agreeing with the evaluations with CyberSecEval2[54]. The results suggested that Llama3 models, while demonstrating outstanding language generation performances, are more prone to hallucination and less capable of causal inference, such as identifying the logical implications between sentences, and intensive reasoning in RAG tasks.

5 Conclusion

In this work, we propose ChemLit-QA, a large (1,054 entries), expert-evaluated, open-source, open-ended scientific QAC dataset. Besides including the context from which the questions are generated, we provide an additional field with semantically similar chunks to the input chunk. These similar chunks help the users to easily simulate a RAG pipeline without the need for an external retriever, and further facilitates the independent evaluation of both retrieval and generation components of RAG systems.

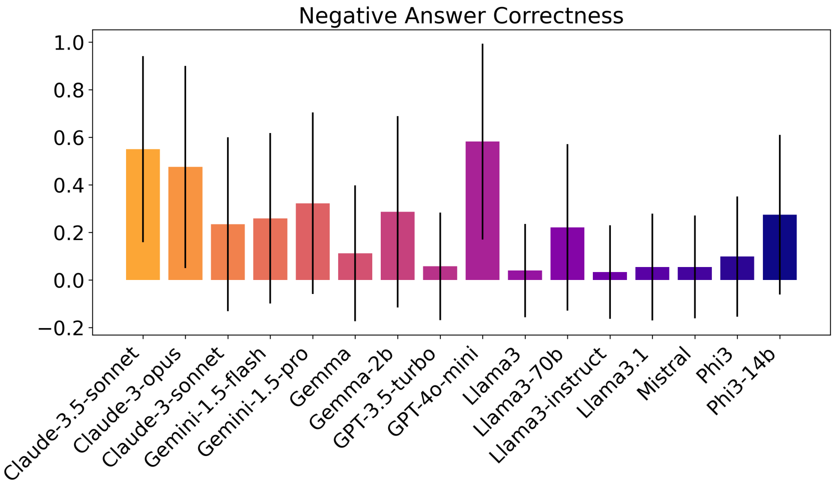


Figure 6: Mean Answer correctness of LLM tested models on ChemLit-QA-neg.

We compared a substantial number of SOTA proprietary and open-source LLMs on the ChemLit-QA for RAG and fine-tuning tasks. We discovered that proprietary models such as GPT-4o mini[49], Claude-3.5-sonnet[43], and Gemini-1.5 Flash[47] outperform open-source models in RAG tasks and rely less on internal knowledge when answering. Although achieving slightly inferior performance using the pre-trained models, the fine-tuned Mistral-7B[40], Llama2-7B[44], Mistral-7B and Llama2-7B achieved performance comparable to GPT-4o mini[49].

We also present and evaluate 2 additional smaller datasets; ChemLit-QA-neg and ChemLit-QA-multi. Surprisingly, no model performed satisfactorily on ChemLit-QA-neg, a dataset for detecting hallucinations in scientific LLMs. This implies that LLMs are yet to achieve human performance on tasks such as distinguishing between arguments, supporting evidence, and reasons in complicated scientific texts.

On the other hand, ChemLit-QA-multi complements the main dataset by providing a more reasoning-intensive and difficult evaluation dataset for fine-tuning, which is proven by the fact that all models performed worse on ChemLit-QA-multi than ChemLit-QA.

Limitations and follow-ups of this work include:

- Only chunks of plain text were used in the generation pipeline with a fixed size of 2'000 characters. We did not investigate the impact of the splitting method or the chunk size in this work. However, we expect advanced text splitting methods, such as splitting by semantic meaning of sentences to have an impact.
- Our pipeline is limited to textual information in published work. However, a significant amount of information can be found in figures and tables, especially in the context of scientific publications. We highlight that a multi-modal data extraction approach is highly valuable, such as incorporating special tokens to represent reaction equations, chemical formulae, and SMILES strings.[32]
- As we observed from the human agreement test, there remains ambiguity and overlap in the assignment of reasoning types despite our effort to define them. More explicit categorizing systems might contribute to resolving these conflicts and further diversify the questions.
- The ChemLit-QA-multi dataset includes only multi-hop questions which require infering a bridge entity. Further research could be performed to develop methods for automating free-form multi-hop questions, which could potentially be more challenging and close to the real-world use case of scientific RAG systems.

In conclusion, with ChemLit-QA, ChemLit-QA-neg and ChemLit-QA-multi, we aim to lessen the need for high-quality datasets in benchmarking the scientific capabilities of LLM models. We expect these datasets to be a valuable addition to the advancement of accelerating scientific discovery.

References

[1] Jaewoong Choi and Byungju Lee. Accelerating materials language processing with large language models. *Communications Materials*, 5(1):13, 2024.

[2] Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

[3] Kevin Maik Jablonka, Philippe Schwaller, Andres Ortega-Guerrero, and Berend Smit. Leveraging large language models for predictive chemistry. *Nature Machine Intelligence*, 6(2):161–169, 2024.

[4] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[5] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, pages 1–11, 2024.

[6] Santiago Miret and NM Krishnan. Are llms ready for real-world materials discovery? *arXiv preprint arXiv:2402.05200*, 2024.

[7] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

[8] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

[9] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020.

[10] Bertalan Meskó and Eric J Topol. The imperative for regulatory oversight of large language models (or generative ai) in healthcare. *NPJ digital medicine*, 6(1):120, 2023.

[11] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

[12] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*, 2021.

[13] Cyril Zakka, Rohan Shad, Akash Chaurasia, Alex R Dalal, Jennifer L Kim, Michael Moor, Robyn Fong, Curran Phillips, Kevin Alexander, Euan Ashley, et al. Almanac—retrieval-augmented language models for clinical medicine. *NEJM AI*, 1(2):AIoa2300068, 2024.

[14] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[15] Wei Zhang, Qinggong Wang, Xiangtai Kong, Jiacheng Xiong, Shengkun Ni, Duanhua Cao, Buying Niu, Mingan Chen, Yameng Li, Runze Zhang, et al. Fine-tuning large language models for chemical text mining. *Chemical Science*, 15(27):10600–10611, 2024.

[16] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021.

[17] Tom B Brown. Language models are few-shot learners. *arXiv preprint ArXiv:2005.14165*, 2020.

[18] Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. *Advances in Neural Information Processing Systems*, 35:11763–11784, 2022.

