Benchmarking Retrieval-Supported Large Language Models for Open-Domain Question Answering

Gökay Gülsoy[1]

İzmir Institute of Technology, Gülbahçe Campus 35430 Urla Izmir Türkiye
gokaygulsoy@iyte.edu.tr
https://ceng.iyte.edu.tr/

Abstract. Advances in large language models (LLMs) initiated a new era for information retrieval (IR) systems, in which queries and answers based on contextual information that were once generated by humans can now also be generated by LLMs with domain knowledge. Existing information retrieval techniques combined with the significant comprehension capabilities of LLMs also made a substantial improvement in the field of question answering systems. In this study, different information retrieval techniques are utilized, such as lexical token-based retrieval, semantic similarity-based retrieval, hypothetical answer generation-based retrieval, pseudo-relevance feedback-based retrieval, and reciprocal RAG fusion-based retrieval, to compare their effectiveness on LLM-based question answering. Three different commercial LLM models from different providers are used in experiments using a subset of the MS MACRO question answering development dataset, which are GPT-3.5 Turbo, Claude 3 Haiku, and Gemini 1.5 Flash. In total, 15 configurations were obtained, where each model is tested with the 5 information retrieval strategies and BLEU, ROUGE-1, ROUGE-L, Bert Scores (precision, recall, F1-score), along with LLM judge-based scores are used in the evaluation of configurations. The experimental results demonstrate that retrieval choice exerts a substantial and model-dependent impact on QA performance, with fusion-based and semantic retrieval strategies consistently outperforming purely lexical and feedback-driven methods, while GPT-3.5-Turbo achieves the strongest overall performance across the majority of retrieval configurations.

Keywords: Information Retrieval · Retrieval Augmented Generation · NLP · Large Language Models · Question Answering Systems

1 Introduction

Information retrieval techniques are widely used for acquiring the most relevant information from a huge pool of data. Many different information retrieval strategies were designed and proposed [29] in the literature for increasing retrieval capabilities. With the latest advances in LLMs, question-answering systems can

be designed to provide promising results where extensive domain knowledge of LLMs is combined with fine-grained information extracted from a context with relevant information retrieval techniques. This study uses 5 different information retrieval approaches, namely, lexical token-based retrieval, semantic similarity-based retrieval, hypothetical answer generation-based retrieval, pseudo-relevance feedback-based retrieval, and reciprocal RAG fusion-based retrieval, in combination with the complex generational capabilities of LLMs to provide the most relevant answer given a query. The MS MARCO question answering development dataset [2] is used for evaluating the performance of retrieval techniques and LLM combinations with BLEU, ROUGE-1, ROUGE-L, and BERT Score (precision, recall, F1-score) metrics. Furthermore, as a more advanced evaluation strategy, GPT-4o-based LLM-as-judge [22] is used to evaluate generated answers in terms of correctness, faithfulness, and context quality. 3 different LLM models from different providers are used, which are GPT-3.5-Turbo, Gemini-2.5-Flash, and Claude-3-Haiku-20240307, along with 5 aforementioned retrieval techniques per LLM, totaling 15 different retrieval configurations. The project is provided in GitHub[1] as a repository for further research, experimentation, and extension.

2 Related Works

A lot of work has been done on the design and development of various information retrieval techniques in literature, ranging from traditional approaches, which are divided into three main categories by [9], namely, set theoretic models, algebraic models, and probabilistic models, to modern deep neural network and large language model-based schemes [13]. The traditional text retrieval systems mainly rely on the matching of terms between the query and documents [7]. Even though it is fast, simple, and easy to implement, term-based retrieval systems have several limitations, such as polysemy, synonymy, and lexical gaps between the query and the documents [1, 47]. In spite of these limitations, it is widely used in practical applications, specifically in small-scale information retrieval systems [27]. Other methods, such as BIM proposed by [37] examines statistical techniques for exploiting relevance information to weight search terms, and TDM proposed by [43] explores one way of removing the independence assumption among index terms.

Instead, the extent of the dependence between index terms is measured and used to construct a non-linear weighting function. [6] presented a method for improving the effectiveness of pseudo-relevance feedback (PRF) in information retrieval systems via re-examining the assumption of PRF that most frequent terms in pseudo-feedback documents are useful for the retrieval. It shows that good expansion terms cannot be distinguished from bad ones merely on their distributions in the feedback documents and proposes to integrate term classification process to predict the usefulness of expansion terms. The field of infor-

[1] https://github.com/GokayGulsoy/Benchmarking-Retrieval-Supported-Large-Language-Models-for-Open-Domain-Question-Answering/tree/main

mation retrieval has also seen a significant amount of research focused on the use of lexical dependency models for document retrieval.

The study presented in [41] provides a thorough analysis of various term weighting schemes and their impact on retrieval performance, and proposes a novel term dependency weighting scheme that accounts for the relationship between terms in a document. The use of probabilistic models for information retrieval has also been explored in the literature. [18] presents a new probabilistic model that is based on the vector space model (VSM) and incorporates term dependency weighting. Recently, [10] presents a new approach to information retrieval based on a language model (LM) that incorporates term dependency by proposing a new Dependence Language Model (DLM) that utilizes term dependencies to improve retrieval effectiveness. Neural-based approaches for semantic retrieval, such as those utilizing CNNs and RNNs have been proposed as a means to improve the effectiveness of information retrieval systems.

