Retrievers for the Real World

Ajinkya Deshpande **Jushaan Kalra**

Abstract

Current state of the art retrievers are known to be insufficient for real world use, especially for domain specific documents or noisy datasets. In this report we perform a literature survey of different retrievers, and their applications. We then reproduce results of retrievers for a specific domain - retrieval for Question Answers on research papers. While retrieval + reranking performs well on this task, we perform error analysis and discuss different avenues for performance improvement. Code to reproduce our results is present here: https://github.com/ajdesh2000/LitSearch

1 Literature Survey

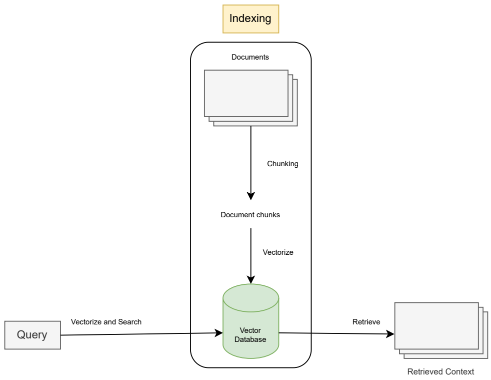

Figure 1: A general Retrieval pipeline

1.1 State of the art Retrievers

Retrieval is an important component in modern Retrieval Augmented Generation (RAG) systems, as the model performance is constrained by the quality of data retrieved. Therefore, many different retrievers have been used to solve passage extraction from a large corpus, such as BM25 (Robertson et al., 2009), SimCSE (Gao et al., 2021) and Dense Passage Retrieval (Karpukhin et al., 2020). Retrievers can be split into two broad categories, Sparse retrievers (such as BM25), which create a sparse vector representation for documents and queries and Dense retrievers, (Such as SimCSE and DPR) which create a dense representation vector for each document and query. Sparse retrievers have the benefit of being simple and fast, which make them relevant for simple queries. Dense retrievers perform better in complex queries requiring information about semantic similarity. We discuss some of the retrievers below

1.1.1 BM25

BM25 (Robertson et al., 2009) is a heuristic based method which uses term frequency and document length to find similar documents. BM25 uses a modified version of *TF-IDF* to account for document length and keyword saturation. It's been shown to perform well for simple and domain specific queries (Cai et al., 2024), possibly because of similarity of domain specific keys which are difficult for neural networks to pick up.

1.1.2 DPR

(Karpukhin et al., 2020) propose DPR, a system which uses 2 independent BERT models to encode and finetune embeddings by maximizing the similarity between correct question-context pairs and minimizing it for negative cases. They take 3 different types of negative examples, the most important of which is taking all the other contexts from the same minibatch of the (query, context) pair. This ensures effective computations due to the nature of the corresponding matrix multiplication. They show marginal improvements on standard datasets, however, it does not work well with long contexts and complicated queries (like with SQUAD). DPR embeddings take up more space than sparse one's like BM25 and are slower to compute

1.1.3 SimCSE

The paper (Gao et al., 2021) proposes a neat way to generate embeddings using a side-effect of dropout and contrastive loss. Because of dropout, passing the same sentence through a dense encoder (such as BERT) would lead to a different representation. Using this fact, they train the model in an unsupervised approach initially and then use a Natural Language Inference dataset to train a supervised version of the model. They show marginal improvements over baselines in both unsupervised and supervised models. However, they don't discuss the effect of more than a single positive pair, essentially merging the training parigigm of DPR and SimCSE. As seen in performance, they struggle with complex queries.

1.1.4 Instruction-Finetuned Text Embeddings

(Su et al., 2022) introduce `Instructor`, a single multitask model that generates task and domain-aware embeddings given a text input and its task instructions. They achieve state-of-the-art performance in various text similarity and retrieval tasks, making them useful for domain specific scenarios. `Instructor` is faster to compute and more effective than other dense retrievers. `Instructor` only use 4 negative examples, and the authors don't mine hard negatives, which is computationally expensive, but could lead to a better training paradigm as negative examples are extremely important to train good retrievers.

