OMG-QA: Building Open-Domain Multi-Modal Generative Question Answering Systems

Linyong Nan[1] **Weining Fang**[1] **Ailin Rasteh**
Pouya Lahabi **Weijin Zou**[2] **Yilun Zhao**[1] **Arman Cohan**[1]
[1]Yale University [2]LinkedIn
{linyong.nan, weining.fang}@yale.edu

Abstract

We introduce OMG-QA, a new resource for question answering that is designed to evaluate the effectiveness of question answering systems that perform retrieval augmented generation (RAG) in scenarios that demand reasoning on multi-modal, multi-document contexts. These systems, given a user query, must retrieve relevant contexts from the web, which may include non-textual information, and then reason and synthesize these contents to generate a detailed, coherent answer. Unlike existing open-domain QA datasets, OMG-QA requires systems to navigate and integrate diverse modalities and a broad pool of information sources, making it uniquely challenging. We conduct a thorough evaluation and analysis of a diverse set of QA systems, featuring various retrieval frameworks, document retrievers, document indexing approaches, evidence retrieval methods, and LLMs tasked with both information retrieval and generation. Our findings reveal significant limitations in existing approaches using RAG or LLM agents to address open questions that require long-form answers supported by multi-modal evidence. We believe that OMG-QA will be a valuable resource for developing QA systems that are better equipped to handle open-domain, multi-modal information-seeking tasks.

1 Introduction

Modern question answering systems are explored within two primary frameworks. The first framework operates under the premise of a limited context, providing all necessary information to answer queries. This approach, which treats QA as a reading comprehension exercise, assesses the system's ability to extract and interpret information from a given context to formulate responses (Yang et al., 2015; Rajpurkar et al., 2016; Chen et al., 2017; Joshi et al., 2017; Kwiatkowski et al., 2019). Although this method offers a detailed examination of the systems' comprehension and reasoning skills, it relies on the availability of chosen context, limiting its applicability in many real-world scenarios where the context to address the question is not directly available. The second framework, also known as open-domain QA, addresses this limitation by requiring the system to source information from large-scale knowledge sources - such as text corpora, databases or the Internet - in response to any user query (Chen et al., 2017; Lee et al., 2019; Yang et al., 2019; Guu et al., 2020; Lewis et al., 2020a; Zhu et al., 2021). Typically, these systems utilize a two-stage design: a retrieval stage that efficiently identifies broadly relevant contexts from extensive knowledge sources, and a subsequent reading stage that mirrors the closed setting. With advancements in large language models (LLMs), these systems have primarily benefited the reading stage, demonstrating enhanced proficiency in interpreting and reasoning with the retrieved content.

Enhancing open-domain QA systems presents two primary challenges. The first challenge involves enhancing retrieval stage using LLMs while ensuring efficiency and scalability. To tackle this, several studies have integrated LLMs into retrieval frameworks through methods like query expansion, ranking adjustments (Lee et al., 2018; Qi et al., 2019; Zhang et al., 2020; Mao et al., 2021), or embedding extraction for dense retrieval (Seo et al., 2019; Nie et al., 2019; Lee et al., 2019; Guu et al., 2020; Lewis et al., 2020a; Karpukhin et al., 2020; Khattab et al., 2021). The second challenge is enabling QA systems to retrieve and interpret multi-modal content, such as tables, images, and videos. Research efforts to address this have included creating a unified embedding space that allows for the retrieval and ranking of context across different modalities (Li et al., 2019; Lu et al., 2019; Herzig et al., 2020; Yin et al., 2020; Qi et al., 2020; Radford et al., 2021; Liu et al., 2022).

Although there have been many attempts to ad-

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1001–1015

dress these challenges, there remains a notable gap: the lack of a comprehensive benchmark capturing the complexities of real-world tasks and can effectively evaluate these advancements. In response, we introduce Open-domain Multi-modal Generative Question Answering Dataset[1] (OMG-QA). Unlike existing open-domain multi-modal QA datasets (Chen et al., 2020; Talmor et al., 2021; Li et al., 2022; Chang et al., 2022) that primarily feature factoid questions (Fu et al., 2020) and call for concise, single noun-phrase or entity-based answers, OMG-QA challenges QA systems to retrieve and reason across content in various modalities within an open setting, ultimately resulting in the generation of detailed narratives or explanations. Additionally, we implement various types of LLM systems, described in Section 3, which are evaluated by our dataset to assess their ability to retrieve multi-modal content in an open setting.

2 OMG-QA

We define open-domain multi-modal generative question answering as the task of producing a long-form answer a, which is a structured discourse that presents entities and their relationships in response to a question q. This process is based on a large-scale knowledge source K, from which the system must retrieve multiple pieces of evidence $e_1, e_2, \ldots, e_n$ to substantiate the answer. To ensure that the systems generate answers grounded in the retrieved evidence, we also mandate systems to explicitly cite the evidences used within the answer. An example of our dataset is provided in Figure 1.

2.1 Question Collection Methods

The task of collecting questions that require the retrieval of multiple multi-modal evidences from different documents presents substantial challenges. Specifically, the identification of multi-modal content that is relevant and shares a common topic for question generation is complex. To address these challenges, we leverage Wikipedia's extensive and diverse content, which includes texts, tables, and images, and developed two question collection pipelines.

Pipeline 1: Text and Table Modality This pipeline processes the Wikipedia dump to select articles containing substantial text and multiple tables. Using the OpenAI text embedding model `text-embedding-ada-002`, we extract embeddings for article introductions, and articles with high cosine similarity are paired. For each pair, tables with overlapping entities are identified, and initial questions are generated based on these table pairs with the aid of `GPT-4`. These questions are then revised to incorporate both textual and tabular content from the articles. The prompts utilized are provided in Figures 3 and 4 in the Appendix.

Pipeline 2: Integrating Texts, Tables and Images The second pipeline is designed to incorporate texts, tables, and images as evidence sources. We start from a single document and extract its table of contents, which included all titles of sections and subsections, and visually represented the parent-child relationships with structured indentation (as illustrated in Figure 2). Additionally, we identify tables and images within each section by extracting their titles and captions. With this table of contents, `GPT-4` is prompted (see Figure 5 in the Appendix for the prompt) to generate questions that required retrieving content from at least two different modalities within the document.