These approaches involve representing queries and text documents in a continuous vector space, allowing for the utilization of neural network-based similarity measures, such as cosine similarity or dot product, to rank the documents according to their relevance to a given query [14, 42]. Additionally, attention-based mechanisms such as the Transformer architecture [44] have been used to improve the ability of IR systems by attending important parts of the query and documents for matching. Additionally, the use of pre-trained language models, such as GPT-2 [35] and BERT [8] have been shown to improve the performance of IR systems by providing a better understanding of semantics and context of natural language queries and documents. [16] shows that attention scores from a sequence-to-sequence reader model are a good measure of document relevance and, inspired by knowledge distillation, proposed a method to iteratively train the retriever from these activations and compare different loss functions on three different question-answering benchmarks.

Since the labeled data can be scarce and because neural retrieval models require vast amounts of data to train, [24] proposed a novel method for generating synthetic training data for retrieval. [28] proposes DTR, which uses large dual encoders for generalizable retrieval. Recently, the most popular application of IR in support of large language models (LLMs) is Retrieval Augmented Generation (RAG) [21]. The underlying principle of RAG is that the implicit evidence grounded in a ranking is provided to the LLMs to generate a natural language answer in response to a user's prompt or query. LLMs are known for generating or *hallucinating* false information from time to time [17]. RAG-based systems provide more accurate and factual answers than LLMs without retrieval-based guidance. Currently, it is more cost-efficient to keep an index updated rather than retraining an LLM [23]. State-of-the-art RAGE-based systems rely on vector databases where each information item is represented by an embedding vector [4]. The retrieved ranked list of items is usually implemented with nearest neighbor search methods. In this context, compromises have to be made between Exact Nearest Neighbors (ENN) and Approximate Nearest Neighbors (ANN) search methods [33].

3 Methodology

In this study, 5 different approaches of information retrieval, namely, lexical token-based retrieval, semantic similarity-based retrieval, hypothetical answer generation-based retrieval, pseudo-relevance feedback-based retrieval, and reciprocal RAG fusion-based retrieval, are used in combination with 3 different LLM models for answer generation from different providers, which are gpt-3.5-Turbo, gemini-2.5-Flash, and claude-3-haiku-20240307. Experimental results for each configuration were evaluated with BLEU [32], ROUGE- scores (specifically ROUGE-1 and ROUGE-L) [25], BertScores (precision, recall, F1-score), and GPT-4o-based LLM judge scores using the subset of a MS MARCO question-answering development dataset version V1.1.

3.1 Dataset

MS MARCO question-answering dataset [3] is used in this study. The MS MARCO dataset has certain advantages over other MR (Machine Reading Comprehension) datasets, which are summarized as follows:

- **Real questions:** All questions have been sampled from real anonymized Bing queries
- **Real Documents:** Most URLs that the passage has been sourced contain full web documents. These can be used as extra contextual information to improve systems.
- **Human Generated Answers:** All questions have an answer written by humans. If there was no answer in the passages the judge read, they have written *'No Answer Present'*

specifically, a subset of the development set of version V1.1 is taken, which contains 400 queries. A subset of the development set is extracted using *ms_marco_qna_dataset_parser.py* Python script and converted to *CSV* file format. The dataset has the following format:

```
{ "answers": [list of answers]
  "passages":[
     {
       "is_selected": 0 or 1,
       "url": url of the taken passage
       "passage_text": passage text itself
     }],
   "query": query text,
   "query_id": unique id of query,
   "query_type": type of query
   }
```

The original QA dataset has 1,010,916 queries, but for feasibility, in this study, 400 queries are taken as a subset. Some of the queries in the dataset do not have

a selected correct answer; only the queries that have a selected correct answer were selected to be in the extracted subset. Descriptions of the fields are as follows:

1. **query_id:** A unique id for each query that is used in the evaluation.
2. **query:** A unique query based on Bing usage.
3. **passages:** A set of 10 passages, URLs, and an annotation if they were used to formulate and answer(is_selected:1). Two passages have been selected as the most relevant passages. If a passage is marked as is_selected:1, it means the judge used that passage to formulate their answer. If a passage is marked as is_selected:0, it means the judge did not use that to generate their answer. Questions that have the answer of 'No Answer Present.' will have all passages marked as is_selected:0.
4. **query type:** A basic division of queries based on a trained classifier. Categories are: {LOCATION, NUMERIC, PERSON, DESCRIPTION, ENTITY} can be used to make smaller more focused datasets.
5. **answers:** An array of answers produced by human judges, most contain a single answer, $\sim 1\%$ contain more than one answer(average of ~ 2 answers if there are multiple answers). These answers were generated by real people in their own words instead of selecting a span of text.

3.2 Used Retrieval Techniques

5 different approaches of information retrieval configurations are used with each LLM model, which are lexical token-based retrieval, semantic similarity-based retrieval, hypothetical answer generation-based retrieval, pseudo-relevance feedback-based retrieval, and reciprocal RAG fusion-based retrieval. Each of the following subsections explains how they work.