1.2 Retrievers in the Real World

In the real world, data can come in many forms. The one size fits all approaches delineated in the previous section may not be sufficient. For such cases, domain specific methods may provide the best performance. Specifically, for retrieval, when the documents have some inherent structure, they can be exploited by methods that take this structure into consideration. The following section, examines cases where the set of documents had some structure, which was utilized by the retrieval method.

1.2.1 Retrieval on code

Code repositories contain complex dependencies and interconnections that present unique challenges for retrieval-augmented generation in code tasks. Recent works have leveraged repository-specific structures to address these challenges.

In CodePlan (Bairi et al., 2023), repository-level coding tasks—such as package migration and error correction—are framed as planning problems requiring multi-step edits across interdependent files. CodePlan uses an incremental dependency analysis and adaptive planning to guide large language models (LLMs) through sequential, context-aware code modifications. The context for context-aware modifications is created by using retrieval methods that leverage the dependency graph. Similarly, the RRR (Deshpande et al., 2024) approach tackles class-level generation within real-world repositories, where dependencies often span multiple files. RRR combines LLMs with static analysis tools to retrieve and integrate cross-file context for generating accurate, functional class code. Evaluated on the RepoClassBench benchmark, RRR demonstrates significant improvements by effectively leveraging repository context, highlighting the value of tailored retrieval methods that account for the intricate structure of code repositories.

1.2.2 Retrieval on tables

Spreadsheet tasks often involve repetitive data processing that end users struggle to automate. By utilizing the structured nature of spreadsheets, the SheetCopilot (Li et al., 2023) agent enhances retrieval and task execution through a domain-specific approach. This system translates natural language commands into a series of atomic actions, representing core spreadsheet functionalities. Using a state machine-based framework, it guides large language models to interpret and manipulate spreadsheet data more accurately. This structured, spreadsheet-specific retrieval approach enables SheetCopilot to complete 44.3% of tasks correctly.

1.3 Retrieval on protein structures

This study (Ma et al., 2023) uses ESM-1b (Rives et al., 2021), a protein language model trained on amino acid sequences, as a retriever, and introduces Retrieved Sequence Augmentation (RSA), which identifies similar protein sequences based on structure and function. This approach taps into protein-specific structural patterns to boost representation and prediction accuracy without requiring computationally expensive alignment methods.

1.3.1 Retrieval on PDFs

Modern knowledge-based question-answering systems increasingly rely on retrieval-augmented generation (RAG) to handle complex queries. However, because much of this knowledge is stored in PDFs, low parsing accuracy often limits RAG effectiveness. Lin (2024) highlights this issue by testing ChatDOC, a RAG system with advanced PDF parsing capabilities. ChatDOC retrieves more accurate and complete segments by better recognizing PDF structure, outperforming traditional RAG systems in 47% of cases and matching them in 38%. This work suggests that improved PDF structure recognition can significantly enhance RAG for professional document retrieval.

1.3.2 Retrieval on financial reports

Financial documents often require specialized retrieval techniques due to their structured, metadata-rich content. FinRAG (Chawla) introduces an improved model, RAPTOR, which utilizes a tree-based retrieval approach with metadata-based clustering by sector, company, and year, aligning retrieval with financial document structure. Evaluated on the FinQA dataset, this method shows enhanced accuracy in answering complex financial queries by drawing on relevant documents more effectively.

Another approach (Yepes et al., 2024) improves retrieval by chunking documents based on structural elements—such as headers and tables—rather than relying on simple paragraph division. This method better preserves context and structure, yielding more accurate RAG-assisted question answering in financial reporting tasks. Both approaches demonstrate that domain-specific structuring significantly enhances retrieval accuracy in finance.