[19] Mayk Caldas Ramos, Shane S Michtavy, Marc D Porosoff, and Andrew D White. Bayesian optimization of catalysts with in-context learning. *arXiv preprint arXiv:2304.05341*, 2023.

[20] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

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

[22] Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G Rodriques, and Andrew D White. Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*, 2023.

[23] Yixuan Tang and Yi Yang. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*, 2024.

[24] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762, 2023.

[25] Xu Guo and Yiqiang Chen. Generative ai for synthetic data generation: Methods, challenges and the future. *arXiv preprint arXiv:2403.04190*, 2024.

[26] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023.

[27] Weijia Xu, Sweta Agrawal, Eleftheria Briakou, Marianna J Martindale, and Marine Carpuat. Understanding and detecting hallucinations in neural machine translation via model introspection. *Transactions of the Association for Computational Linguistics*, 11:546–564, 2023.

[28] Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*, pages 6565–6576. PMLR, 2021.

[29] Hadas Kotek, Rikker Dockum, and David Sun. Gender bias and stereotypes in large language models. In *Proceedings of the ACM collective intelligence conference*, pages 12–24, 2023.

[30] Jon M. Laurent, Joseph D. Janizek, Michael Ruzo, Michaela M. Hinks, Michael J. Hammerling, Siddharth Narayanan, Manvitha Ponnapati, Andrew D. White, and Samuel G. Rodriques. Lab-bench: Measuring capabilities of language models for biology research, 2024.

[31] Tanik Saikh, Tirthankar Ghosal, Amish Mittal, Asif Ekbal, and Pushpak Bhattacharyya. Scienceqa: A novel resource for question answering on scholarly articles. *International Journal on Digital Libraries*, 23(3):289–301, 2022.

[32] Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Benedict Emoekabu, Aswanth Krishnan, Mara Wilhelmi, Macjonathan Okereke, Juliane Eberhardt, Amir Mohammad Elahi, Maximilian Greiner, Caroline T. Holick, Tanya Gupta, Mehrdad Asgari, Christina Glaubitz, Lea C. Klepsch, Yannik Köster, Jakob Meyer, Santiago Miret, Tim Hoffmann, Fabian Alexander Kreth, Michael Ringleb, Nicole Roesner, Ulrich S. Schubert, Leanne M. Stafast, Dinga Wonanke, Michael Pieler, Philippe Schwaller, and Kevin Maik Jablonka. Are large language models superhuman chemists?, 2024.

[33] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[34] Meta. Meta-llama-3, 2024.

[35] Open AI. Hello gpt-4o, 2024.

[36] Yuwei Wan, Aswathy Ajith, Yixuan Liu, Ke Lu, Clara Grazian, Bram Hoex, Wenjie Zhang, Chunyu Kit, Tong Xie, and Ian Foster. Sciqag: A framework for auto-generated scientific question answering dataset with fine-grained evaluation. *arXiv preprint arXiv:2405.09939*, 2024.

[37] Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*, 2024.

[38] Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions, 2017.

[39] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, 2019.

[40] 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. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[41] Open AI. Gpt-4 turbo in the openai api, 2024.

[42] Open AI. Models, 2023.

[43] Anthropic. Claude 3, 2023.

[44] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[45] Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, Suriya Gunasekar, Mojan Javaheripi, et al. Phi-2: The surprising power of small language models.(2023). *URL https://www. microsoft. com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models*, 2023.

[46] Microsoft. Phi-3, 2024.

[47] Google Deepmind. Gemini, 2024.

[48] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[49] Open AI. Gpt-4o mini: advancing cost-efficient intelligence, 2024.

[50] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[51] Kunal Sawarkar, Abhilasha Mangal, and Shivam Raj Solanki. Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers. *arXiv preprint arXiv:2404.07220*, 2024.

[52] Liang Zhang, Katherine Jijo, Spurthi Setty, Eden Chung, Fatima Javid, Natan Vidra, and Tommy Clifford. Enhancing large language model performance to answer questions and extract information more accurately, 2024.

[53] Jeyoon Yeom, Hakyung Lee, Hoyoon Byun, Yewon Kim, Jeongeun Byun, Yunjeong Choi, Sungjin Kim, and Kyungwoo Song. Tc-llama 2: fine-tuning llm for technology and commercialization applications. *Journal of Big Data*, 11(1):100, 2024.

[54] Manish Bhatt, Sahana Chennabasappa, Yue Li, Cyrus Nikolaidis, Daniel Song, Shengye Wan, Faizan Ahmad, Cornelius Aschermann, Yaohui Chen, Dhaval Kapil, et al. Cyberseceval 2: A wide-ranging cybersecurity evaluation suite for large language models. *arXiv preprint arXiv:2404.13161*, 2024.

SI: ChemLit-QA: A human evaluated dataset for chemistry RAG tasks

Geemi P. Wellawatte[a‡]
LIAC, EPFL

Huixuan Guo[b‡§]
CCEB, NTU Singapore

Marta Brucka[a]
LIAC, EPFL

Anna Borisova[a]
LIAC, EPFL

Matthew Hart[c§]
UNC Chapel Hill

Magdalena Lederbauer[d§]
ETH Zurich

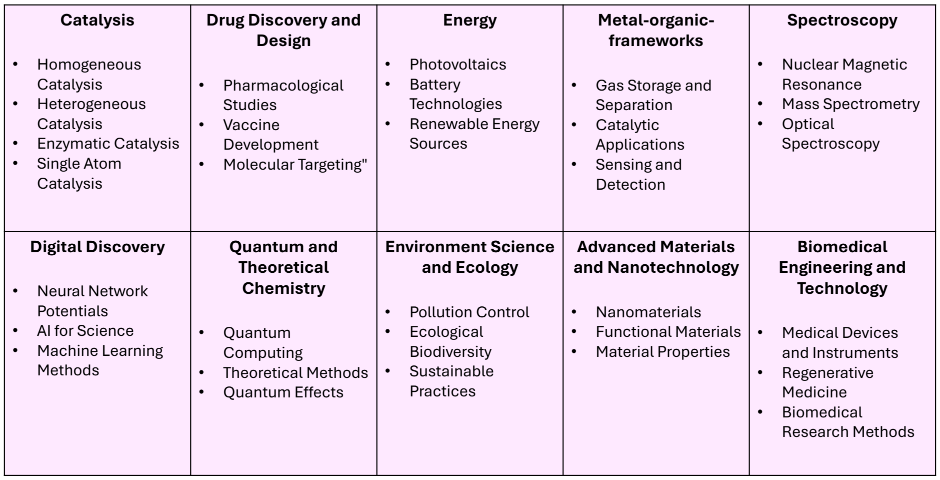
Philippe Schwaller[*a]
LIAC, EPFL
philippe.schwaller@epfl.ch