2.2 Document and Evidence Retrieval Annotation

The questions from both pipelines yield a set of primary documents or evidences. To expand these into a broader set of relevant evidences, a pooling annotation procedure (Buckley and Voorhees, 2004; Voorhees and Tice, 2000; Voorhees, 2002) is employed. This process unfolds through several structured steps: 1) **Collection of Systems Results**: We deploy various systems, as detailed in section 3, which execute queries against the entire Wikipedia, retrieving a preliminary set of documents and evidences; 2) **Creation of the Pool**: Outputs from all systems are combined, undergoing a deduplication process to forge a unified pool of documents and evidences for each query; 3) **Relevance Judgments**: The relevance of each pooled evidence to its corresponding query is evaluated; 4) **Evaluation**: The collected relevance judgments serve as a ground truth to evaluate each system's efficacy.

2.3 Statistics

Utilizing the above question collection pipelines, we gather a total of 1,000 questions, with each pipeline contributing 500 questions. Following the pooling process, we annotated the document and evidence retrieval tasks, with each question linked

[1] Our dataset and code can be found at https://github.com/linyongnan/OMG-QA

to an average of 10 relevant documents and 33 pieces of relevant evidence. Table 8 of the Appendix presents key statistics of our dataset.

3 QA Systems

Constructing an open-domain QA system requires three fundamental components: an index, a retriever, and an answerer. For OMG-QA, we assume that all QA systems in our study first retrieve documents from Wikipedia using established search APIs. Furthermore, we aim to assess the LLM systems' capabilities of retrieving fine-grained content. To this end, we implement an evidence retrieval process in all our QA systems, which involves indexing fine-grained content within documents and deploying corresponding evidence retrievers. Following these two retrieval steps, the system employs an answerer to aggregate, reason, and synthesize various pieces of evidence to produce the final answer. An illustration of our QA systems is provided in Figure 1.

Module Configurations We benchmark our dataset against LLM systems that incorporate several key modules: query rewriter, document retriever, document reranker, evidence indexer, evidence retriever, multi-modal evidence reranker, and answerer. The implementation of different systems is primarily distinguished by variations in the configurations of these components:

- **Document Retriever**: We evaluate the effectiveness of using Wikipedia's own search API[2] versus DuckDuckGo's search API[3], restricted to Wikipedia content.

- **Evidence Indexer**: We explore several methods to index evidence from Wikipedia documents, utilizing a document parser that structures data into a tree with section titles as non-leaf nodes and evidence (text paragraphs, tables, images) as leaf nodes. We extract each leaf node's location and content, creating dense indexes with metadata for efficient search. Our three indexing strategies for multi-modal content include:
  - **Text-Only Index**: Textual representations of non-textual content are created using titles, captions, or synthesized text, which are then embedded for dense retrieval.
  - **Textual-Visual Index**: Separate indices are maintained for textual and image evidence, using respective embedding models for indexing.
  - **Modality-Specific Index**: Distinct indices for each evidence type are created using modality-appropriate embedding models.

- **Evidence Retriever**: We compare four types of evidence retrievers: sparse, dense, generative and hybrid. Detailed descriptions of each type can be found in Section A.1 of the Appendix.

- **Additional Modules**: Query rewriter, document reranker, multi-modal evidence reranker and answerer tasks are implemented by prompting LLMs, which perform tasks requiring semantic interpretation of queries, ranking retrieved documents or evidences, and synthesizing final answers with citation attributions. We have evaluated LLMs including `Llama-3-[8, 70b]` (Meta LLaMA Team, 2024), `Mistral-[7b, 8x7b]` (Jiang et al., 2023), `GritLM-[7b, 8x7b]` (Muennighoff et al., 2024), `GPT-3.5-Turbo`, and `GPT-4`.

Fusion of Multi-modal Evidences When using multiple indices, each with unique indexer and retriever setups, we initially retrieve top-k evidences from each index. To integrate these results, non-textual evidences are transformed into textual format by extracting or synthesizing titles for tables and images. These are then embedded using a unified text embedding model. We re-rank these evidences by comparing their embeddings' proximity to the query's embeddings, selecting the top-k for the final evidence retrieval results.

One-round versus Multi-rounds Retrieval We implemented and evaluated both one-round (RAG) (Lewis et al., 2020a) and multi-round (LLM Agent) retrieval strategies. The one-round strategy follows the procedure depicted in Figure 1 once. Conversely, the multi-round strategy employs episodic memory to record all prior retrieval efforts (Su et al., 2021; Yao et al., 2023; Zhong et al., 2023; Lu et al., 2023; Liu et al., 2023), and includes an evaluation module after each round. This module determines the adequacy of retrieved evidence and guides the refinement of subsequent retrieval efforts through feedback. Due to budget limits, we restrict retrieval to a maximum of three rounds.

[2] https://www.mediawiki.org/wiki/API:Search
[3] https://serpapi.com/duckduckgo-search-api

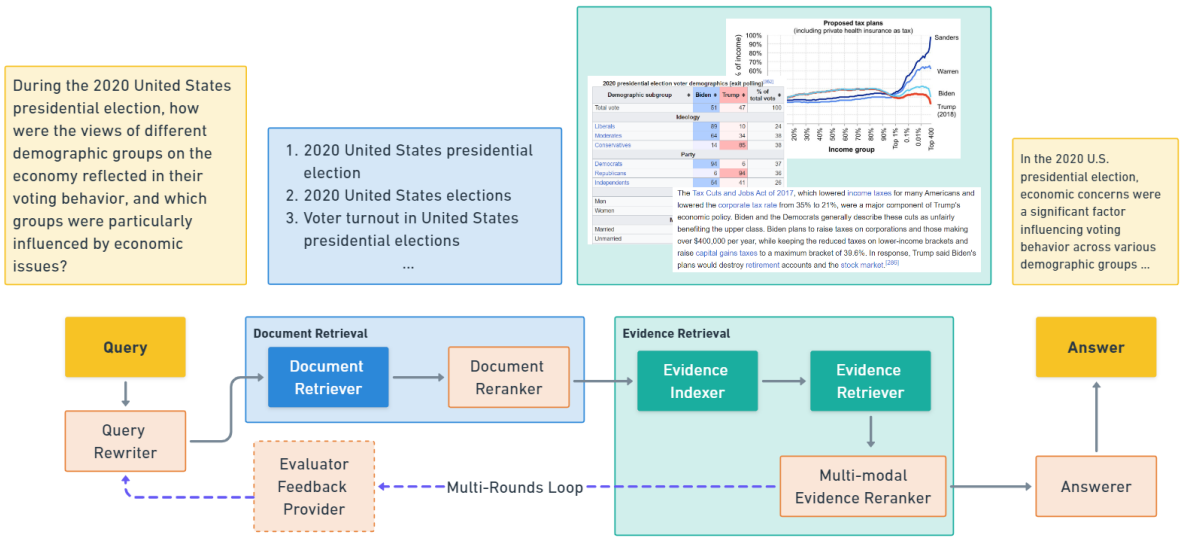

Figure 1: Example of OMG-QA and illustration of modular design of QA systems. Modules in orange color are implemented with LLMs learned in zero-shot.