1.3.3 Retrieval on scientific literature

Citation prediction is a popular task with many existing works. The task can vary on the basis of the granularity of the query, with global prediction (query is the entire document) and local prediction (query is the line which cites the other paper). (https://doi.org/10.18653/v1/2020.acl-main.207) uses a citation informed training objective to train the encoder. (Content-based citation recommendation) performs reranking using a specially trained paired scoring model. (Improved Local Citation Recommendation Based on Context Enhanced with Global Information) improves local citation by including the title and abstract in the context. (https://doi.org/10.1007/s11192-020-03561-y) uses a graph neural network to incorporate the neighbor context in the embeddings. (Local Citation Recommendation with Hierarchical-Attention Text Encoder and SciBERT-Based Reranking) uses a hierarchical attention based transformer for text embedding and then performs reranking using SciBert. However, all of these fall under the "citation recommendation task". While useful, this is different from the "literature search" task, which involves a search query explicitly designed to look for relevant papers in a corpus. LitSearch is a recent paper that proposes a novel benchmark with natural language questions about papers in the corpus. In their analysis, retrieval using Grit-LM followed by reranking using GPT-4 worked the best.

2 Project Proposal

We propose using LitSearch as a benchmark for our retrieval methods. LitSearch offers a dataset large enough to thoroughly evaluate retrieval techniques while remaining computationally feasible for a compute-constrained project like ours. Its queries are highly relevant, addressing the increasingly important task of querying research paper databases amid the rapidly expanding body of online literature.

While the paper itself explores a few retrieval methods, we believe that it under explores the re ranking component, that can be exploited in a domain specific scenario. To reproduce the paper, we replace `GPT-4` with `GPT-4o`, a model much cheaper and which shows similar trends of performance as shown in Table 1. In the following sections, we delve deeper into the error analysis of current methods and future directions.

3 Baseline Reproduction

NOTE: Due to the prohibitive cost of using the GPT-4 model used in the original paper, we used GPT-4o-mini. Due to this change there are slight differences in the reproduced numbers, although the overall trend remains the same.
Table 1 compares recall@20 for broad questions and recall@5 and @20 for specific questions across different retrieval and reranking methods, with original and reproduced scores (using GPT-4o-mini) shown in each cell. BM25, the baseline retriever, achieves relatively modest recall, with GritLM significantly outperforming it across all categories. The reranking approaches using GPT-4o-mini, whether in one-hop or direct reranking modes, show marked improvements over BM25 and are competitive with or slightly below GritLM's original performance. Particularly, GPT-4 reranking paired with GritLM consistently

achieves the highest recall, notably enhancing performance on specific author-written questions. These results underscore the benefits of combining strong base retrievers with advanced reranking, although differences between the original and reproduced values suggest minor performance variance when using GPT-4o-mini.

4 Error Analysis

4.1 Example 1

Query: Which paper found that mutual learning benefits multilingual models?

Rank of ground truth: 27

Ground truth title: Towards Higher Pareto Frontier in Multilingual Machine Translation

Ground truth abstract: Multilingual neural machine translation has witnessed remarkable progress in recent years. However, the long-tailed distribution of multilingual corpora poses a challenge of Pareto optimization, i.e., optimizing for some languages may come at the cost of degrading the performance of others. Existing balancing training strategies are equivalent to a series of Pareto optimal solutions, which trade off on a Pareto frontier 1 . In this work, we propose a new training framework, Pareto Mutual Distillation (Pareto-MD), towards pushing the Pareto frontier outwards rather than making trade-offs. Specifically, Pareto-MD collaboratively trains two Pareto optimal solutions that favor different languages and allows them to learn from the strengths of each other via knowledge distillation. Furthermore, we introduce a novel strategy to enable stronger communication between Pareto optimal solutions and broaden the applicability of our approach. Experimental results on the widely-used WMT and TED datasets show that our method significantly pushes the Pareto frontier and outperforms baselines by up to +2.46 BLEU 2 .

Analysis: The model may be unable to rank this paper highly because the abstract does not explicitly contain the words multilingual models and mutual learning. This can be potentially be solved by adding information from the Introduction which contains the lines '... train two multilingual models simultaneously. These two models learn from each other at each training step with knowledge distillation. '

4.2 Example 2

Query: Which paper is among the earliest to train on extensive collection of signing video and subtitle pairs available from online platforms?