Abstract

Retrieval-Augmented Generation (RAG) is a widely used strategy in Large-Language Models (LLMs) to extrapolate beyond the inherent pre-trained knowledge. Hence, RAG is crucial when working in data-sparse fields such as Chemistry. The evaluation of RAG systems is commonly conducted using specialized datasets. However, existing datasets, typically in the form of scientific Question-Answer-Context (QAC) triplets or QA pairs, are often limited in size due to the labor-intensive nature of manual curation or require further quality assessment when generated through automated processes. This highlights a critical need for large, high-quality datasets tailored to scientific applications. We introduce ChemLit-QA, a comprehensive, expert-validated, open-source dataset comprising over 1,000 entries specifically designed for chemistry. Our approach involves the initial generation and filtering of a QAC dataset using an automated framework based on GPT-4 Turbo, followed by rigorous evaluation by chemistry experts. Additionally, we provide two supplementary datasets: ChemLit-QA-neg focused on negative data, and ChemLit-QA-multi focused on multihop reasoning tasks for LLMs, further enhancing the resources available for advanced scientific research.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Clusters of papers from ChemRxiv corpus



Catalysis • Homogeneous Catalysis • Heterogeneous Catalysis • Enzymatic Catalysis • Single Atom Catalysis	**Drug Discovery and Design** • Pharmacological Studies • Vaccine Development • Molecular Targeting"	**Energy** • Photovoltaics • Battery Technologies • Renewable Energy Sources	**Metal-organic-frameworks** • Gas Storage and Separation • Catalytic Applications • Sensing and Detection	**Spectroscopy** • Nuclear Magnetic Resonance • Mass Spectrometry • Optical Spectroscopy
Digital Discovery • Neural Network Potentials • AI for Science • Machine Learning Methods	**Quantum and Theoretical Chemistry** • Quantum Computing • Theoretical Methods • Quantum Effects	**Environment Science and Ecology** • Pollution Control • Ecological Biodiversity • Sustainable Practices	**Advanced Materials and Nanotechnology** • Nanomaterials • Functional Materials • Material Properties	**Biomedical Engineering and Technology** • Medical Devices and Instruments • Regenerative Medicine • Biomedical Research Methods

Fig. S 1: Hierarchy of topics and subtopics used to cluster ChemRxiv corpus. We used each paper's title and abstract with Mistral to classify level 1 (shown in bold face) and level 2 labels.

Reasoning distribution in ChemLit-QA

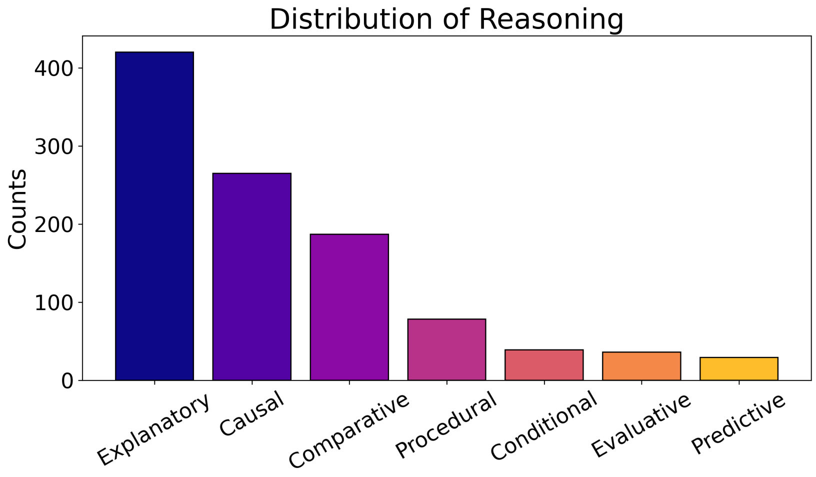


Fig. S 2: Distribution of reasoning in ChemLit-QA.

Distribution of clusters in ChemLlit-QA

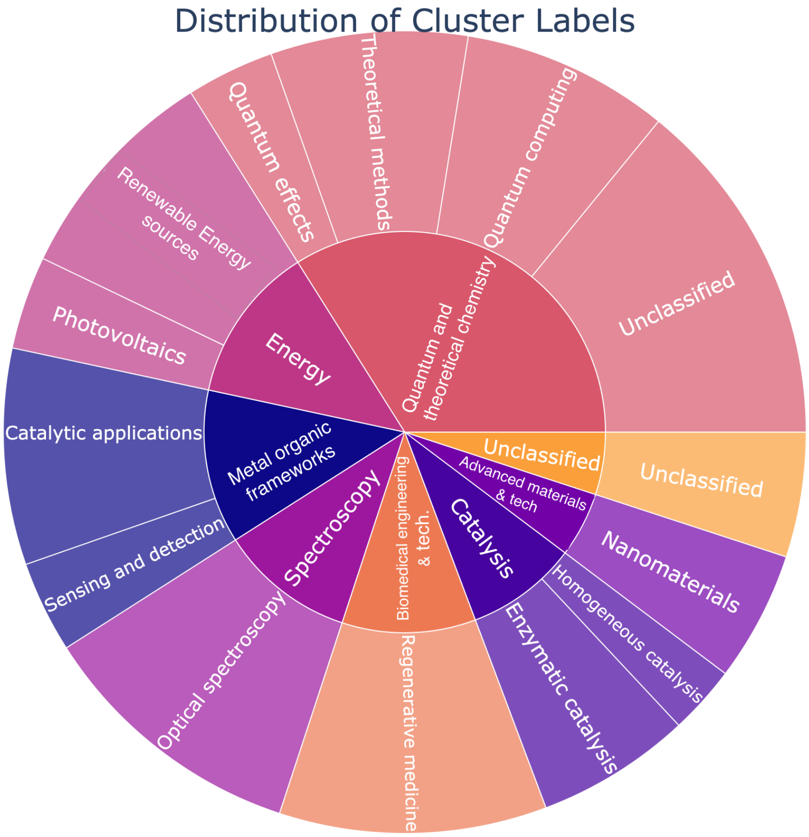


Fig. S 3: Distribution on cluster labels in the ChemLit-QA dataset.

