# Comparative Analysis of Context Retrieval Methods in In-context RALM

**Jiaxuan Li**
jackljx@student.ubc.ca

**Haotian Gong**
ht2012@student.ubc.ca

**Yuxiang Fu**
strive2p@student.ubc.ca

## Abstract

A refined, context-oriented answer highly depends on the quality of the context presented to the language models. This paper aims to investigate the effectiveness of in-context retrieval augmented language models (RALMs) in answering questions by comparing and evaluating different context retrieval methods. To achieve this objective, an end-to-end two-stage question answering application was implemented. The study compared three context retrieval methods: Term Frequency - Inverse Document Frequency (TF-IDF), Dense Passage Retrieval (DPR), and Hypothetical Document Embeddings (HyDE) under the designed benchmarks. The performance of the context retrieval methods was evaluated, and the quality of the final answer obtained was compared. Among the three levels of complexity for questions (*remember*, *understand*, and *apply*), TF-IDF was found to have the highest context scores and achieved the highest document interpretation scores in precision, *relevance*, and *coherence* for the *understand* category. HyDE outperformed both DPR and TF-IDF for questions at the highest level of complexity (*apply*).

## 1 Introduction

The frontier of machine learning has been greatly advanced by large language models (LLMs). We have seen great popularity and success of recent applications such as ChatGPT, which exhibits human-level performance on various benchmarks. However, LLMs do not have built-in source attribution mechanisms, and are known to "hallucinate" answers, producing counterfactual or unreliable output (Maynez et al. 2020).

Transfer learning approaches such as fine-tuning enables the model to be more context-aligned (Ouyang et al. 2022), but requires expensive re-training every time. An alternative approach is to design the prompt to include specific context while querying the language model. Recent studies (Ram et al. 2023) show a 2-stage model, split between context retrieval and document interpretation, yields substantial gains. The context retrieval phase collects pieces of text most relevant to the user question, and the document interpretation phase involves sending that text to pre-trained LLMs along with the query.

In our work, we hope to investigate the In-context RALMs in depth. We will:

**Implement** an end-to-end two-stage question answering application that yields factually grounded answers.

**Design benchmarks** to evaluate and compare the performance of the context retrieval methods, and the resulting quality of the final answer obtained from GPT.

**Compare** the performance of three context retrieval methods: Term Frequency - Inverse Document Frequency (TF-IDF), Dense Passage Retrieval (DPR), and Hypothetical Document Embeddings (HyDE) under our benchmark.

Course project for UBC CPSC 440/540, April 2023.

**Determine** the significance of context retrieval methods for In-context RALM, providing empirical evidence on which methods work well, and whether using a better context retieval methods leads to a higher quality answer.

# 2 Background

## 2.1 Context Retrieval Methods

For our study, we have selected the following context retrieval methods for comparison:

**TF-IDF**: TF-IDF is a measure used in information retrieval and machine learning to quantify the importance of words, phrases, and other string representations in a document relative to a corpus. TF (term frequency) refers to the frequency of a particular term in a document, while IDF (inverse document frequency) looks at how uncommon a word is in the entire corpus.

TF-IDF vectorization involves calculating the product between TF and the logarithm of IDF for every word in a corpus and using that information to create a feature vector for each document, which can then be used for various purposes, such as measuring document resemblance using cosine similarity. *Understanding TF-IDF for Machine Learning* 2021

**Dense Passage Retrieval**: A Dense Passage Retriever (DPR) retrieves relevant passages for a given question by comparing the low-dimensional representations of the passages and questions. To ensure fast processing, an index of these representations is pre-computed and maintained.