Rank of ground truth: 36

Ground truth title: Gloss-Free End-to-End Sign Language Translation

Ground truth abstract: In this paper, we tackle the problem of sign language translation (SLT) without gloss annotations. Although intermediate representation like gloss has been proven effective, gloss annotations are hard to acquire, especially in large quantities. This limits the domain coverage of translation datasets, thus handicapping real-world applications. To mitigate this problem, we design the Gloss-Free End-to-end sign language translation framework (GloFE). Our method improves the performance of SLT in the gloss-free setting by exploiting the shared underlying semantics of signs and the corresponding spoken translation. Common concepts are extracted from the text and used as a weak form of intermediate representation. The global embedding of these concepts is used as a query for cross-attention to find the corresponding information within the learned visual features. In a contrastive manner, we encourage the similarity of query results between samples containing such concepts and decrease those that do not. We obtained state-of-the-art results on large-scale datasets, including OpenASL and How2Sign.

Analysis: Similar to the previous example the abstract does not contain explicitly the words "signing video". Instead using the introduction which contains lines like "... relevant features from the signing video does not match the order of the queries in the translation..." may prove more to be easier for the model. Additionally, there is a temporal component to this question as well since it asks the question "Which paper is among the earliest". The date is not present in the abstract, and this points to the possible usefulness of incorporating other metadata in the keys.

4.3 Example 3

Query: I know about prompt tuning, but have any works tried learning embeddings that are inputted to every transformer layer in a language model?

Rank of ground truth: 98

Ground truth title: Prefix-Tuning: Optimizing Continuous Prompts for Generation

Ground truth abstract: Fine-tuning is the

	Inline-citation			**Author-written**			Avg.	Avg.
	Broad	**Specific**		**Broad**	**Specific**		Broad	Specific
	R@20	R@5	R@20	R@20	R@5	R@20	R@20	R@5
BM25	37.4 / **37.4**	38.5 / **38.5**	55.8 / **55.8**	48.6 / **48.6**	62.6 / **62.6**	73.5 / **73.5**	39.9 / **39.9**	50.0 / **50.0**
Grit-LM	69.7 / **69.7**	67.7 / **67.7**	77.9 / **77.9**	74.3 / **74.3**	82.5 / **82.5**	89.1 / **89.1**	70.8 / **70.8**	74.8 / **74.8**
GPT-4 one-hop (w/ BM25)	62.0 / **52.9**	64.1 / **55.4**	71.6 / **63.2**	74.3 / **62.9**	73.5 / **70.6**	77.7 / **77.7**	64.8 / **55.2**	68.6 / **62.7**
GPT-4 one-hop (w/ Grit-LM)	72.9 / **68.5**	70.3 / **66.9**	78.4 / **81.0**	74.3 / **74.3**	84.4 / **83.4**	87.2 / **88.6**	73.2 / **69.8**	77.0 / **74.8**
GPT-4 reranking (w/ BM25)	54.9 / **52.39**	60 / **59.74**	67.5 / **64.72**	77.1 / **71.43**	76.8 / **72.51**	82.9 / **81.04**	59.9 / **56.69**	68 / **65.84**
GPT-4 reranking (w/ Grit-LM)	74.7 / **70.8**	73.2 / **70.8**	79.9 / **80.3**	77.1 / **80.0**	85.8 / **81.5**	92.4 / **91.9**	75.3 / **72.9**	79.2 / **75.9**

Table 1: In this table, similar to the original, we report the recall@20 (R@20) for broad questions and recall@5 and @20 (R@5, R@20) for specific questions. The cells contain original / reproduced with the reproduced scores bolded. For all the rows in which the original paper used GPT-4, we use GPT-4o-mini for the reproduction, which explains the difference in performance numbers.

de facto way of leveraging large pretrained language models for downstream tasks. However, fine-tuning modifies all the language model parameters and therefore necessitates storing a full copy for each task. In this paper, we propose prefix-tuning, a lightweight alternative to fine-tuning for natural language generation tasks, which keeps language model parameters frozen and instead optimizes a sequence of continuous task-specific vectors, which we call the prefix. Prefix-tuning draws inspiration from prompting for language models, allowing subsequent tokens to attend to this prefix as if it were "virtual tokens". We apply prefix-tuning to GPT-2 for table-totext generation and to BART for summarization. We show that by modifying only 0.1% of the parameters, prefix-tuning obtains comparable performance in the full data setting, outperforms fine-tuning in low-data settings, and extrapolates better to examples with topics that are unseen during training.