Keyword distribution in ChemLit-QA

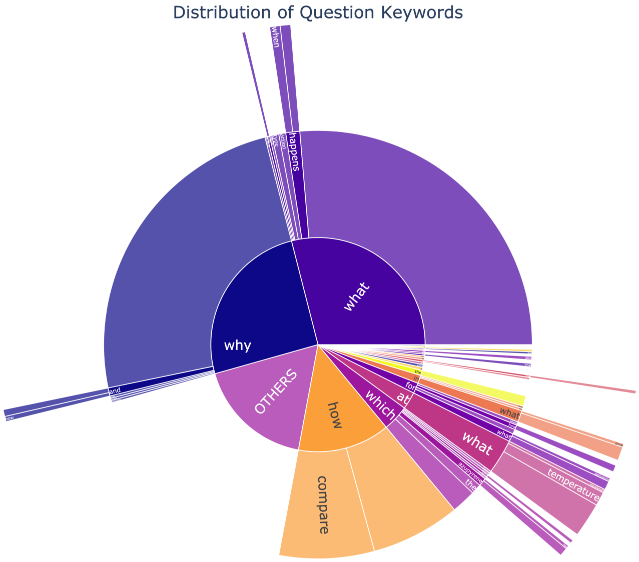


Fig. S 4: Distribution of question keywords in ChemLit-QA.

Expert agreement results

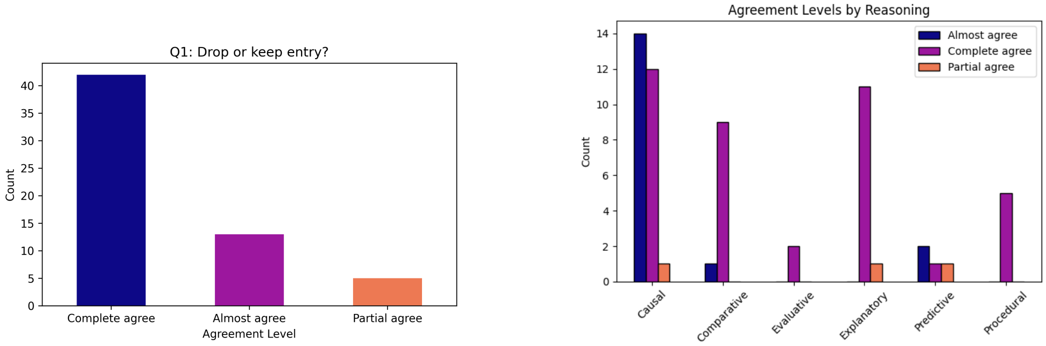


Fig. S 5: Agreement between humans on a) keeping or dropping the dataset entry b) reasoning type

Tab. S 1: Agreement among experts

Task	Complete Agree	Almost Agree	Partial Agree	Disagree
Question quality: Keep or drop	70%	22%	8%	0%
Reasoning type	68%	27%	5%	0%
Difficulty level	44%	40%	8%	8%

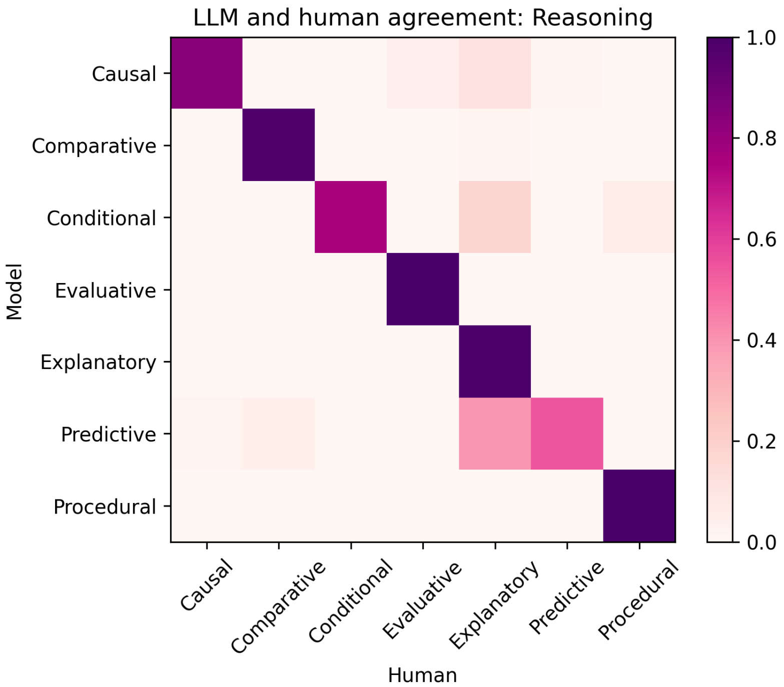


Fig. S 6: Agreement between LLM and experts on answer reasoning type.

Analysis statistics of the ChemLit-QA dataset

Tab. S 2: Statistical distribution of metrics. All of the given LLM-based metrics were implemented using DeepEval[1] framework and GPT-4o[2].

Metric	**Mean ± std dev.**
Answer Relevancy Score (GPT-4o)	0.99 ± 0.02
Faithfulness Score (GPT-4o)	0.99 ± 0.01
Hallucination Score (GPT-4o)	0.0 ± 0.0
Question Faithfulness Score (GPT-4o)	0.93 ± 0.10
Penalized semantic entropy (GPT-4o)	0.20 ± 0.44