Specifically, a DPR encodes a large number of passages in a low-dimensional, continuous space using embeddings learned from a limited set of questions and passages through a dual encoder framework $E_P$ and $E_Q$ (Karpukhin et al. 2020). The encoder $E_P$ maps any text passage to a high dimensional real-valued vector and establishes an index for the entire resource. Similarly, the encoder $E_Q$ maps the query to the same codomain. During inference, the system retrieves the most relevant passages using large-scale minimum inner product search (Johnson, Douze, and Jégou 2017), i.e.

$$\mathrm{sim}(p, q) = E_Q(q) \cdot E_P(p).$$

**Hypothetical Document Embeddings (HyDE)**:

To capture relevance patterns, HyDE directly inputs a given question to a generative language model, which creates a hypothetical document with potential factual inaccuracies. This hypothetical document is then encoded into an embedding vector using an unsupervised contrastive encoder $f = E_{con}$. $g$ is a generative LM that maps queries to "hypothetical" documents. By sampling from $g$ and setting a specific instruction $\mathrm{INST}_i$ with a designated query, we have the equation $\mathbb{E}[v_{q_{ij}}] = \mathbb{E}[f(g(q_{ij}, \mathrm{INST}_i))]$. We then take the inner-product between $v_{q_{ij}}$ and the set of all document vectors. The most similar documents are retrieved eventually (Gao et al. 2022).

## 2.2 In-Context RALM

The primary objective of language models (LMs) is to define probability distributions over sequences of tokens. In order to model the probability of a given sequence $x_1, \ldots, x_n$, the standard approach is to use next-token prediction,

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid x_1, \ldots, x_{i-1})$$

where the sequence of tokens preceding $x_i$ is denoted as the prefix. This autoregressive model is customarily implemented through a learned transformer network, optimizing a set of parameters $\theta$. The causal self-attention mask(Radford et al. 2018) that underpins structure of these conditional probabilities is based on the sequence of tokens preceding $x_i$. The next-token prediction with this mask is an effective and straightforward paramtrization paradigm followed by the current LMs(Brown et al. 2020)(Zhang et al. 2022)(Reed et al. 2021).

RALMs are a type of language model that incorporates an operation for retrieving one or more documents from an external corpus $\mathcal{C}$ and conditioning the model's predictions on these documents.

In addition, In-context RALM is a type of RALM that belongs to the retrieve and read model family, which includes separate context retrieval and document reading components (Ram et al. 2023). It coalesces the retrieved grounding corpus within the Transformer's input prior to the prefix, without altering the LM weights $\theta$.

### 2.3 Benchmarks for QA Models

The existing benchmark suites for question answering are insufficient for us to conduct a fair evaluation of our current model. Based on the results of a recent survey paper (Wang 2022), out of the 41 proposed benchmarks, over 1/3 are true/false or multiple-choice based. Another 1/3 are variants of named entity recognition (Marrero et al. 2013), which only requires the model to return simple words or phrases to a given "what", "when", and "where" question. With large language models now having the power to surpass the performance of the average student on university entrance exams (Bubeck et al. 2023), the surveyed benchmarks are mostly outdated and inadequate for evaluating the fine-grained performance of such models.

Additionally, the evaluation metric for QA also requires careful design. The most popular metrics primarily measure the overlap degree to which a predicted answer meets a target answer as an indicator of accuracy. The higher the overlap, the more accurate we think the model is. However, natural language is complex, and there may exist multiple correct ways to express the same answer, especially when we focus on "why" or "how" questions. Currently, we do not have access to the model's parameters to evaluate metrics such as perplexity, which is a probability measure of how "sure" the model is about its answer (Ranjan et al. 2016).

## 3 Framework

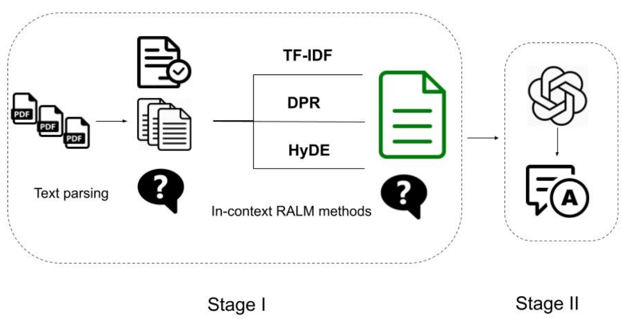



Figure 1: A schematic diagram of our framework. The corpora are preprocessed initially. We manually label the most relevant segments in the text according to the given query. We compare the context retrieved from three In-context RALM methods with the segments in Stage I. In stage II, we stack the context retrieved from three methods and the query and arrange the prompts. We pass the prompt to GPT and achieve the final answer.