4 Experiments

4.1 Evaluation

We employed GPT-4 for the following evaluation tasks, with prompts shown in Figures 6-8 of the Appendix: 1) **Evidence Relevancy**: The evaluator determines whether an evidence is relevant and should be retrieved given a query; 2) **Correct Usage of Evidence**: The evaluator assesses whether an answer properly uses all the retrieved evidence from different documents, ensuring consistency in content; 3) **Citation Completeness**: The evaluator checks if all relevant evidence to any content in the answer is cited.

After obtaining these evaluations, we calculate various metrics focusing on different aspects of each system. Using the relevancy labels of all evidence in the pools of testing instances, we compute precision (PER), recall (RER), and F1 (F1-ER) scores for evidence retrieval. Assuming the documents containing all relevant evidence should be retrieved, we also calculate precision (PDR), recall (RDR), and F1 (F1-DR) scores for document retrieval. Additionally, we measure Effective Retrieval Usage (ERU), the proportion of retrieved evidence that is both relevant and accurately used in the generated answer, and Relevance of Used Evidence (RUE), the proportion of evidence cited in the answer that is relevant to the question. For Correct Usage of Evidence (CUE) and Citation Completeness (CCM), we calculate the percentage of instances where the evaluator predicts True.

Document Retrieval	Wikipedia			DuckDuckGo		
	Pre.	Rec.	F1	Pre.	Rec.	F1
Llama-3-8b	0.62	0.26	0.32	**0.71**	**0.32**	**0.40**
Llama-3-70b	0.63	0.27	0.34	**0.70**	**0.35**	**0.41**
Mistral-7b	0.61	0.28	0.34	**0.71**	**0.34**	**0.41**
Mistral-8x7b	0.70	0.28	0.37	**0.71**	**0.33**	**0.40**
GritLM-7b	0.55	0.25	0.31	**0.66**	**0.31**	**0.38**
GritLM-8x7b	0.61	0.27	0.33	**0.67**	**0.31**	**0.37**

Table 1: Comparison of performance of systems with different document retrievers on document retrieval.

4.2 Results

We present the results in Tables 1 to 6, where we analyze the performance impact of varying specific system modules while keeping others constant.

Document Retriever We compare the effectiveness of using Wikipedia's own search API versus DuckDuckGo's search API for document retrieval across different LLM configurations. This comparison takes into account both the quality of the queries and the document retrieval algorithms employed. As demonstrated in Table 1, DuckDuckGo's search API consistently provides superior precision, recall, and F1 scores for document retrieval.

Document Indexing Strategy Next, we evaluate the performance of systems utilizing different indexing strategies, namely text-only, textual-visual, and modality-specific settings. We assess these configurations based on evidence retrieval precision, recall, and F1 scores. For systems with a text-only

Evidence Retrieval	Text-Only			Textual Visual			Modality Specific		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Llama-3-8b	0.43	0.16	0.19	0.52	0.20	0.24	**0.53**	**0.20**	**0.24**
Mistral-7b	0.43	0.17	0.20	**0.52**	**0.21**	**0.25**	0.51	0.21	0.24
GritLM-7b	0.38	0.15	0.17	**0.48**	**0.19**	**0.23**	**0.49**	**0.19**	**0.23**

Table 2: Comparison of performance of systems with different indexers on evidence retrieval.

Evidence Retrieval	Llama-3-8b			Mistral-7b			GritLM-7b		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Sparse	0.38	0.10	0.14	0.35	0.10	0.14	0.31	0.10	0.12
Dense-SFR	0.46	0.20	0.23	0.44	0.21	0.23	0.39	0.18	0.20
Dense-GTE	**0.52**	**0.20**	**0.24**	0.52	0.21	0.23	**0.49**	**0.19**	**0.23**
Dense-Arctic	0.35	0.12	0.15	0.35	0.12	0.15	0.35	0.13	0.15
Generative	0.39	0.12	0.16	0.37	0.13	0.16	0.32	0.11	0.13
Hybrid-SFR	0.45	0.19	0.22	**0.52**	**0.21**	**0.25**	0.39	0.18	0.20
Hybrid-GTE	**0.50**	**0.20**	**0.24**	0.52	0.20	0.24	0.46	0.18	0.21
Hybrid-Arctic	0.34	0.12	0.15	0.39	0.15	0.18	0.31	0.12	0.14

Table 3: Comparison of performance of systems with different text retrievers on evidence retrieval.

index, we report the average scores for various text retrievers compatible with this indexing approach. Table 2 shows that multi-index setups outperform the text-only index in terms of evidence retrieval for our dataset. This enhanced performance is primarily because our dataset demands the retrieval of multi-modal evidences, and a multi-index design facilitates the retrieval of non-textual modalities more effectively.

Evidence Retriever We then proceed to assess the performance of systems equipped with different text retrievers, comparing setups that utilize three distinct types of LLMs. Based on the results shown in Table 3, the sparse retriever exhibits the poorest performance. Both the generative retriever and the dense retriever using the `snowflake-arctic-embed-l` model generally underperform compared to other dense retrievers. Additionally, hybrid retrievers, which narrow the search space to specific sections before dense retrieval, do not demonstrate any clear advantage over the corresponding dense retrievers that retrieve from a broader set of evidences.

LLMs for Retrieval Subsequently, we aim to evaluate the performance of systems that use different LLMs for document and evidence retrieval. Table 4 demonstrates the performance outcomes for document retrieval and evidence retrieval, with various LLMs and two index and retriever configurations. Interestingly, across both indexing and retrieval settings, the choice of LLM appears to have a minimal impact on retrieval performance. The performance disparity between smaller models like `Llama-3-8b` and powerful models such as `GPT-4` is negligible. This suggests that other factors, such as the design of the index or the choice of retrievers, play a more significant role in influencing performance.