Lexical Token-Based Retrieval Lexical token-based retrieval used in this study relies on the BM25 Okapi [39]. It works by tokenizing the whole passage corpus and creates a token pool. Then, tokenizes the query and finds the most relevant passages based on the matching token count. BM25 weighting funtion is defined as in in the Equation 1 given below:

$$\sum_{T \in Q} w^{(1)} \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k_3+qtf} + k_2.Q.\frac{avdl-dl}{avdl+dl} \tag{1}$$

where

- Q is a query, containing terms T
- $w^(1)$ is the Robertson Sparck Jones weight [37]

$$log\frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} \tag{2}$$

- N is the number of items (documents) in the collection

- n is the number of documents containing the term
- R is the number of documents known to be relevant to a specific topic
- r is the number of relevant documents containing the term
- K is $k_1((1-b)+b.dl/avdl)$
- k_1, b, k_2 and k_3 are parameters which depend on the database and possibly on the nature of the topics. For the TREC-4 experiments [38], typical k_1, k_3 and k_3 and b were 1.0, 2.0, and 0.6, 0.75, and k_2 was zero
- tf is the frequency of the term within a specific document
- qtf is the frequency of the term within the topic from w which Q was derived
- dl is the document length (arbitrary units)
- $avdl$ is the average document length

Semantic Similarity-Based Retrieval In semantic search, Sentence Transformers are used to compute the embeddings for queries and paragraphs. Given a search query, the first retrieval system retrieves a large list of possible hits that are potentially relevant for the query using a dense retriever. However, the retrieval system might retrieve documents that are not that relevant to the search query. Hence, in a second stage, a re-ranker is used based on some Cross Encoder [36] model that scores the relevance of all candidates for the given search query.
The output of the Cross Encoder model is a ranked list of hits that can be used in further downstream tasks. In this study "all-MiniLM-L6-v2" model [45] is used as a Sentence Transformer for generating query and passage embeddings. Then query embedding is compared with passage embeddings using cosine similarity to retrieve the most relevant 3 passages by fetching the 3 highest scoring passages. Re-ranker is not used in this study as our pool of retrieval is fairly small, containing up to 10 passages for each query, which, in that case, makes it plausible to rely on retrieval without using re-ranker.

Hypothetical Answer Generation-Based Retrieval Given a query, HyDE (Hypothetical Document Embeddings) [11] first zero-shot instructs an instruction-following language model to generate a *hypothetical* answer. The answer captures the relevance patterns but is unreal and may contain false information. Then, a supervised contrastively learned encoder (e.g., Contriever) encodes the document into an embedding vector. This vector identifies a neighborhood in the corpus embedding space, where similar *real* passages are retrieved based on vector similarity.
In this study, 'all-MiniLM-L6-v2' is used as a sentence encoder to generate an embedding vector for hypothetical answer and passages. Then, this hypothetical embedding is used to retrieve the most relevant 3 passages, and the final answer is generated using the original query and the retrieved 3 passages. This procedure is depicted in Figure 1.
Dense retrieval models calculate the similarity between a query and a document with inner product similarity. Given a query q and document d, it uses two

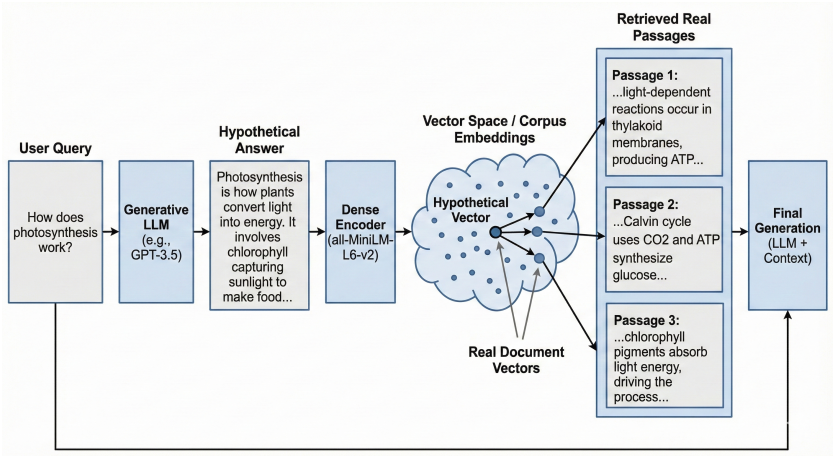

Fig. 1. HyDE-based Information Retrieval and Answer Generation

encoder functions enc_q and enc_d to map them into d dimensional vectors $\mathbf{v_q}$, $\mathbf{v_d}$, whose inner product is used as a similarity measurement.

$$sim(q, d) = \langle enc_q(q), enc_d(d) \rangle = \langle \mathbf{v_q}, \mathbf{v_d} \rangle \tag{3}$$

For zero-shot retrieval, we consider L query sets $Q_1, Q_2, \cdots, Q_L$ and their corresponding search corpus, document sets $D_1, D_2, \cdots, D_L$. Denote the $j-th$ query from the $i-th$ query set Q_i as q_{ij}. We need to fully define mapping functions enc_q and enc_d without access to any query set Q_i, document set D_i, or any relevance judgement r_{ij}.