Given a collection of documents, we first split each document into basic retrieval units $p_1, p_2, ...p_n$ of the same length that form the corpus $\mathcal{C}$.

In Stage I, our goal is to find a span $p_s, p_{s+1}...p_e$ that can answer the user question $q$ using a retriever. A retriever is defined as a filter function $\mathcal{R} : (q, \mathcal{C}) \to \mathcal{C}_q$, where $\mathcal{C}_q \subset \mathcal{C}$, and $|\mathcal{C}_q| \ll |\mathcal{C}|$. We use 3 comparable context retrieval methods (i.e. $\mathcal{R}^{\text{TF-IDF}}_{\mathcal{C}}(\cdot), \mathcal{R}^{\text{DPR}}_{\mathcal{C}}(\cdot), \mathcal{R}^{\text{HyDE}}_{\mathcal{C}}(\cdot)$).

In stage II, we combine the retrieved evidence from stage I and pass it to a generative LM without altering the LM's parameters. In particular, this operation can be expressed as

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p_\theta(x_i \mid [\mathcal{R}_{\mathcal{C}}(x_1, \ldots, x_{i-1}); x_1; \ldots; x_{i-1}])$$

where $\mathcal{R}_{\mathcal{C}}(\cdot)$ denotes the retrieval operation and $[s; t]$ represents the concatenation of strings $s$ and $t$ (Ram et al. 2023).

A detailed schematic diagram is provided in Figure 1.

# 4 Benchmark

Our benchmark consists of 4 major parts: a corpus of facts which we want to base our inference on; a set of questions we hope to ask; a metric for evaluating the performance of context retrieval methods; and a metric for evaluating the quality of the final natural language answer that GPT-3.5 yields.

## 4.1 Corpus Selection

We selected 3 collections of texts from different sources: a recently published Master's thesis in bioinformatics by a UBC graduate student; a textbook used for first year economics courses; and official CPSC 340 course slides from last term.

These materials vary in presentation style, and require different levels of domain-specific knowledge to interpret.

## 4.2 Question Design

Inspired by the work done by professor Knorr (2020) from UBC, we carefully design the questions according to the framework of Bloom's taxonomy. Bloom's Taxonomy (Bloom 1974) categorizes educational objectives for the cognitive domains. There are six levels of the cognitive processes in the taxonomy which is arranged in a hierarchical order of increasing difficulty. We focus on the first three:

1. **Remembering**: recalling or recognizing exact words or paragraphs from memory. This corresponds to the "what", "when", and "where" questions common in traditional QA benchmarks.
2. **Understanding**: comprehending the semantics of information. This corresponds to "why" and "how" questions that are often absent in previous benchmarks.
3. **Applying**: utilizing information in a new situation or context. This requires an understanding of the question in context of the given corpera, in addition to answering "how" or "why".

Due the excess manual effort of setting ground-truth labels associated with the introduction of each new question, we only developed a suite of 27 questions within the project time frame. We hope that our small yet carefully annotated test set can produce preliminary findings that serve as a basis for future in-depth analysis.

## 4.3 Metric for Context Retrieval

Given the corpus $\mathcal{C}$ and a user query $q$, context retrievers produces $C_q \subset \mathcal{C}$ containing sections the retriever believes are the most relevant for answering $q$. In this method, we compare $C_q$ with set $T_q$ which contains most relevant sections for answering $q$ by human labelling. We simulate the labelling process that previous benchmarks such as MS-Marco takes (Nguyen et al. 2016). Since we did not deploy crowd-sourcing methods like MS-Marco, labelling has proved to be extremely time consuming for our 3-person team.