One-round versus Multi-rounds Retrieval We now evaluate the performance of systems utilizing either a one-round retrieval or a multi-rounds retrieval process. The results for document and evidence retrieval are shown in Table 5. As anticipated, the multi-rounds retrieval process significantly enhances the recall for document retrieval, thereby improving overall document retrieval outcomes. However, this does not necessarily translate to better results in evidence retrieval; in fact, evidence performance noticeably declines in some cases. We hypothesize that although retrieving a greater number of documents can improve document recall, maintaining the same top-k for evidence retrieval might introduce a significant amount of irrelevant evidence. Each subsequent retrieval round generates new queries for both document and evidence retrievals, and the evidences retrieved in these rounds are ranked according to the latest queries. This ranking process could inadvertently displace previously retrieved evidences that were relevant, resulting in a deterioration of overall evidence retrieval performance.

	Modality-Specific Indexer						Text-Only Indexer w/ Sparse Text Retriever					
	Document Retrieval			Evidence Retrieval			Document Retrieval			Evidence Retrieval		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Llama-3-8b	0.83	0.29	0.40	0.53	0.20	0.24	0.71	0.32	0.40	0.38	0.10	0.14
Llama-3-70b	0.84	0.34	0.44	0.52	0.21	0.24	0.70	0.35	0.41	0.37	0.11	0.14
Mistral-7b	0.86	0.32	0.42	0.51	0.21	0.24	0.71	0.34	0.41	0.35	0.10	0.14
Mistral-8x7b	0.86	0.33	0.43	**0.52**	**0.22**	**0.25**	0.71	0.33	0.40	0.34	0.10	0.14
GritLM-7b	0.84	0.30	0.41	0.49	0.19	0.23	0.66	0.31	0.38	0.31	0.10	0.12
GritLM-8-7b	0.82	0.31	0.41	**0.54**	**0.21**	**0.25**	0.67	0.31	0.37	0.36	0.12	0.14
GPT-35-Turbo	0.81	0.32	0.42	0.50	0.21	0.24	0.69	0.33	0.40	**0.38**	**0.13**	**0.15**
GPT-4	**0.87**	**0.35**	**0.45**	**0.53**	**0.22**	**0.25**	**0.71**	**0.36**	**0.43**	**0.38**	**0.12**	**0.15**

Table 4: Comparison of performance of systems with different LLMs with different indexers and retrievers on document and evidence retrieval

	Document Retrieval						Evidence Retrieval					
	Single-Round			Multi-Rounds			Single-Round			Multi-Rounds		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Llama-3-8b	0.83	0.29	0.40	0.81	0.32	0.42	0.53	0.20	0.24	0.50	0.19	0.23
Llama-3-70b	0.84	0.34	0.44	0.81	0.36	0.45	0.52	0.21	0.24	**0.51**	**0.21**	**0.24**
Mistral-7b	0.86	0.32	0.42	0.80	0.36	0.45	0.51	0.21	0.24	**0.49**	**0.21**	**0.24**
Mistral-8x7b	0.86	0.33	0.43	**0.81**	**0.36**	**0.46**	**0.52**	**0.22**	**0.25**	0.46	0.19	0.22
GritLM-7b	0.84	0.30	0.41	0.75	0.38	0.44	0.49	0.19	0.23	0.46	0.20	0.22
GritLM-8-7b	0.82	0.31	0.41	0.77	0.36	0.44	**0.54**	**0.21**	**0.25**	0.46	0.19	0.22
GPT-35-Turbo	0.81	0.32	0.42	0.80	0.35	0.44	0.50	0.21	0.24	**0.50**	**0.20**	**0.24**
GPT-4	**0.87**	**0.35**	**0.45**	0.86	0.30	0.42	**0.53**	**0.22**	**0.25**	0.62	0.16	0.23

Table 5: Comparison of performance of systems with different retrieval strategies on document and evidence retrieval.

	ERU	RUE	CUE	CCM
Llama-3-8b	0.46	0.68	0.33	0.31
Llama-3-70b	0.54	0.69	0.53	0.47
Mistral-7b	0.34	0.67	0.25	0.29
Mistral-8x7b	0.42	0.65	0.44	0.27
GritLM-7b	0.17	0.69	0.60	0.26
GritLM-8-7b	0.36	0.95	1.00	0.27
GPT-35-Turbo	0.54	0.70	0.52	0.45
GPT-4	0.59	0.71	0.59	0.52

Table 6: Comparison of systems with different LLMs on answer and citation quality evaluation. Abbreviations in the column headers are explained in Section 4.1.

LLMs for Answer Synthesis Next, we evaluate the performance of systems using different LLMs based on answer quality and citation quality metrics. As shown in Table 6, proprietary models like `GPT-3.5-Turbo` and `GPT-4` excel in effective retrieval usage and citation completeness, with `Llama-3-70b` also delivering competitive results. However, when it comes to the relevance and accuracy of attributed evidences in the answers, `GritLM-8x7b` clearly outperforms the others.

Overall Configurations Finally, in Table 9 of the Appendix, we present the aggregated performance of all systems sorted by averaging the results of 10 evaluation metrics detailed in Section 4.1. We see that the best-performing QA system configuration utilizes the DuckDuckGo search API for document retrieval and employs modality-specific indexing strategies. It leverages a `gte-large-en-v1.5` embedding model for retrieving text and table evidence, `CLIP` for retrieving image evidence, and integrates `GPT-4` for tasks requiring LLM capabilities. Additionally, the system incorporates multi-round retrieval with a memory of retrieval history and a self-reflection mechanism to utilize feedback for further enhancing retrieval performance.

4.3 Human Evaluation

We also conduct human evaluations on a subset of samples from tasks evaluated by GPT-4 to assess the alignment between human judgments and those of GPT-4. For the evidence relevancy task, we manually assess the relevance of all evidences in 50

Evaluation Task	Agreement
Evidence Relevancy	96.6%
Correct Usage of Evidences	75.7%
Citation Completeness	65.7%

Table 7: Agreements between judgements made by human and GPT-4 evaluators on retrieval, answer and citation evaluation tasks.

instances. For tasks assessing correct evidence usage and citation completeness, we randomly select 100 outputs from all system-generated responses for manual evaluation. As indicated in Table 7, there is a high level of agreement between human evaluators and the GPT-4 evaluator.