Case study: Performance of RAG models in ChemLit-QA dataset

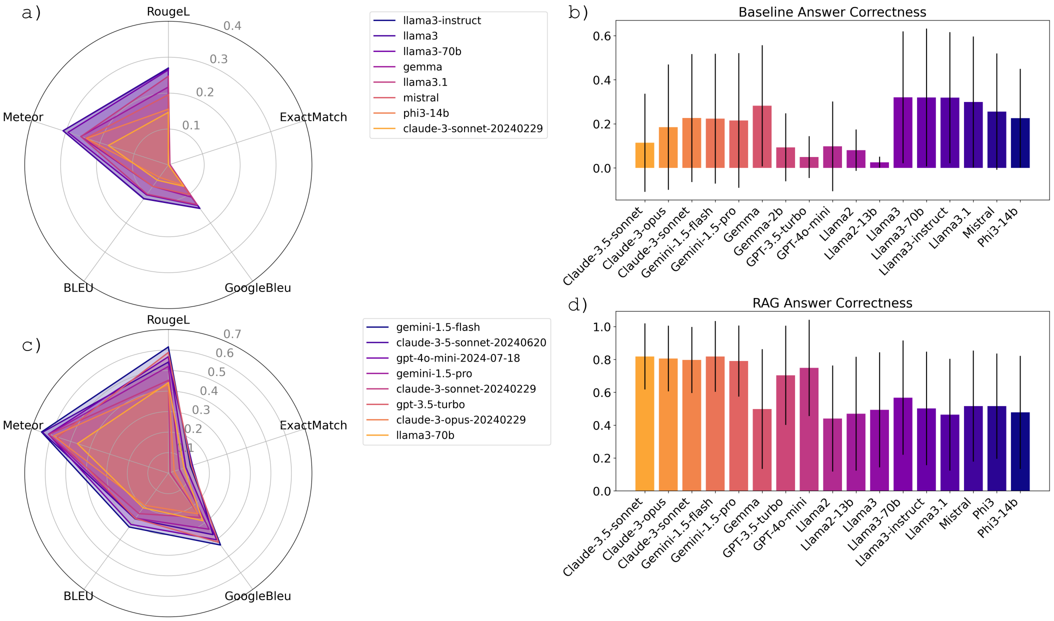


Fig. S 7: (a) The top 8 LLMs' text-based performance on baseline QA. (b) The answer correctness of all tested LLMs on baseline QA. (c) The top 8 LLMs' text-based performance on RAG. (d) The answer correctness of all tested LLMs on RAG.

Case study: Finetuned model performance on ChemLitQA-multi

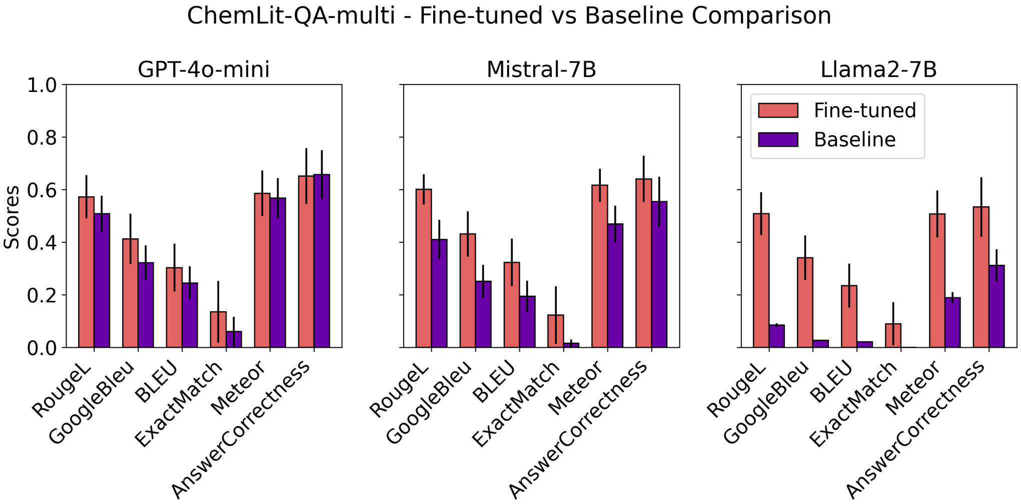
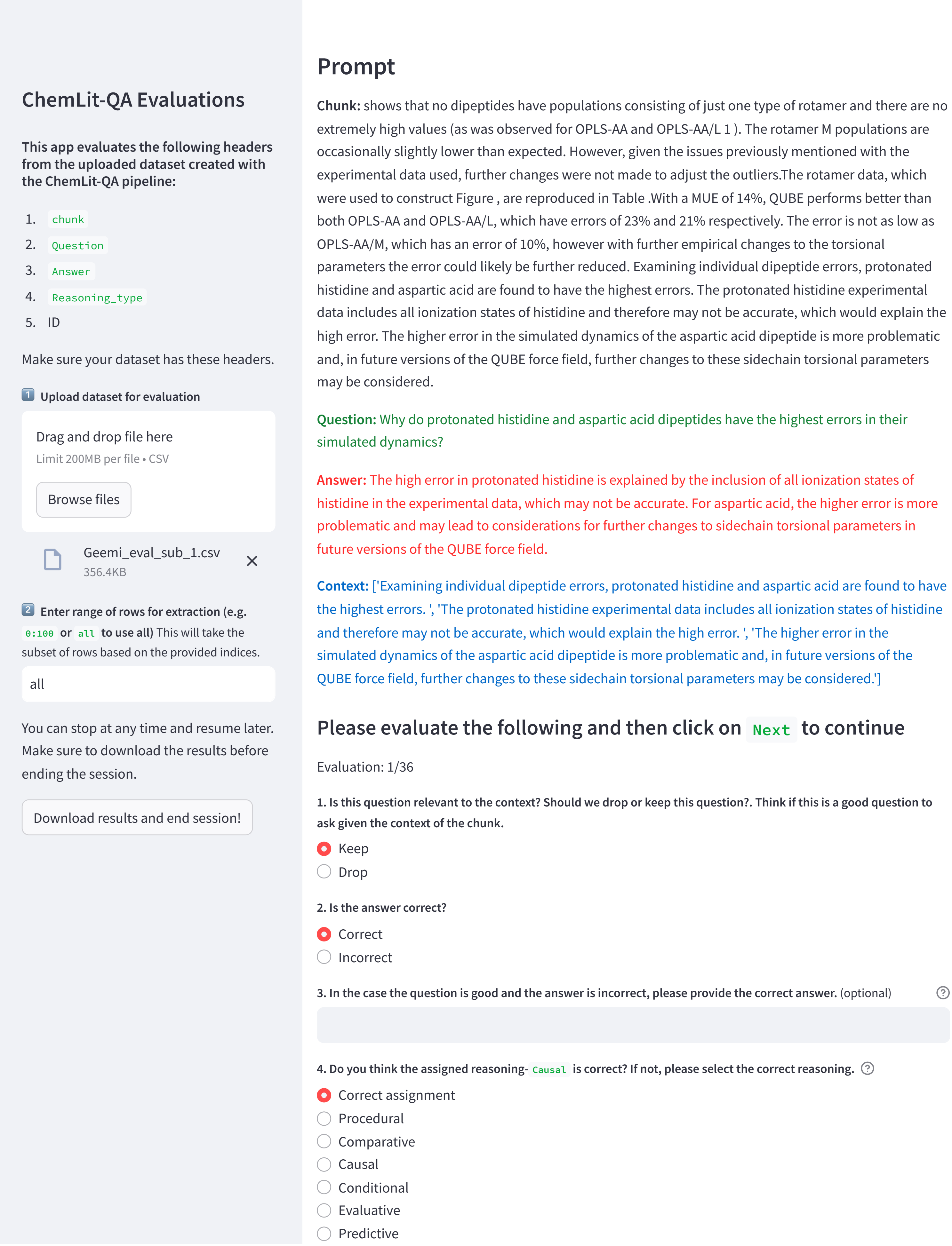