We define a metric as $m = \dfrac{|T_q \cap S_q|}{|T_q|}$ for comparing method performance. In this case $m = 1$ implies all ground truth sections are selected while $m = 0$ implies none are selected.

## 4.4 Metric for Document Interpretation

We designed the following 3 criteria for determining the quality of an answer given by the LLM:

- **Precision**: Is the answer precise and specific? Does it avoid broad generalizations or vague statements?
- **Relevance**: Is the answer relevant to the question being asked? Does it directly address the question or is it tangential?
- **Coherence**: Are the ideas in the answer presented in a logical and coherent way?

These measures are constructed based on modifications to the TOEFL (Test of English as a Foreign Language) writing rubric. The three scores are largely independent of each other, and aim to objectively reflect the quality of a answer. We avoid assessing the output simply based on “accuracy", as the notion of “accuracy" is challenging to define for open or half-open questions.

We manually grade the questions on a 0-5 scale following a modified TOEFL rubric. We recognize that this process is highly subjective, and could introduce bias into our results. Thus, we required that every member of our 3-person team assigns a grade to all the answers simultaneously, and for each answer we take the average grade. Ideally, this cross-validating procedure will mitigate the bias we incept during grading to the greatest extent.

# 5 Experiment

In this section, we describe the experimental setup and results from both the context retrieval and quantitative measures upon the final answer.

## 5.1 Implementation Details

We connected the open source pre-trained implementation of DPR(Karpukhin et al. 2020) to our model. We also connected the original HyDE implementation(Gao et al. 2022), which includes GPT-3.5 as its hypothetical document generator.

To extract text from PDF files and perform initial data cleaning, we employed the `PyMuPDF` package. For question answering, we utilized the `LangChain` package and OpenAI’s GPT-3.5.

We engineered the prompt we send to GPT-3.5 specifically to circumvent “hallucinations" or using facts that were not provided. The model is instructed to limit its answer within the prefixed context and output “I don’t know" when it receives insufficient context to answer the query. Experimental results show that this instruction is indeed effective.

The source code including the benchmark suite can be found here:
*https://github.com/JackLiJXL/cpsc440-project*.

## 5.2 Results

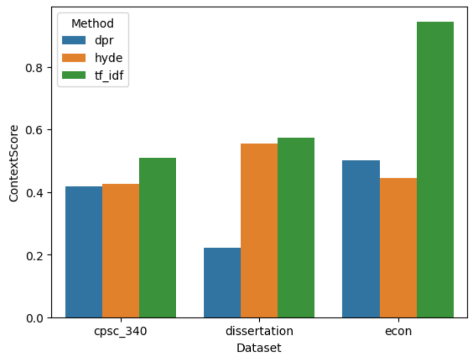
Table 1 summarizes our model performance, evaluated based on the previously proposed benchmarks. From the table, we observe that among all three levels (*remember*, *understand*, and *apply*), TF-IDF obtained the highest context scores. However, this does not directly lead to a higher document interpretation score – except for the *understand* level, TF-IDF is outperformed by HyDE, despite HyDE having a lower context score.

Next, we examined the context retrieval performance specific to the datasets we used in Figure 2a. We notice that DPR with the master’s dissertation yields very poor results while TF-IDF seems to do extremely well with the economics textbook.

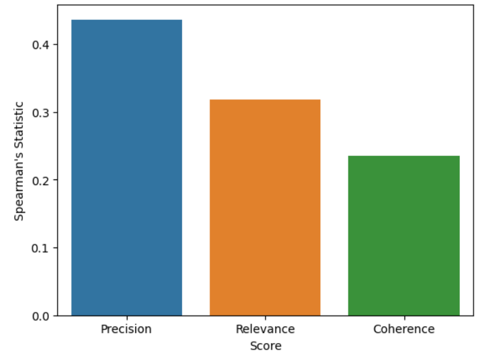
Finally, we analyzed the correlation between context score and the *precision, relevance, coherence* scores using Spearman’s rank correlation coefficient(“Spearman Rank Correlation Coefficient” 2008) in Figure 2b. We selected this statistical test because it is appropriate for both continuous and discrete ordinal variables. We discover that the answer quality scores are in general moderately to weakly correlated with the context score.