The difficulty of zero-shot dense retrieval lies precisely in Equation 3, which requires learning of two embedding functions (for query and document, respectively) into the *same* embedding space where inner product captures *relevance*. Without relevance judgements/scores to fit, learning becomes intractable. HyDE circumvents the aforementioned learning problem by performing search in document only embedding space that captures document-document similarity. This can be easily learned using contrastive learning [15]. We set document encoder enc_d directly as a contrastive encoder enc_{con}.

$$f = enc_d = enc_{con} \tag{4}$$

This function is also denoted as f for simplicity. This unsupervised contrastive encoder will be shared by all incoming document corpora.

$$\mathbf{v_d} = f(d) \quad \forall \in D_1 \cup D_2 \cup \cdots \cup D_L \tag{5}$$

To build the query vector, we consider in addition an instruction following LLM, InstructLM. It takes a query q and a textual instruction INST and follows them to perform the task indicated by INST. For simplicity it is denoted as:

$$g(q, \text{INST}) = \text{InstructLM}(q, \text{INST}) \tag{6}$$

Now queries can be mapped to "hypothetical" documents by sampling from g, setting INST to the instruction to be followed. The generated document is not read, can, and likely to be ungrounded factually [5]. It is only required to capture the relevance pattern. Critically, here, relevance modelling is offloaded from the representation learning model to a Natural Language Generation model that generalizes substantially more easily and effectively. We can encode the generated answer using the document encoder f as follows:

$$\mathbf{v}_{ij} = f(g(q_{ij}, \text{INST})) \tag{7}$$

Pseudo-Relevance Feedback-Based Retrieval In the field of information retrieval, one of the key challenges is delivering relevant and precise information that effectively addresses user queries. Pseudo-relevance feedback (PRF) is a technique that is widely used to improve retrieval effectiveness by refining queries based on initially retrieved documents [46]. Traditional PRF methods such as RM3 [26] and Rocchio [40] are applied with bag-of-words approach models like BM25, a lexical sparse retrieval method.

Even though these approaches help re-ranking results using term frequency, there are limitations because of their reliance on exact word matches, often leading to issues like polysemy and synonymy being ignored. With the rise of large language models (LLMs) such as GPT-3 and BERT, more sophisticated methods of integrating PRF with semantic comprehension have emerged. Those models enable retrieval systems to go beyond surface-level keyword matches, capturing deeper semantic relationships between queries and documents. Many studies applied language models to PRF methods for advancing retrieval effectiveness via identifying semantic relevance between pseudo-relevant documents and the original query [30, 31]. Dense retrieval models [20] and [19], built on dual-encoder architectures, encode queries and documents independently. In spite of the improved performance of these language models, dense retrieval models can miss out the rich interaction between query and document content, mainly due to a separate encoding process.

Incorporating external knowledge, generated by LLM, is a powerful technique to augment the retrieval process. The generative capabilities of large language models, like ChatGPT, Gemini, etc., offer an enhanced semantic understanding. The approach not only expands the original query using information from the top K pseudo-relevant documents but also re-encodes the query with external knowledge from generative language models, enhancing the semantic understanding between the original query and pseudo-relevant documents. By generating external knowledge related to the query using a language model and incorporating information from pseudo-relevant documents, the representation of the original query is further enhanced. Giving the user query as an input to the language model produces semantically rich terms that address the areas not covered by the

original query but are relevant to the user's information needs. This procedure is described as given in the Figure 2.

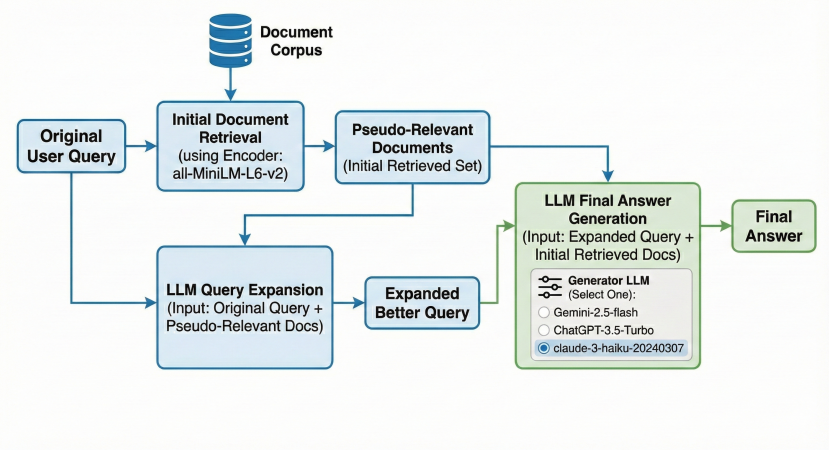

Fig. 2. Pseudo-Relevance Feedback-based Information Retrieval and Answer Generation