5 Related Work

5.1 Open-Domain QA

Open-domain question answering systems typically operate within a *Retriever-Reader* framework, where a retriever module identifies relevant documents, and a reader module employs a language model to extract the final answer from these documents (Hermann et al., 2015; Chen et al., 2017; Nguyen et al., 2017; Kwiatkowski et al., 2019; Lazaridou et al., 2023). Several studies (Nishida et al., 2018; Karpukhin et al., 2020; Khattab et al., 2021) have developed neural retrieval models that enhance the accuracy of document retrieval using neural networks. (Lee et al., 2018; Wang et al., 2018; Nogueira and Cho, 2019) focused on improving OpenQA systems by re-ranking documents before they are processed by the reader. Other research efforts include iterative document retrieval (Das et al., 2019; Feldman and El-Yaniv, 2019; Qi et al., 2019), and training end-to-end OpenQA systems (Lee et al., 2019; Lewis et al., 2020b; Sachan et al., 2024).

5.2 Multi-Modal QA

Multi-modal question answering requires retrieving and processing information from various modalities, often demanding cross-modal reasoning. Several benchmarks have been established to test these capabilities, including Chen et al. (2020); Talmor et al. (2021); Reddy et al. (2021); Chang et al. (2021); Singh et al. (2021); Li et al. (2022). Previous research has focused on different strategies for integrating these modalities. Some studies have developed methods for creating joint embeddings of different modalities (Hannan et al., 2020; Li et al., 2022; Chen et al., 2022; Yu et al., 2023). Yang et al. (2023) utilized entity-based fusion models to align content from disparate modalities. Additionally, Zhang et al. (2023) proposed using LLMs to extract and subsequently fuse information from multiple knowledge sources of different modalities.

6 Conclusion

In this study, we introduce OMG-QA, which challenges QA systems to retrieve and reason across text, tables, and images to generate long-form answers. Our experiments reveal that multi-index setups outperform single index setting in evidence retrieval, dense retrievers excel over sparse and generative retrievers, and multi-round retrieval enhances document recall but not necessarily evidence relevance in all cases. The choice of LLMs has minimal impact on retrieval performance; however, the best retrieval configuration paired with `GPT-4`, equipped with memory on retrieval history and self-reflection, showed superior results in overall evaluations. These findings emphasize the importance of integrating multi-modal content and sophisticated retrieval strategies in developing more capable QA systems, positioning OMG-QA as a robust benchmark for future advancements.

Limitation

Due to the open-ended nature of our questions, some might be solvable using information from a single modality, challenging the presumed necessity for a multi-modal approach.

There are inherent limitations when employing `GPT-4` to assess evidence relevance. Human annotators, without knowing the final answers, initially gather what they consider potentially relevant evidence for multi-hop questions. This initiates a dynamic process in which the evidence pool is continuously adjusted - irrelevant evidence is discarded, and pertinent evidence is enhanced as more information becomes available. Conversely, `GPT-4` evaluates evidence in isolation, without the capability to update its assessments based on new insights. This static approach can result in a greater tendency to overlook relevant evidence.

Acknowledgments

We are grateful for the compute support provided by the Microsoft Research's Accelerate Foundation Models Research (AFMR) program and Google's TRC program.

References

Chris Buckley and Ellen M. Voorhees. 2004. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, page 25–32, New York, NY, USA. Association for Computing Machinery.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2021. Webqa: Multihop and multimodal QA. *CoRR*, abs/2109.00590.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal qa. *Preprint*, arXiv:2109.00590.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. 2022. MuRAG: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5558–5570, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. *CoRR*, abs/1905.05733.

Yair Feldman and Ran El-Yaniv. 2019. Multi-hop paragraph retrieval for open-domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2309, Florence, Italy. Association for Computational Linguistics.

Bin Fu, Yunqi Qiu, Chengguang Tang, Yang Li, Haiyang Yu, and Jian Sun. 2020. A survey on complex question answering over knowledge base: Recent advances and challenges. *Preprint*, arXiv:2007.13069.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Darryl Hannan, Akshay Jain, and Mohit Bansal. 2020. Manymodalqa: Modality disambiguation and QA over diverse inputs. *CoRR*, abs/2001.08034.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for OpenQA with ColBERT. *Transactions of the Association for Computational Linguistics*, 9:929–944.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Jan Stokowiec, and Nikolai Grigorev. 2023. Internet-augmented language models through few-shot prompting for open-domain question answering.

Jinhyuk Lee, Seongjun Yun, Hyunjae Kim, Miyoung Ko, and Jaewoo Kang. 2018. Ranking paragraphs for improving answer recall in open-domain question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 565–569, Brussels, Belgium. Association for Computational Linguistics.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020a. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. 2019. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *Preprint*, arXiv:1908.06066.

Yongqi Li, Wenjie Li, and Liqiang Nie. 2022. MMCoQA: Conversational question answering over text, tables, and images. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4220–4231, Dublin, Ireland. Association for Computational Linguistics.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. Agentbench: Evaluating llms as agents. *Preprint*, arXiv:2308.03688.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. *ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks*. Curran Associates Inc., Red Hook, NY, USA.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. *Preprint*, arXiv:2304.09842.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, Online. Association for Computational Linguistics.

Luke Merrick, Danmei Xu, Gaurav Nuti, and Daniel Campos. 2024. Arctic-embed: Scalable, efficient, and accurate text embedding models. *Preprint*, arXiv:2405.05374.

Meta LLaMA Team. 2024. Introducing meta llama 3: The most capable openly available llm to date. `https://ai.meta.com/blog/meta-llama-3/`.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. *Preprint*, arXiv:2402.09906.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2017. MS MARCO: A human-generated MAchine reading COmprehension dataset.

Yixin Nie, Songhe Wang, and Mohit Bansal. 2019. Revealing the importance of semantic retrieval for machine reading at scale. 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 2553–2566, Hong Kong, China. Association for Computational Linguistics.

Kyosuke Nishida, Itsumi Saito, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2018. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 647–656, New York, NY, USA. Association for Computing Machinery.

Rodrigo Frassetto Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *CoRR*, abs/1901.04085.

Di Qi, Lin Su, Jianwei Song, Edward Cui, Taroon Bharti, and Arun Sacheti. 2020. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *ArXiv*, abs/2001.07966.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D. Manning. 2019. Answering complex open-domain questions through iterative query generation. 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 2590–2602, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Revanth Gangi Reddy, Xilin Rui, Manling Li, Xudong Lin, Haoyang Wen, Jaemin Cho, Lifu Huang, Mohit Bansal, Avirup Sil, Shih-Fu Chang, Alexander G. Schwing, and Heng Ji. 2021. Mumuqa: Multimedia multi-hop news question answering via cross-media knowledge extraction and grounding. *CoRR*, abs/2112.10728.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Shafiq Rayhan Joty Caiming Xiong Yingbo Zhou Semih Yavuz Rui Meng, Ye Liu. 2024. Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog.