# 6 Discussion

We made various attempts to understand why TF-IDF performs very well with respect to the context retrieval score. We think that it is due to the nature of academic corpora, where key phrases are effective indicators of the topic of any given text segment, especially in the case of textbooks. While naively extracting large amounts of loosely related segments makes the context score high, it may also include segments that are similar but provides no meaningful information with respect to the question, confounding the LM.




(a) Performance of context retrieval methods, grouped by dataset.




(b) Correlation between context score and answer quality, measured by Spearman's statistic. Between the precision score and the context score, the Spearman rank coefficient has a p-value of 4e-5 with a sample size 81. While the Spearman rank coefficient has a p-value of 0.004 and 0.034 corresponding to variable pairs the (relevence score, context score) and (coherence score, context score) respectively.

Figure 2

Table 1: Aggregated results of running and evaluating our two-stage model on custom benchmarks.

| Method | Bloom Taxonomy | ContextScore | PrecisionScore | RelevanceScore | CoherenceScore |
|---|---|---|---|---|---|
| DPR | Remember | 0.52 | 3.89 | 4.89 | 4.78 |
| | Understand | 0.36 | 3.11 | 3.89 | 3.78 |
| | Apply | 0.26 | 4.00 | 3.89 | 4.11 |
| HyDE | Remeber | 0.74 | 4.33 | 4.78 | 4.89 |
| | Understand | 0.37 | 3.22 | 3.89 | 4.00 |
| | Apply | 0.31 | 4.00 | 4.44 | 4.11 |
| TF-IDF | Remember | 0.89 | 4.33 | 3.89 | 4.11 |
| | Understand | 0.73 | 4.78 | 4.78 | 4.89 |
| | Apply | 0.41 | 3.11 | 3.67 | 3.11 |

We also identified some limitations of more advanced methods, including poor performance of DPR with dissertation questions. This highlights the need for fine-tuning of DPR to better handle specialized texts, such as dissertations.

HyDE outperformed both DPR and TF-IDF for questions at the highest level of complexity (*apply*). We hypothesize that this is due to several factors. Firstly, the question prompt for such complex questions is often semantically different from the viable answers and relevant passages. Secondly, the InstructGPT LLM utilized by HyDE has access to more general knowledge, making it more capable of producing embeddings that are semantically similar to viable answers and relevant passages from the question prompt, as compared to the supervised encoder used by DPR.

The correlation between context retrieval effectiveness and document interpretation quality is not as high as we hypothesized it would be, especially in the *relevance* metric. We realize that this may be an issue with its design. We should make our *relevance* criteria stricter as to measure whether the answer touches upon the “top $k$" most relevant facts. The high correlation between *precision* relative to the 2 other metrics is also unexpected. We hypothesize that this is because a high-quality context enables the language model to capture a short sequence in the embedding space that has the same probability as a longer sequence. In future work, we should explore the connection between perplexity of the model and its output length in fuller extent.

This study has several limitations, the main issue being a small sample size of questions and a limited focus on academic datasets. To address these limitations, further comprehensive studies are needed that consider a wider range of documents as context and questions. Additionally, the black box nature of the system restricted the use of certain metrics such as perplexity, emphasizing the need for more interpretable models.

## 7 Conclusion

Our study provides a novel framework and benchmark suite for the comparison context-retrieval methods in a end-to-end question answering setting. Initial results show that the relationship between context-retrieval and question answering quality is complex and requires greatly refined metrics to capture. We observe certain limitations for advanced methods such as DPR that should be investigated. We call for more comprehensive studies with complete benchmarks and the development of more interpretable models to better understand NLP systems.