Specifically, the external knowledge representation of the query obtained through the language model is denoted by the equation 8 as follows, where Q_0 represents the original query:

$$f = LLM[Q_o] \tag{8}$$

This paper uses the strategy to modify the representation of the original query by utilizing information from pseudo-relevant documents and external knowledge from the original query, as given in the equation 9 as follows:

$$q = (q, [\text{SEP}], d_1, [\text{SEP}], d_2, [\text{SEP}], \cdots, [\text{SEP}], d_k) \tag{9}$$

Reciprocal RAG Fusion-Based Retrieval Traditional RAG systems rank documents in order of relevance to the query, usually by vector distances. This means that the more relevant a document is in a query, its priority will be higher, being in the answer. Recently, however, researchers have explored implementing different re-ranking in retrieval augmented generation, which plays a substantial role in improving retrieval results and in increasing relevance, accuracy, and comprehensiveness of answers [12]. Once the original query is received, the model sends the original query to the large language model to generate a varying number of search queries, which were selected as 4 in this study, based on the original query. For example, if the user's original query is "Tell me about the NVIDIA GPUs", the generated queries may include:

– *What are NVIDIA GPUs and how they work ?*
– *What are the advantages of using NVIDIA GPUs ?*
– *What are some of the recommended NVIDIA GPUs ?*

The RAG-fusion algorithm [34] then performs a vector search to find a number of relevant documents, as the usual RAG. But instead of sending those documents with queries to the large language model to generate the output, the model performs reciprocal rank fusion. Reciprocal rank fusion is an algorithm used to assign scores to every document and re-rank them according to the scores. The scores assigned to each document (rrf scores) are computed by Equation 10 as follows:

$$rrfscore = \frac{1}{rank + k} \tag{10}$$

Where rank is the position of a document among the retrieved passages sorted by similarity score, and k is a constant smoothing factor which is used as a weight given to the existing ranks. After each calculation of the score, the rrf score is accumulated with the previous scores for the same document, and when all scores are accumulated, documents are fused together and reranked according to their scores. Then the model sends the reranked results along with the original query to a large language model to produce an answer. Overall procedure for RAG-Fusion is illustrated in the Figure 3 as below.

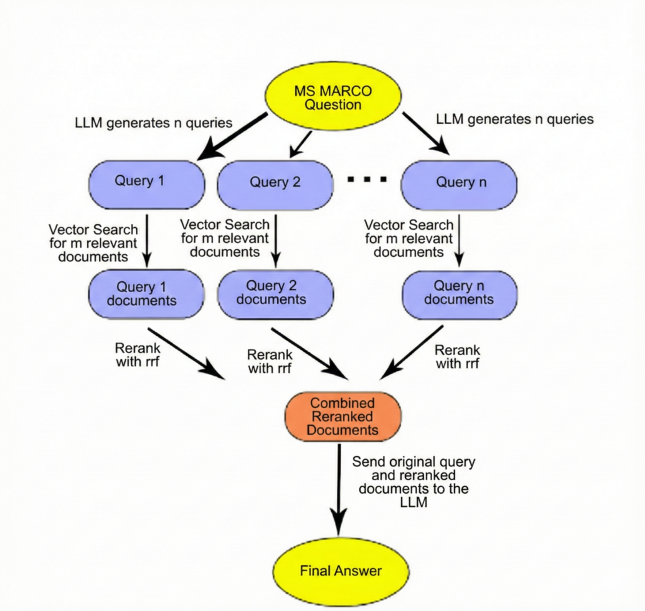

Fig. 3. RAG-Fusion-Based Information Retrieval and Answer Generation

4 Evaluation and Experimental Results

In total, 15 configurations along with 5 retrieval approaches for 3 different LLM models, namely, GPT-3.5 Turbo, Gemini-2.5-Flash, and Claude-3-Haiku-20240307 are used in experiments. BLEU, ROUGE-1, ROUGE-L, and BERTScore with roberta-large backbone (precision, recall, F1-score) were used as evaluation metrics. Additionally, LLM-as-a-judge evaluation using GPT-4o was employed. The LLM judge was prompted to rate correctness score, faithfulness score, and context quality score on a 1-5 scale. Descriptions of each LLM judge's score and the prompt of the judge LLM are as follows:

- **correctness score:** Score 1-5: Does the generated answer convey the same meaning as the ground truth answer
- **faithfulness score:** Score 1-5: Is the generated answer supported by the Retrieved Context?
- **context quality score:** Score 1-5: Does the Retrieved Context contain the same key information as the ground Truth Passages?
- **explanation:** Brief justification for the scores

Judge LLM's Prompt

"""
You are an expert evaluator for the Retrieval-Augmented Generation (RAG) system.
Your goal is to grade the system's performance based on multiple perspectives.