Devendra Singh Sachan, Siva Reddy, William Hamilton, Chris Dyer, and Dani Yogatama. 2024. End-to-end training of multi-document reader and retriever for open-domain question answering. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441, Florence, Italy. Association for Computational Linguistics.

Hrituraj Singh, Anshul Nasery, Denil Mehta, Aishwarya Agarwal, Jatin Lamba, and Balaji Vasan Srinivasan. 2021. MIMOQA: Multimodal input multimodal output question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5317–5332, Online. Association for Computational Linguistics.

Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 895–909, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. Multimodal{qa}: complex question answering over text, tables and images. In *International Conference on Learning Representations*.

Ellen M. Voorhees. 2002. The philosophy of information retrieval evaluation. In *Evaluation of Cross-Language Information Retrieval Systems*, pages 355–370, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018. Evidence aggregation for answer re-ranking in open-domain question answering. In *International Conference on Learning Representations*.

Qian Yang, Qian Chen, Wen Wang, Baotian Hu, and Min Zhang. 2023. Enhancing multi-modal multi-hop question answering via structured knowledge and unified retrieval-generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 5223–5234, New York, NY, USA. Association for Computing Machinery.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Bowen Yu, Cheng Fu, Haiyang Yu, Fei Huang, and Yongbin Li. 2023. Unified language representation for question answering over text, tables, and images. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4756–4765, Toronto, Canada. Association for Computational Linguistics.

Le Zhang, Yihong Wu, Fengran Mo, Jian-Yun Nie, and Aishwarya Agrawal. 2023. MoqaGPT : Zero-shot multi-modal open-domain question answering with large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1195–1210, Singapore. Association for Computational Linguistics.

Yuyu Zhang, Ping Nie, Arun Ramamurthy, and Le Song. 2020. Ddrqa: Dynamic document reranking for open-domain multi-hop question answering. *ArXiv*, abs/2009.07465.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2023. Memorybank: Enhancing large language models with long-term memory. *Preprint*, arXiv:2305.10250.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *CoRR*, abs/2101.00774.

A Appendix

A.1 Evidence Retriever Configuration

We compare four types of evidence retrievers, each suited to specific indexing configurations:

- **Sparse Retriever**: Utilizes a BM25 retriever (Robertson and Zaragoza, 2009) to extract top-k textual evidences from a text-only index.

- **Dense Retriever**: Employs various text-embedding models to retrieve top-k evidences from both the text-only and corresponding text index in multi-index settings. Specifically, we used `SFR-Embedding-Mistral`[4] (Rui Meng, 2024), `gte-large-en-v1.5`[5] (Li et al., 2023), and `snowflake-arctic-embed-l`[6] (Merrick et al., 2024). For images, we use the `CLIP` model[7] (Radford et al., 2021), and for tables, we use the `gte-large-en-v1.5` model to build and retrieve from their respective indices.

- **Generative Retriever**: Extracts a table of contents from a document (excluding explicit mentions of tables and images) and prompts LLMs to predict relevant (sub)sections in order of potential relevancy, focusing on nodes closer to leaf nodes to minimize the volume of evidence retrieved. Top-k evidences are then selected based on predicted relevancy.

- **Hybrid Retriever**: Combines the generative and dense retrievers by using the generative approach to identify potentially relevant (sub)sections, followed by dense retrieval to rank and finalize the top-k evidences within the predicted sections.

Property	Value
Dataset Size	1,000
Question Length (Median/Avg)	37.4
No. Documents Relevant per Question	10.4
No. Evidences Relevant per Question	33.4
Percentage of questions that involve 3 modalities	40%
Percentage of questions that involve 2 modalities	60%
Modality Distribution of Evidences	Text 74.6% Table 13.2% Image 12.1%

Table 8: OMG-QA Statistics

[4] https://huggingface.co/Salesforce/SFR-Embedding-Mistral

[5] https://huggingface.co/Alibaba-NLP/gte-large-en-v1.5

[6] https://huggingface.co/Snowflake/snowflake-arctic-embed-l

[7] https://github.com/openai/CLIP/blob/main/model-card.md

```
Background [9]
        Procedure [10]
        Simultaneous elections [14]
Nominations [16]
        Democratic Party [20]
        Republican Party [24]
        Libertarian Party [30]
                [table] 2020 Libertarian Party ticket [32]
        Green Party [34]
                [table] 2020 Green Party ticket [36]
General election campaigns [38]
        Ballot access [39]
        Party conventions [41]
        Issues unique to the election [46]
                Impeachment [47]
                Effects of the COVID-19 pandemic [50]
                        [image] States and territories with at least one local, state, or
                            federal primary election date or method of voting altered
                            as of August 5, 2020. [51]
                        [image] A poll worker sanitizes an election booth in Davis,
                            California [53]
                Foreign interference [60]
                Trump's potential rejection of election results [68]
                Election delay suggestion [71]
                Postal voting [73]
                        [image] Chart of July 2020 opinion survey on likelihood of voting
                            by mail in November election, compared to 2016 [74]
                Federal Election Commission issues [79]
                Supreme Court vacancy [81]
                        [image] President Donald Trump with Amy Coney Barrett and her family,
                            just prior to Barrett being announced as the nominee,
                            September 26, 2020 [82]
                Pre-election litigation [85]
        Debates [87]
                [table] Debates for the 2020 U.S. presidential election sponsored by
                        the CPD [94]
        Polling [96]
                Two-way [97]
                        [table] Polling aggregates [99]
                        [table] Donald Trump vs. Joe Biden [101]
                Four-way [102]
                        [table] Donald Trump vs. Joe Biden vs. Jo Jorgensen vs.
                            Howie Hawkins [104]
                Swing states [105]
        Endorsements [109]
        Total cost estimate [110]
Campaign issues [112]
        COVID-19 pandemic [113]
        (Further text omitted for brevity)
```

Figure 2: Example of table of content representation of the Wikipedia page https://en.wikipedia.org/wiki/2020_United_States_presidential_election. Numbers represent the node ids that are used to locate contents in a document.

```
[
    {"system": "You are a question designer that develop questions by stages. You update your
    question based on the previous question and the new material given by the user at each stage.
    Now begin!"}
    {"user": "Generate a question that requires complex analyses and syntheses (ex. multihop)
    of information in the material. The question should be concrete enough to have only one
    single-fact objective answer. Questions asking 'impact', 'factors', 'reason' etc are considered
    too general and undesired. The answer to the question should be able to be determined from
    the following material:\n\n{material_1}"}
]
```