Analysis: In this example, it appears that although the abstract gave some hints as to the relevance of the paper, there were several other papers that did something similar that got ranked more highly. For someone familiar with the literature however, it is clear that prefix-tuning is the paper being referenced. This is because for most queries we can apply the prior that more popular and highly cited papers are more likely to be relevant to the query. This points to the possibility of using citation information as a proxy for popularity to change the rank.

5 Future Directions

While `GritLM` and GPT-based reranking have performed well in paper retrieval, our analysis highlights that certain types of queries remain challenging to retrieve effectively. Specifically, relying *solely* on the title and abstract may not always provide enough context for accurate retrieval. To address these limitations, we plan to explore the following areas:

1. **Rerank using additional paper sections** As observed in the error analysis, the absence of explicit keywords in the title and abstract can hinder the retrieval of highly relevant papers. To address this, we propose incorporating more granular sections of the paper, such as the Introduction, Conclusion, Methodology, and Related Work. These sections often contain crucial context that is not always captured in the title or abstract, enabling the system to rerank papers more effectively and accurately. By using the title and abstract for initial retrieval and reranking based on additional sections, we aim to improve the overall relevance of retrieved results while keeping the index small enough to be computationally viable.

2. **Incorporate additional metadata** Metadata such as citation counts and the year of publication can provide important context, especially when the query pertains to the historical significance or timeline of a paper. For example, in the case of a query asking for one of the earliest papers on a specific topic (as in 4.2), incorporating the publication year will be key to ranking older, more foundational papers higher. Adding this metadata will allow us to better answer queries where the historical context or the age of the paper is a critical factor in determining relevance.

3. **Prioritize more prevalent papers:** We also plan to incorporate a priority mechanism where more prevalent papers, such as those with higher citation counts are given preference over less prevalent ones. This could improve retrieval relevance for certain types of queries, as discussed in 4.3

4. **Using other retriever models:** While `GritLM` has been useful for retrieval, exploring domain-specific models such as `SciBERT`, which is

pre-trained on scientific texts, could lead to better performance. SciBERT is particularly designed for tasks that involve scientific papers, and it is trained on a large corpus of research literature and thus has a specialized understanding of the vocabulary and nuances of scientific texts, potentially improving the retrieval and ranking of papers in highly specialized domains.

References

Ramakrishna Bairi, Atharv Sonwane, Aditya Kanade, Vageesh D C, Arun Iyer, Suresh Parthasarathy, Sriram Rajamani, B. Ashok, and Shashank Shet. 2023. Codeplan: Repository-level coding using llms and planning. *Preprint*, arXiv:2309.12499.

Fengyu Cai, Xinran Zhao, Tong Chen, Sihao Chen, Hongming Zhang, Iryna Gurevych, and Heinz Koeppl. 2024. Mixgr: Enhancing retriever generalization for scientific domain through complementary granularity. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10369–10391.

Krrish Chawla. Finrag: A retrieval-based financial analyst.

Ajinkya Deshpande, Anmol Agarwal, Shashank Shet, Arun Iyer, Aditya Kanade, Ramakrishna Bairi, and Suresh Parthasarathy. 2024. Class-level code generation from natural language using iterative, tool-enhanced reasoning over repository. *Preprint*, arXiv:2405.01573.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Vladimir Karpukhin, 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 preprint arXiv:2004.04906*.

Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and Zhaoxiang Zhang. 2023. Sheetcopilot: Bringing software productivity to the next level through large language models. *Preprint*, arXiv:2305.19308.

Demiao Lin. 2024. Revolutionizing retrieval-augmented generation with enhanced pdf structure recognition. *Preprint*, arXiv:2401.12599.

Chang Ma, Haiteng Zhao, Lin Zheng, Jiayi Xin, Qintong Li, Lijun Wu, Zhihong Deng, Yang Lu, Qi Liu, and Lingpeng Kong. 2023. Retrieved sequence augmentation for protein representation learning. *Preprint*, arXiv:2302.12563.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. 2021. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. 2022. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*.

Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Renyu Li. 2024. Financial report chunking for effective retrieval augmented generation. *Preprint*, arXiv:2402.05131.