Fig. S 8: Comparison between baseline and fine-tuned performance on the test dataset for GPT-4o-mini, Mistral-7B, and Llama2-7B.

Human evaluations interface

The following figure illustrates the interface used in this work to conduct that human evaluations. This app was developed using Streamlit(`https://discuss.streamlit.io/`). The left hand panel allows the users to upload the dataset under review and select the number of entries to review expert evaluations are collected in the right hand panel.



ChemLit-QA Evaluations

This app evaluates the following headers from the uploaded dataset created with the ChemLit-QA pipeline:

1. `chunk`
2. `Question`
3. `Answer`
4. `Reasoning_type`
5. ID

Make sure your dataset has these headers.

1 Upload dataset for evaluation

Drag and drop file here
Limit 200MB per file • CSV

Geemi_eval_sub_1.csv
356.4KB

2 Enter range of rows for extraction (e.g. `0:100` or `all` to use all) This will take the subset of rows based on the provided indices.

all

You can stop at any time and resume later. Make sure to download the results before ending the session.

Prompt

Chunk: shows that no dipeptides have populations consisting of just one type of rotamer and there are no extremely high values (as was observed for OPLS-AA and OPLS-AA/L 1). The rotamer M populations are occasionally slightly lower than expected. However, given the issues previously mentioned with the experimental data used, further changes were not made to adjust the outliers.The rotamer data, which were used to construct Figure , are reproduced in Table .With a MUE of 14%, QUBE performs better than both OPLS-AA and OPLS-AA/L, which have errors of 23% and 21% respectively. The error is not as low as OPLS-AA/M, which has an error of 10%, however with further empirical changes to the torsional parameters the error could likely be further reduced. Examining individual dipeptide errors, protonated histidine and aspartic acid are found to have the highest errors. The protonated histidine experimental data includes all ionization states of histidine and therefore may not be accurate, which would explain the high error. The higher error in the simulated dynamics of the aspartic acid dipeptide is more problematic and, in future versions of the QUBE force field, further changes to these sidechain torsional parameters may be considered.

Question: Why do protonated histidine and aspartic acid dipeptides have the highest errors in their simulated dynamics?

Answer: The high error in protonated histidine is explained by the inclusion of all ionization states of histidine in the experimental data, which may not be accurate. For aspartic acid, the higher error is more problematic and may lead to considerations for further changes to sidechain torsional parameters in future versions of the QUBE force field.

Context: ['Examining individual dipeptide errors, protonated histidine and aspartic acid are found to have the highest errors. ', 'The protonated histidine experimental data includes all ionization states of histidine and therefore may not be accurate, which would explain the high error. ', 'The higher error in the simulated dynamics of the aspartic acid dipeptide is more problematic and, in future versions of the QUBE force field, further changes to these sidechain torsional parameters may be considered.']

Please evaluate the following and then click on `Next` to continue

Evaluation: 1/36

1. Is this question relevant to the context? Should we drop or keep this question?. Think if this is a good question to ask given the context of the chunk.

● Keep
○ Drop

2. Is the answer correct?

● Correct
○ Incorrect

3. In the case the question is good and the answer is incorrect, please provide the correct answer. (optional) ⓘ

4. Do you think the assigned reasoning- `Causal` is correct? If not, please select the correct reasoning. ⓘ

● Correct assignment
○ Procedural
○ Comparative
○ Causal
○ Conditional
○ Evaluative
○ Predictive
]

○ Explanatory

5. How would you rate the difficulty level of the given Q-A pair? Think an easy question must be answered quickly based on the context.

● Easy
○ Medium
○ Hard

6. Do you think the given context is accurate. ie. does it correlate with the answer?

● Correct
○ Incorrect

7. In case the context is not accurate, please provide the correct context. Context should be complete sentences. (optional) ⓘ

⏮ Previous

Next ⏭

All prompts used in the work

The following figure shows all prompts used during the generation process. Tab. S 3 3 explains the function of each prompt.

Tab. S 3: The function of each prompt in the generation process.

Name	Function
CLEAN_PROMPT	The prompt used for classifying the usefulness of the text chunks
EXAMPLES_USEFUL	Example of a useful text chunk, integrated into CLEAN_PROMPT
EXAMPLES_USELESS	Example of a useless text chunk, integrated into CLEAN_PROMPT
REASONING_PROMPT	The prompt for identifying all possible reasoning types from cleaned text chunks
PROCEDURAL_PROMPT	The prompt for constructing a procedural question
COMPARATIVE_PROMPT	The prompt for constructing a comparative question
CAUSAL_PROMPT	The prompt for constructing a causal question
CONDITIONAL_PROMPT	The prompt for constructing a conditional question
EVALUATIVE_PROMPT	The prompt for constructing a evaluative question
PREDICTIVE_PROMPT	The prompt for constructing a predictive question
EXPLANATORY_PROMPT	The prompt for constructing a explanatory question
DIFFICULTY_PROMPT	The prompt for assign difficulty to a question given its corresponding answer and the original text chunk