Figure 3: Prompt used for the initial question generation of Pipeline 1.

```
[
    {"system": "You are a question designer that develop questions by stages. You update your
    question based on the previous question and the new material given by the user at each stage.
    Now begin!"}
    {"user": "Generate a question that requires complex analyses and syntheses (ex. multihop)
    of information in the material. The question should be concrete enough to have only one
    single-fact objective answer. Questions asking 'impact', 'factors', 'reason' etc are considered
    too general and undesired. The answer to the question should be able to be determined from
    the following material:\n\n{material_1}"},
    {"assistant": "{initial_question}"},
    {"user": "Now I have one more material: \n\n{material_2}\n\nPlease update your
    question so that the new question:\n1. Uses both the information in the previous question
    and in the new material provided;\n
    2. The new question should also have only one objective correct answer, so avoid general
    questions about relation, impact, etc."}
]
```

Figure 4: Prompt used for the question revision of Pipeline 1.

```
Wikipedia page
{page_name}

Table of content:
{table_of_content}

Task: Given the above Wikipedia page table of contents and basic information of the tables and
images, generate a list of questions that require retrieving information from at least two different
modalities (e.g., text, table, image) to formulate an answer. For each question, also indicate which
section in the table of contents, which table and which image the question is referring to.
```

Figure 5: Prompt used for Pipeline 2 question generation.

```
Task: Determine if the provided evidence contains useful information to answer the given question.

Question:
{question}

Evidence:
From Document - {document_title}

'''
{evidence}
'''

Instructions: Review the question and evidence. If the evidence provides useful information for
answering the question, respond with a single letter "Y" for Yes. If it does not, respond with
a single letter "N" for No. Do not include any explanation or additional text in your response.

Your Answer:
```

Figure 6: Prompt used for the evidence relevancy evaluation task.

```
Your task is to evaluate whether the answer provided properly cites the specific evidence
excerpts given from different documents. Assess only the accuracy of the citations related
to the provided evidence excerpts, reflecting their content as presented in the original
documents. If there is inconsistent content between how the evidence is cited in the answer and
the content of the original evidence, this is an example of not properly using the evidence.
Ignore any additional evidence mentioned in the answer that is not among the provided excerpts.
Your response should be strictly limited to either 'Y' for Yes, if all provided evidences are
accurately cited, or 'N' for No, if any of the provided evidences are inaccurately cited.
Do not include any explanations or additional text—only the letter 'Y' or 'N' is required.

Question: {question}

Evidences:
{evidences}

Answer:
{answer}

Your Evaluation:
```

Figure 7: Prompt used for the correct usage of evidences (CUE) evaluation task.

```
Your task is to evaluate the citation completeness of the provided answer. Determine whether
all evidences that are relevant to any content in the answer are cited. Assess if every piece
of information in the answer that requires support from documents has a corresponding, properly
cited evidence mentioned. Your response should strictly be 'Y' for Yes if every relevant piece
of evidence is cited in the answer, or 'N' for No if any relevant evidence is missing or not cited.
Do not include any explanations or additional text—only the letter 'Y' or 'N' is required
as a response.

Question: {question}

Evidences:
{evidences}

Answer:
{answer}

Your Evaluation:
```

Figure 8: Prompt used for the citation completeness (CCM) evaluation task.