–INPUT DATA–
1. User Query: query
2. Ground Truth Answer (Human Written):
ground_truth_answer
3. Ground Truth Context (Passages used by Human):
ground_truth_context
4. Generated Answer (AI written):
generated_answer
5. Retrieved Context (Passages found by AI):
retrieved_context

–EVALUATION CRITERIA–
1. Correctness (1-5): Compare the Generated Answer to the Ground Truth Answer AND Ground Truth Context. Does the AI answer the question correctly? (Allow for phrasing differences).
2. Faithfulness (1-5): Is the Generated Answer derived *only* from the Retrieved Context? (Penalize hallucinations not found in #5).
3. Compare #5 (Retrieved Context) vs #3 (Ground Truth Context). Did the AI retrieve the necessary information to answer the question?
"""

Experimental results for GPT-3.5 Turbo, Gemini-2.5-Flash, and Claude-3-Haiku-20240307 with 5 different retrieval techniques are given in Tables 1-2, 3-4, and 5-6 respectively.

Table 1. BLEU, ROUGE, and BERT Score Metrics for GPT-3.5-Turbo

Retrieval Technique	BLEU	ROUGE-1	ROUGE-L	BERT Scores		
				Precision	Recall	F1-score
lexical token-based retrieval	0.081	**0.311**	**0.277**	**0.862**	**0.878**	**0.869**
semantic similarity-based retrieval	0.085	**0.321**	**0.283**	**0.863**	**0.881**	**0.871**
HyDE-based retrieval	0.081	**0.318**	**0.283**	**0.863**	**0.882**	**0.872**
pseudo-relevance feedback-based retrieval	0.034	0.236	0.201	**0.849**	0.871	0.859
reciprocal RAG fusion-based retrieval	0.083	**0.332**	**0.296**	**0.873**	**0.883**	**0.873**

Table 2. LLM Judge Scores for GPT-3.5-Turbo

Retrieval Technique	Correctness Scores	Faithfulness Score	Context Quality Score
lexical token-based retrieval	3.675	4.757	**4.137**
semantic similarity-based retrieval	3.860	4.832	**4.462**
HyDE-based retrieval	3.850	**4.755**	4.377
pseudo-relevance feedback-based retrieval	3.390	4.502	**4.392**
reciprocal RAG fusion-based retrieval	3.770	**4.832**	**4.457**

Table 3. BLEU, ROUGE, and BERT Score Metrics for Gemini-2.5-Flash

Retrieval Technique	BLEU	ROUGE-1	ROUGE-L	BERT Scores		
				Precision	Recall	F1-score
lexical token-based retrieval	**0.091**	0.290	0.255	0.853	**0.878**	0.865
semantic similarity-based retrieval	**0.104**	0.311	0.278	0.855	**0.881**	0.867
HyDE-based retrieval	**0.100**	0.312	0.276	0.854	0.881	0.867
pseudo-relevance feedback-based retrieval	**0.088**	**0.284**	**0.248**	0.848	**0.878**	**0.862**
reciprocal RAG fusion-based retrieval	**0.103**	0.314	0.280	0.854	**0.883**	0.868

Table 4. LLM Judge Scores for Gemini-2.5-Flash

Retrieval Technique	Correctness Score	Faithfulness Score	Context Quality Score
lexical token-based retrieval	**3.695**	**4.955**	4.110
semantic similarity-based retrieval	**3.905**	**4.937**	4.450
HyDE-based retrieval	3.772	4.587	4.297
pseudo-relevance feedback-based retrieval	3.667	**4.535**	4.347
reciprocal RAG fusion-based retrieval	3.767	4.535	4.335

Table 5. BLEU, ROUGE, and BERT Score Metrics for Claude-3-Haiku-20240307

Retrieval Technique	BLEU	ROUGE-1	ROUGE-L	BERT Scores		
				Precision	Recall	F1-score
lexical token-based retrieval	0.062	0.251	0.213	0.839	0.876	0.856
semantic similarity-based retrieval	0.069	0.268	0.232	0.841	**0.881**	0.860
HyDE-based retrieval	0.069	0.267	0.229	0.841	0.881	0.860
pseudo-relevance feedback-based retrieval	0.057	0.249	0.212	0.837	0.877	0.856
reciprocal RAG fusion-based retrieval	0.073	0.270	0.235	0.843	0.881	0.861

Table 6. LLM Judge Scores for Claude-3-Haiku-20240307

Retrieval Technique	Correctness Score	Faithfulness Score	Context Quality Score
lexical token-based retrieval	3.572	4.297	3.995
semantic similarity-based retrieval	3.852	4.527	4.377
HyDE-based retrieval	**3.862**	4.492	**4.382**
pseudo-relevance feedback-based retrieval	**3.672**	4.402	4.252
reciprocal RAG fusion-based retrieval	**3.825**	4.507	4.355

According to experimental results, the highest score model achieved within a specific technique compared to other models is indicated in bold in the above tables. In terms of BLEU, ROUGE, and BERT scores out of 30 results, GPT-3.5-Turbo achieved the highest scores in 21, Gemini-2.5-Flash achieved the highest scores in 12, and Claude-3-Haiku-20240307 achieved the highest scores in only 1 result. For the LLM-as-judge scores out of 15 results, GPT-3.5-Turbo achieved the highest scores in 6, Gemini-2.5-Flash achieved the highest scores in 5, and Claude-3-Haiku-20240307 achieved the highest scores in 4 results.