## References


Bloom, Benjamin Samuel (1974). *Taxonomy of educational objectives: The classification of educational goals*. en.

Brown, Tom B. et al. (2020). *Language Models are Few-Shot Learners*. DOI: 10.48550/ARXIV.2005.14165. URL: https://arxiv.org/abs/2005.14165.

Bubeck, Sébastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang (2023). *Sparks of Artificial General Intelligence: Early experiments with GPT-4*. arXiv: 2303.12712 [cs.CL].

Gao, Luyu, Xueguang Ma, Jimmy Lin, and Jamie Callan (2022). *Precise Zero-Shot Dense Retrieval without Relevance Labels*. arXiv: 2212.10496 [cs.IR].

Johnson, Jeff, Matthijs Douze, and Hervé Jégou (2017). *Billion-Scale Similarity Search with GPUs*. arXiv: 1702.08734 [cs.CV].

Karpukhin, Vladimir, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih (2020). *Dense Passage Retrieval for Open-Domain Question Answering*. arXiv: 2004.04906 [cs.CL].

Knorr, Edwin M. (Feb. 2020). "Worked Examples, Cognitive Load, and Exam Assessments in a Senior Database Course." *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM. DOI: 10.1145/3328778.3366915. URL: https://doi.org/10.1145/3328778.3366915.

Marrero, Mónica, Julián Urbano, Sonia Sánchez-Cuadrado, Jorge Morato, and Juan Miguel Gómez-Berbís (2013). "Named Entity Recognition: Fallacies, challenges and opportunities." *Computer Standards & Interfaces* 35.5, pp. 482–489. ISSN: 0920-5489. DOI: https://doi.org/10.1016/j.csi.2012.09.004. URL: https://www.sciencedirect.com/science/article/pii/S0920548912001080.

Maynez, Joshua, Shashi Narayan, Bernd Bohnet, and Ryan McDonald (July 2020). "On Faithfulness and Factuality in Abstractive Summarization." *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 1906–1919. DOI: 10.18653/v1/2020.acl-main.173. URL: https://aclanthology.org/2020.acl-main.173.

Nguyen, Tri, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng (2016). "MS MARCO: A human generated machine reading comprehension dataset." *choice* 2640, p. 660.

Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe (2022). *Training language models to follow instructions with human feedback*. arXiv: 2203.02155 [cs.CL].

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. (2018). "Improving language understanding by generative pre-training."

Ram, Ori, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham (2023). *In-Context Retrieval-Augmented Language Models*. arXiv: 2302.00083 [cs.CL].

Ranjan, Nihar, Kaushal Mundada, Kunal Phaltane, and Saim Ahmad (Jan. 2016). “A Survey on Techniques in NLP.” *International Journal of Computer Applications* 134, pp. 6–9. DOI: 10.5120/ijca2016907355.

Reed, Lena, Cecilia Li, Angela Ramirez, Liren Wu, and Marilyn Walker (2021). “Jurassic is (almost) All You Need: Few-Shot Meaning-to-Text Generation for Open-Domain Dialogue.” DOI: 10.48550/ARXIV.2110.08094. URL: https://arxiv.org/abs/2110.08094.

“Spearman Rank Correlation Coefficient” (2008). *The Concise Encyclopedia of Statistics*. New York, NY: Springer New York, pp. 502–505. ISBN: 978-0-387-32833-1. DOI: 10.1007/978-0-387-32833-1_379. URL: https://doi.org/10.1007/978-0-387-32833-1_379.

*Understanding TF-IDF for Machine Learning* (Oct. 2021). URL: https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/.

Wang, Zhen (2022). *Modern Question Answering Datasets and Benchmarks: A Survey*. arXiv: 2206.15030 [cs.CL].

Zhang, Susan, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer (2022). *OPT: Open Pre-trained Transformer Language Models*. DOI: 10.48550/ARXIV.2205.01068. URL: https://arxiv.org/abs/2205.01068.