Document Retriever	Indexer	Evidence Retriever	LLM	Retrieval Strategy	DR P/R/F1	ER P/R/F1	ERU/RUE	CUE/CCM	Avg
D	MS	te, tae-gte ie-clip	GPT-4	MR	0.859/0.3/0.417	0.616/0.158/0.226	0.656/0.759	0.681/0.472	0.514
D	MS	te, tae-gte ie-clip	GritLM-8x7b	SR	0.824/0.311/0.408	0.539/0.214/0.25	0.365/0.95	1.0/0.27	0.513
D	MS	te, tae-gte ie-clip	GPT-4	SR	0.869/0.349/0.453	0.526/0.219/0.251	0.591/0.707	0.588/0.52	0.507
D	MS	te, tae-gte ie-clip	GPT-35-turbo	MR	0.8/0.35/0.439	0.502/0.201/0.237	0.506/0.699	0.556/0.54	0.483
D	MS	te, tae-gte ie-clip	Llama-3-70b	SR	0.843/0.337/0.437	0.521/0.211/0.245	0.545/0.686	0.53/0.47	0.482
D	MS	te, tae-gte ie-clip	Mistral-8x7b	MR	0.814/0.357/0.455	0.46/0.186/0.219	0.53/0.681	0.588/0.44	0.473
D	MS	te, tae-gte ie-clip	GPT-35-turbo	SR	0.815/0.323/0.417	0.5/0.212/0.239	0.543/0.697	0.521/0.45	0.472
D	MS	te, tae-gte ie-clip	Llama-3-70b	MR	0.81/0.358/0.451	0.513/0.207/0.243	0.527/0.699	0.434/0.42	0.466
D	MS	te, tae-gte ie-clip	GritLM-8x7b	MR	0.768/0.357/0.444	0.459/0.191/0.219	0.396/0.628	0.833/0.28	0.457
D	TO	h-gte	Mistral-7b	SR	0.811/0.338/0.425	0.524/0.196/0.238	0.352/0.667	0.536/0.32	0.441
D	TO	h-sfr	Mistral-7b	SR	0.787/0.353/0.433	0.522/0.208/0.249	0.347/0.717	0.361/0.42	0.44
D	MS	te, tae-gte ie-clip	Mistral-8x7b	SR	0.861/0.326/0.43	0.517/0.216/0.247	0.424/0.648	0.438/0.27	0.438
D	TO	s	GPT-4	SR	0.714/0.363/0.425	0.375/0.117/0.148	0.712/0.587	0.382/0.48	0.43
D	TO	h-gte	GritLM-7b	SR	0.793/0.305/0.396	0.459/0.182/0.212	0.465/0.786	0.5/0.18	0.428
D	TO	h-sfr	Llama-3-8b	SR	0.79/0.324/0.42	0.449/0.191/0.216	0.53/0.635	0.375/0.344	0.428
D	MS	te, tae-gte ie-clip	Llama-3-8b	SR	0.83/0.295/0.397	0.526/0.2/0.241	0.455/0.677	0.333/0.312	0.427
D	TO	te-sfr	Llama-3-8b	SR	0.809/0.32/0.421	0.463/0.202/0.229	0.534/0.624	0.175/0.49	0.427
D	TO	te-sfr	Mistral-7b	SR	0.817/0.345/0.434	0.44/0.207/0.226	0.359/0.604	0.344/0.49	0.427
D	MS	te, tae-gte ie-clip	Llama-3-8b	MR	0.814/0.324/0.42	0.505/0.194/0.23	0.463/0.653	0.333/0.295	0.423
D	TO	s	GPT-35-turbo	SR	0.685/0.334/0.4	0.381/0.126/0.153	0.724/0.643	0.385/0.39	0.422
D	TO	te-gte	GritLM-7b	SR	0.817/0.316/0.415	0.492/0.189/0.227	0.488/0.667	0.375/0.23	0.422
D	TO	te-gte	Mistral-7b	SR	0.842/0.335/0.433	0.524/0.213/0.246	0.299/0.592	0.3/0.43	0.421
D	TO	te-gte	Llama-3-8b	SR	0.827/0.298/0.403	0.527/0.2/0.241	0.431/0.715	0.312/0.24	0.419
D	TI	te-gte ie-clip	GritLM-7b	SR	0.826/0.328/0.429	0.479/0.191/0.228	0.489/0.75	0.2/0.27	0.419
D	MS	te, tae-gte ie-clip	GritLM-7b	SR	0.839/0.305/0.411	0.493/0.191/0.229	0.171/0.686	0.6/0.26	0.419
D	TI	te-gte ie-clip	Llama-3-8b	SR	0.815/0.304/0.407	0.522/0.197/0.238	0.434/0.639	0.286/0.292	0.413
D	MS	te, tae-gte ie-clip	Mistral-7b	SR	0.855/0.318/0.42	0.513/0.207/0.24	0.335/0.673	0.25/0.29	0.41
D	MS	te, tae-gte ie-clip	Mistral-7b	MR	0.8/0.355/0.445	0.488/0.207/0.236	0.281/0.621	0.294/0.33	0.406
D	TI	te-gte ie-clip	Mistral-7b	SR	0.837/0.336/0.434	0.523/0.212/0.245	0.324/0.521	0.333/0.29	0.406
D	TO	te-sfr	GritLM-7b	SR	0.78/0.323/0.413	0.393/0.178/0.2	0.544/0.8	0.167/0.22	0.402
D	MS	te, tae-gte ie-clip	GritLM-7b	MR	0.754/0.376/0.444	0.456/0.195/0.221	0.417/0.917	0.0/0.22	0.4
D	TO	h-gte	Llama-3-8b	SR	0.801/0.31/0.406	0.502/0.201/0.232	0.41/0.679	0.205/0.25	0.4
D	TO	s	Llama-3-70b	SR	0.701/0.348/0.411	0.369/0.106/0.139	0.71/0.601	0.27/0.33	0.399
D	TO	s	Mistral-8x7b	SR	0.707/0.334/0.405	0.339/0.105/0.135	0.644/0.624	0.422/0.21	0.393
D	TO	g	Mistral-7b	SR	0.875/0.253/0.345	0.368/0.128/0.158	0.455/0.651	0.417/0.27	0.392
D	TO	g	GritLM-7b	SR	0.809/0.224/0.309	0.317/0.111/0.133	0.553/0.738	0.5/0.14	0.383
W	TO	s	Llama-3-70b	SR	0.634/0.275/0.341	0.365/0.115/0.143	0.711/0.587	0.328/0.271	0.377
W	TO	s	Mistral-8x7b	SR	0.705/0.285/0.37	0.376/0.129/0.156	0.601/0.616	0.263/0.194	0.369
D	TO	s	GritLM-7b	SR	0.657/0.315/0.384	0.308/0.098/0.12	0.548/0.575	0.4/0.15	0.355
W	TO	s	Mistral-7b	SR	0.605/0.28/0.339	0.335/0.101/0.127	0.515/0.557	0.424/0.245	0.353
D	TO	s	Mistral-7b	SR	0.71/0.34/0.411	0.353/0.101/0.135	0.427/0.553	0.25/0.24	0.352
D	TO	s	Llama-3-8b	SR	0.709/0.319/0.399	0.377/0.098/0.139	0.608/0.496	0.167/0.188	0.35
D	TO	s	GritLM-8x7b	SR	0.674/0.309/0.375	0.36/0.117/0.143	0.304/0.539	0.5/0.16	0.348
D	TO	g	Llama-3-8b	SR	0.871/0.202/0.299	0.394/0.123/0.157	0.446/0.507	0.269/0.198	0.347
D	TO	h-sfr	GritLM-7b	SR	0.77/0.308/0.397	0.39/0.181/0.199	0.305/0.429	0.25/0.2	0.343
W	TO	s	Llama-3-8b	SR	0.62/0.265/0.325	0.329/0.105/0.135	0.674/0.523	0.125/0.226	0.333
W	TO	s	GritLM-7b	SR	0.554/0.249/0.306	0.323/0.084/0.118	0.5/0.5	0.5/0.082	0.322
W	TO	s	GritLM-8x7b	SR	0.606/0.267/0.327	0.332/0.107/0.133	0.444/0.494	0.25/0.152	0.311
D	TO	h-arctic	Mistral-7b	SR	0.757/0.342/0.421	0.393/0.154/0.179	0/0	0/0.15	0.24
D	TO	te-arctic	Mistral-7b	SR	0.759/0.342/0.42	0.345/0.115/0.147	0/0	0/0.19	0.232
D	TO	te-arctic	Llama-3-8b	SR	0.769/0.32/0.409	0.355/0.119/0.152	0/0	0/0.188	0.231
D	TO	te-arctic	GritLM-7b	SR	0.743/0.322/0.403	0.349/0.125/0.153	0/0	0/0.07	0.216
D	TO	h-arctic	Llama-3-8b	SR	0.754/0.312/0.396	0.342/0.125/0.152	0/0	0/0.062	0.214
D	TO	h-arctic	GritLM-7b	SR	0.757/0.303/0.388	0.308/0.117/0.136	0/0	0/0.1	0.211

Table 9: System Ranking by Average Evaluation Results. D - DuckDuckGo Search API, W - Wikipedia Search API, TO - text-only indexer, TI - text-image indexer, MS - modality-specific indexer, s - sparse retriever, g - generative retriever, h - hybrid retriever, te - text embedding, ie - image embedding, tae - table embedding, SR - single-round, MR - multi-rounds