When scores are compared internally for each model, in the case of GPT-3.5-Turbo, semantic similarity-based retrieval achieved the highest score for BLEU,

reciprocal RAG fusion-based retrieval achieved the highest score for ROUGE-1, ROUGE-L, and BERT Scores (precision, recall, F1-score). For Gemini-2.5-Flash, semantic similarity-based retrieval achieved the highest score for BLEU and BERT precision, reciprocal RAG fusion-based retrieval achieved the highest score for ROUGE-1, ROUGE-L, BERT Recall, and BERT F1-score. For Claude-3-Haiku-20240307, reciprocal RAG fusion-based retrieval achieved the highest score for BLEU, ROUGE-1, ROUGE-L, BERT Precision, and BERT F1-score; semantic similarity-based retrieval also achieved the highest score for BERT Recall.
For the judge LLM scores, GPT-3.5-Turbo achieved the highest correctness score with pseudo-relevance feedback-based retrieval, the highest faithfulness score achieved with both semantic similarity-based retrieval and reciprocal RAG fusion-based retrieval, and the highest context quality score achieved with semantic similarity-based retrieval. Gemini-2.5-Flash achieved the highest for correctness, faithfulness, and context quality score with semantic similarity-based retrieval. Claude-3-Haiku-20240307 achieved the highest for correctness, faithfulness, and context quality score with HyDE-based retrieval. Overall, the number of experimental configurations in which each LLM model achieved the highest scores out of 45 experimental configurations is as follows:

1. **Chat-3.5-Turbo** $\rightarrow$ 27
2. **Gemini-2-5-Flash:** $\rightarrow$ 17
3. **Claude-3-Haiku-20240307:** $\rightarrow$ 5

It was observed that GPT-3.5-Turbo demonstrated the best performance, followed by Gemini-2.5-Flash, followed by Claude-3-Haiku-20240307 in the MS MARCO question-answering development subset. Also, different retrieval choices lead to the highest scores for different LLM model which signifies the importance of the retrieval technique to be selected for RAG integration specific to the LLM model.

5 Conclusion

In this research, 5 different information retrieval techniques are used and compared, which are lexical token-based, semantic similarity-based, hypothetical answer generation-based, pseudo-relevance feedback-based, and reciprocal RAG fusion-based retrieval. Comparison of the effectiveness of these techniques on the LLM-based question answering problem was carried out using a subset of the MS MARCO question-answering dataset. Three different commercial LLM models from different providers are used in experiments, which are GPT-3.5-Turbo, Gemini 2.5 Flash, and Claude 3 Haiku. Experimental results showed that GPT-3.5-Turbo achieved the highest scores for most of the retrieval configurations, which was 27, followed by Gemini-2.5-Flash model with 17, and finally Claude-3-Haiku-20240307 with 5 out of 45 configurations. Moreover, results also demonstrated that answer-generation quality is substantially affected by the used retrieval technique, in which the retrieval technique that leads to the best score can be different according to the type of LLM model.

References

1. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern information retrieval, vol. 463. ACM press New York (1999)
2. Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., Wang, T.: Ms marco: A human generated machine reading comprehension dataset (2018), arXiv preprint arXiv:1611.09268, https://arxiv.org/abs/1611.09268
3. Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., et al.: Ms marco: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268 (2016)
4. Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Van Den Driessche, G.B., Lespiau, J.B., Damoc, B., Clark, A., et al.: Improving language models by retrieving from trillions of tokens. In: International conference on machine learning. pp. 2206–2240. PMLR (2022)
5. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
6. Cao, G., Nie, J.Y., Gao, J., Robertson, S.: Selecting good expansion terms for pseudo-relevance feedback. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 243–250. SIGIR '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1390334.1390377, https://doi.org/10.1145/1390334.1390377
7. Croft, W.B., Metzler, D., Strohman, T., et al.: Search engines: Information retrieval in practice, vol. 520. Addison-Wesley Reading (2010)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). pp. 4171–4186 (2019)
9. Dong, H., Hussain, F.K., Chang, E.: A survey in traditional information retrieval models. In: 2008 2nd IEEE International Conference on Digital Ecosystems and Technologies. pp. 397–402 (2008). https://doi.org/10.1109/DEST.2008.4635214
10. Gao, J., Nie, J.Y., Wu, G., Cao, G.: Dependence language model for information retrieval. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 170–177 (2004)
11. Gao, L., Ma, X., Lin, J., Callan, J.: Precise zero-shot dense retrieval without relevance labels. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1762–1777 (2023)
12. Glass, M., Rossiello, G., Chowdhury, M.F.M., Naik, A., Cai, P., Gliozzo, A.: Re2g: Retrieve, rerank, generate. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 2701–2715 (2022)
13. Hambarde, K.A., Proenca, H.: Information retrieval: recent advances and beyond. IEEE Access **11**, 76581–76604 (2023)
14. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 2333–2338 (2013)

15. Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., Grave, E.: Unsupervised dense information retrieval with contrastive learning (2022), https://arxiv.org/abs/2112.09118
16. Izacard, G., Grave, E.: Distilling knowledge from reader to retriever for question answering. arXiv preprint arXiv:2012.04584 (2020)
17. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P.: Survey of hallucination in natural language generation. ACM computing surveys **55**(12), 1–38 (2023)
18. Jones, K.S., Walker, S., Robertson, S.E.: A probabilistic model of information retrieval: development and comparative experiments: Part 2. Information processing & management **36**(6), 809–840 (2000)
19. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6769–6781. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.550, https://aclanthology.org/2020.emnlp-main.550/
20. Khattab, O., Zaharia, M.: Colbert: Efficient and effective passage search via contextualized late interaction over bert. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 39–48. SIGIR '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3397271.3401075, https://doi.org/10.1145/3397271.3401075
21. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems **33**, 9459–9474 (2020)
22. Li, D., Jiang, B., Huang, L., Beigi, A., Zhao, C., Tan, Z., Bhattacharjee, A., Jiang, Y., Chen, C., Wu, T., et al.: From generation to judgment: Opportunities and challenges of llm-as-a-judge. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 2757–2791 (2025)
23. Li, S., Stenzel, L., Eickhoff, C., Bahrainian, S.A.: Enhancing retrieval-augmented generation: a study of best practices. arXiv preprint arXiv:2501.07391 (2025)
24. Liang, D., Xu, P., Shakeri, S., dos Santos, C.N., Nallapati, R., Huang, Z., Xiang, B.: Embedding-based zero-shot retrieval through query generation (2020), https://arxiv.org/abs/2009.10270
25. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81 (2004)
26. Lv, Y., Zhai, C.: A comparative study of methods for estimating query language models with pseudo feedback. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. p. 1895–1898. CIKM '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1645953.1646259, https://doi.org/10.1145/1645953.1646259
27. Manning, C.D.: Introduction to information retrieval. Syngress Publishing, (2008)
28. Ni, J., Qu, C., Lu, J., Dai, Z., Hernandez Abrego, G., Ma, J., Zhao, V., Luan, Y., Hall, K., Chang, M.W., Yang, Y.: Large dual encoders are generalizable retrievers. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing.

pp. 9844–9855. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022). https://doi.org/10.18653/v1/2022.emnlp-main.669, https://aclanthology.org/2022.emnlp-main.669/
29. Nyamisa, M.F., Mwangi, W., Cheruiyot, W.: A survey of information retrieval techniques. Advances in Networks **5**(2), 40–46 (2017). https://doi.org/10.11648/j.net.20170502.12, https://doi.org/10.11648/j.net.20170502.12
30. Pan, M., Wang, J., Huang, J.X., Huang, A.J., Chen, Q., Chen, J.: A probabilistic framework for integrating sentence-level semantics via bert into pseudo-relevance feedback. Inf. Process. Manage. **59**(1) (Jan 2022). https://doi.org/10.1016/j.ipm.2021.102734, https://doi.org/10.1016/j.ipm.2021.102734
31. Pan, M., Zhou, S., Chen, J., Huang, E.A., Huang, J.X.: A semantic framework for enhancing pseudo-relevance feedback with soft negative sampling and contrastive learning. Inf. Process. Manage. **62**(3) (May 2025). https://doi.org/10.1016/j.ipm.2024.104058, https://doi.org/10.1016/j.ipm.2024.104058
32. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
33. Quinn, D., Nouri, M., Patel, N., Salihu, J., Salemi, A., Lee, S., Zamani, H., Alian, M.: Accelerating retrieval-augmented generation. In: Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1. pp. 15–32 (2025)
34. Rackauckas, Z.: Rag-fusion: a new take on retrieval-augmented generation. arXiv preprint arXiv:2402.03367 (2024)
35. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
36. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
37. Robertson, S.E., Jones, K.S.: Relevance weighting of search terms. Journal of the American Society for Information science **27**(3), 129–146 (1976)
38. Robertson, S.E., Walker, S., Beaulieu, M., Gatford, M., Payne, A.: Okapi at trec-4. Nist Special Publication Sp pp. 73–96 (1996)
39. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at trec-3. In: Text Retrieval Conference (1994), https://api.semanticscholar.org/CorpusID:41563977
40. Rocchio, J.J.: Relevance feedback in information retrieval. In: Salton, G. (ed.) The Smart retrieval system - experiments in automatic document processing, pp. 313–323. Englewood Cliffs, NJ: Prentice-Hall (1971)
41. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information processing & management **24**(5), 513–523 (1988)
42. Shen, Y., He, X., Gao, J., Deng, L.: Latent semantic models with deep neural networks for information retrieval. In: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. ACM. pp. 269–278 (2014)
43. VAN RIJSBERGEN, C.: A theoretical basis for the use of co-occurrence data in information retrieval. Journal of Documentation **33**(2), 106–119 (02 1977). https://doi.org/10.1108/eb026637, https://doi.org/10.1108/eb026637

44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
45. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers (2020), https://arxiv.org/abs/2002.10957
46. Wang, Z., Pei, Q.: Dense retrieval systems with llm-based query expansion. In: 2024 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT). pp. 682–686. IEEE (2024)
47. Ziviani, N., Gonçalves, M.A., de Moura, E.S., Ribeiro-Neto, B., da Silva, A.S., Veloso, A.: Information retrieval research at ufmg. Journal of Information and Data Management **2**(2), 77–77 (2011)