Prompts used during dataset curation

```
CLEAN_PROMPT = """Given the following chunk of text from an academic paper,
please classify if the text is useful or not. Output 'Yes' for useful chunks and
'No' for useless chunks.\n
The following are some general traits of useful and useless chunks,
along with some examples. \n

Useful chunks usually: \n
1. Mainly contain coherent English sentences. \n
2. Include one of the following: in-depth discussion scientific entities, coherent
experiment procedures, meaningful comparison, intensive reasoning.

Useless chunks usually: \n
1. Are too short (only one or two sentences). \n
2. Contain non-relevant information to the main text such as title,
author information, figure captions, references, declarations, etc. \n
3. Contain simple introduction to concepts without futher discussions. \n
4. Contain ill-formatted formulae or tables that not readable by humans. \n
5. Simply recorded the authors' experimental procedures without explicit order. \n

Examples of useful chunks: \n
{example_useful}

Examples of useless chunks: \n
{example_useless}


Text to classify: {chunk}

usefulness: Yes or No

Format instructions: \n{format_instructions}
"""


EXAMPLES_USEFUL = """
'd was accurate according to our criteria
(0.8 < K d / K d,inp < 1.25) for r 1 / r 2  2.5 but not for r 1 / r 2  5.
At r 1 / r 2 = 0.25  we obtained a binding isotherm with anomalous shape (Figures S2)
and K d / K d,inp = 1.27. This anomaly was due to a numerical artifact from meshing in
COMSOL; by using a more refined mesh we obtained K d / K d,inp < 1.02.
The improvement in accuracy by mesh refinement may suggest that the large deviations
in K d at r 1 / r 2  5 are also due to too coarse meshes as well.  Thus, more refined
and optimized meshes (in particular, for boundary regions  between small and
large areas) could improve K d determination in a virtual ACTIS experiment.
```

```
We confirmed this for the extreme  value of r 1 / r 2 = 50  and found an  optimal
K d / K d,inp = 1.00 at the expense of excessively increasing  the computational
time ( 72 h instead of  3 h) and the potential risk of overfitting (SI).
In order to keep studies consistent, comparable and in a reasonable time', \n
"""

EXAMPLES_USELESS = """
' ASSOCIATED CONTENT Supporting InformationThe Supporting Information is available
free of charge on the ACS Publications website and on ChemRxiv
(DOI:10.26434/chemrxiv.12345644). Theoretical background for computer simulation
and data evaluation; Simulation of separagrams; Figure , Variation in k off,
inp-separagrams and binding isotherms; Figure , Variation in injection loop
dimensions -separagrams and binding isotherms; Figure , Variation in injection
loop dimensions -sample-plug distribution; Figure , Variation in separation capillary
radii -separagrams and binding isotherms; Figure , Velocity streamlines at different
separation capillary radii; Figure , Variation in the initial', \n
"""


REASONING_PROMPT = """
Please identify all the suitable types of questions to generate
given a piece of text. Your available options are: ['Procedural', 'Comparative',
'Causal', 'Conditional', 'Evaluative', 'Predictive', 'Explanatory']. Please choose
solely from the options. The options are defined as follows:\n

A Procedural question asks about the order between steps in a clearly formulated
procedure. These procedures are often indicated by words such as 'first', 'then',
'finally', followd by actions. \n
A Comparative question asks about the relation between mutual properties of comparable
entities, Common mutual properties include numbers, years, etc. \n
A Causal question asks about the reasons for a specific phenomenon. The phenomenon
can be given implicitly or by explicit clauses such as 'for example'. \n
A Conditional question asks about the possible outcomes given a scenario.
Scenarios are often given by conditional clauses such as 'if', 'when', etc.\n
An Evaluative question asks about the benefits and drawbacks of a given entity.\n
A Predictive question asks for reasonable inference, often on the properties of
entites closely related to but not mentioned in the text. \n
An Explanatory question asks for a component from a statement made in the text.  \n

Text: {text}

Structure your output in the following format:
Process: <Record here in detail how you go though each step of the instruction.>
Reasoning_types: <The reasoning types you chose>
Format instructions: \n{format_instructions}
"""
```

```
PROCEDURAL_PROMPT = """
Please follow the instruction below to formulate a Procedural question based on
the given text. A Procedural question asks about the order between steps in a
\clearly formulated procedure. These procedures are often indicated by words such as
\'first', 'then', 'finally', followd by actions.
You should go through the entire text and form questions only base on complete
\sentences. \n

1. Identify the procedure mentioned in the text. If no processes are mentioned,
skip the following steps and output 'NaN' for <question>, <answer> and <context>.
2. List all steps in the process mentioned by the question in the exact same order
as provided. \n
3. Choose one step (step1) from the process.\n
4. Determine its position in the process. i.e. where is it ranked in the process,
the first, the second, or other?\n
5. Raise a question in the format: What is the <position> step in <summary of the
process>? \n
6. Optionally, choose another step (step2) from the process. Determine the relative
position of step1 to step2.\n
7. Raise a question in the following format: What is the <ordinal, relative position>
step before/after <step2> in <summary of the process>? Replace the original question
with the new one. \n
8. Record the question, answer, and context in the output. <question> should be the
question you raised. <answer> should be step1, rephrased to be grammatically correct
when necessary. <context> should be the original text containing the full process only.


Text: {text}

Structure your output in the following format:
Process: <Record here in detail how you go though each step of the instruction.>
Question: <question>
Answer: <answer>
Context: <context>

Format instructions: \n{format_instructions}
"""

COMPARATIVE_PROMPT = """
Please follow the instruction below to formulate a Comparative question based on the
given text.
A Comparative question asks about the relation between mutual properties of comparable
entities, Common mutual properties include numbers, years, etc.
You should go through the entire text and form questions only base on complete
sentences.\n

1. Identify the comparable entities in the text, the comparable properties,
e.g. numbers, years, etc, and their relation from the text. If there are no comparable
```

```
properties or no relations are mentioned, skip the following steps and output 'NaN'
for <question>, <answer> and <context>.
2. Identify the entities associated with the comparable values. \n
3. Randomly choose at least two entities and raise a question which asks about the
relation between the comparable values of these entities. You shoule not disclose
information on the relation in the question.\n
4. Record the question, answer, and context in the output. <question> should be the
question you raised. <answer> should be the relation you are asking for, including the
result of comparison (e.g. bigger, smaller, similar, etc). Rephrase the answer to be
grammatically correct. <context> should be all sentences in the original text excerpts
describing the entities and their comparable values only. \n


Text: {text}

Structure your output in the following format:
Process: <Record here in detail how you go though each step of the instruction.>
Question: <question>
Answer: <answer>
Context: <context>

Format instructions: \n{format_instructions}
"""

CAUSAL_PROMPT = """
Please follow the instruction below to formulate a Causal question based on the given
text. A Causal question asks about the reasons for a specific phenomenon.
The phenomenon can be given implicitly or by explicit clauses such as 'for example'.
You should go through the entire text and form questions only base on complete
sentences.\n

1. Identify the reasoning and scenario in the text. If no examples are mentioned,
skip the following steps and output 'NaN' for <question>, <answer> and <context>.
2. Rephrase the scenario into a question. Do not add or delete any information. \n
3. Record the question, answer, and context in the output. <question> should be the
question you raised. <answer> should be an explanation of the scenario based on the
reasoning, rephrased to be grammatically correct when necessary.
<context> should be all sentences in the original text containing the claims only.


Text: {text}

Structure your output in the following format:
Process: <Record here in detail how you go though each step of the instruction.>
Question: <question>
Answer: <answer>
Context: <context>
```

```
Format instructions: \n{format_instructions}
"""

CONDITIONAL_PROMPT = """
Please follow the instruction below to formulate a Conditional question based on the
given text.
A Conditional question asks about the possible outcomes given a scenario. Scenarios are
often given by conditional clauses such as 'if', 'when', etc.
You should go through the entire text and form questions only base on complete
sentences.\n

1. Identify the text containing conditions, e.g. clauses with 'if'.  If no conditions
are mentioned, skip the following steps and output 'NaN' for <question>, <answer> and
<context>.
2. Identify the possible scenarios and the corresponding actions. \n
3. Formulate a question which asks for the action given one of the scenarios.
You can choose scenarios not mentioned in the text. \n
4. Record the question, answer, and context in the output. <question> should be the
question you raised. <answer> should be the corresponding action, rephrased to be
grammatically correct when necessary. <context> should be all sentences in the original
text containing the statements only. \n


Text: {text}

Structure your output in the following format:
Process: <Record here in detail how you go though each step of the instruction.>
Question: <question>
Answer: <answer>
Context: <context>

Format instructions: \n{format_instructions}
"""


EVALUATIVE_PROMPT = """
Please follow the instruction below to formulate an Evaluative question based on the
given text.
An Evaluative question asks about the benefits and drawbacks of a given entity. \n
You should go through the entire text and form questions only base on complete
sentences.\n

1. List all statements made in the text. Find if any statements explain the properties
of a specific entity and imply value judgements. Define these statement as 'necessary
statements'. If no statements satisfy the requirements, skip the following steps and
output 'NaN' for <question>, <answer> and <context>.
2. Reformulate the 'necessary statements' in the format: <entity>: <properties> \n
3. Classify the properties as positive or negative. \n
```

```
4. Raise a question based on the format: What are the pros and cons / benefits /
drawbacks of <entity>? Paraphrase the question. \n
5. Record the question, answer, and context in the output. <question> should be the
question you raised. <answer> should contain all <properties> associatd with the
authors' attitude, rephrased to be grammatically correct when necessary. <context>
should be all sentences in the original text containing 'necessary statements'only.


Text: {text}

Structure your output in the following format:
Process: <Record here in detail how you go though each step of the instruction.>
Question: <question>
Answer: <answer>
Context: <context>

Format instructions: \n{format_instructions}
"""

PREDICTIVE_PROMPT = """
Please follow the instruction below to formulate a Predictive question based on the
given text.
A Predictive question asks for reasonable inference, often on the properties of
entites closely related to but not mentioned in the text.
You should go through the entire text and form questions only base on complete
sentences.\n

1. List all statements made in the text. Find if any statements explain the properties
of a specific entity. Define these statement as 'necessary statements'.
If no statements satisfy the requirements, skip the following steps and output
'NaN' for <question>, <answer> and <context>.
2. Randomly choose from the following one category of transformations with equal
probability: \n
    a. Negation \n
    b. Generalization/specification \n
    c. Analogy \n
3. Apply to the most suitable entity-property pair. The transformed entity and
property must both make sense scientifically. \n
4. Raise a question which asks for the property of the transformed entity. Do not
disclose any information about the transformed property in the question. \n
5. Record the question, answer, and context in the output. <question> should be the
question you raised. <answer> should contain <transformed properties> and be
rephrased to be grammatically correct when necessary. <context> should be all s
entences in the original text containing the 'necessary statements' only. \n


Text: {text}
```

```
Structure your output in the following format:
Process: <Record here in detail how you go though each step of the instruction.>
Question: <question>
Answer: <answer>
Context: <context>

Format instructions: \n{format_instructions}
"""


EXPLANATORY_PROMPT = """
Please follow the instruction below to formulate an Explanatory question based on the
given text.
An Explanatory question asks for a component from a statement made in the text.  \n
You should go through the entire text and form questions only base on complete
sentences.\n

1. List all statements made in the text.
2. Choose a statement and replace part of it with an appropriate interrogative pronoun.
The part you replace should be specific. You should not mention the replaced information
in the question. \n
3. Rephrase the question to be grammatically correct. \n
4. Record the question, answer, and context in the output. <question> should be the
question you raised. <answer> should be the part you replaced, rephrased to be
grammatically correct when necessary. <context> should be all sentences in the original
text containing the chosen statement only. \n


Text: {text}

Structure your output in the following format:
Process: <Record here in detail how you go though each step of the instruction.>
Question: <question>
Answer: <answer>
Context: <context>

Format instructions: \n{format_instructions}
"""

DIFFICULTY_PROMPT = """
You are given a text chunk and a question-answer pair derived from the chunk. Please
assign one of the labels from 'Easy', 'Medium' and 'Hard' to <difficulty>, where the
easiest question is one whose answer is directly available in a single sentence in the
chunk, and the hardest question is one which requires information from multiple sentences
in the chunk and complex reasoning to arrive at the answer.

Question: {question}
Answer: {answer}
```

```
Chunk: {chunk}

Structure your output in the following format:
Difficulty: <difficulty>

Format instructions: \n{format_instructions}
"""
```

References

[1] Jeffrey Ip et al. Deepeval: The open-source llm evaluation framework. *Confident AI*, 2024.

[2] Open AI. Hello gpt-4o, 2024.