Binary Document Filtering for Retrieval-Augmented Generation

A dissertation submitted in partial fulfilment for the degree of

Master of Technology

in

Computer Science

by

Sreyan Saha
Roll No. CS2325

Under the supervision of

Dr. Debapriyo Majumdar
Computer Vision and Pattern Recognition Unit (CVPRU), ISI Kolkata

Dr. Rajkiran Panuganti
Ola Krutrim

Indian Statistical Institute, Kolkata
June, 2025

Certificate

This is to certify that the dissertation entitled **"Binary Document Filtering for Retrieval-Augmented Generation"** submitted by **Sreyan Saha** to the Indian Statistical Institute, Kolkata, in partial fulfillment of the requirements for the degree of *Master of Technology in Computer Science*, is an authentic and genuine record of the research work carried out by the candidate under our supervision and guidance. We affirm that the dissertation has met all the necessary requirements in accordance with the regulations of this institute.

Dr. Debapriyo Majumdar
CVPR Unit
Indian Statistical Institute, Kolkata

Dr. Rajkiran Panuganti
Ola Krutrim

Acknowledgement

I would like to express my deepest gratitude to **Dr. Debapriyo Majumdar**, my advisor at the Computer Vision and Pattern Recognition Unit (CVPRU), Indian Statistical Institute, for his insightful guidance, valuable feedback, and continued support throughout this project. His mentorship helped shape the direction and quality of this research.

I am also sincerely thankful to **Dr. Rajkiran Panuganti** of Ola Krutrim, for his co-supervision, constructive suggestions, and technical insight that greatly enhanced the rigor and applicability of this work.

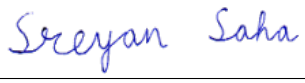
I extend my gratitude to the faculty and staff of the Indian Statistical Institute for providing a nurturing academic environment. Special thanks to my peers and friends for their constant encouragement, collaboration, and stimulating discussions during this journey.

Finally, I am profoundly grateful to my family for their unwavering love, patience, and moral support — none of this would have been possible without them.

Declaration

I, **Sreyan Saha**, bearing Roll No. **CS2325**, hereby declare that the material presented in the dissertation titled **"Binary Document Filtering for Retrieval-Augmented Generation"** represents original work carried out by me for the degree of *Master of Technology in Computer Science* at the Indian Statistical Institute, Kolkata.

I further affirm that no sections of this report have been copied from external sources without proper attribution. I understand that any instance of plagiarism or use of unacknowledged content will result in strict academic consequences.



Sreyan Saha
M.Tech (CS), Roll No. CS2325
Indian Statistical Institute

Abstract

Retrieval-Augmented Generation (RAG) has become a popular technique to enhance Large Language Models (LLMs) with access to external information sources. However, the success of RAG systems critically depends on the relevance and quality of the retrieved documents. In particular, supplying irrelevant or noisy context can lead to degraded downstream generation quality. To address this, our project focuses on improving the **document filtering stage** in a RAG pipeline through **binary relevance classification** — deciding whether a retrieved document is suitable to include in the final context window based on its usefulness in directly answering the user query. We explore a wide range of approaches to this task, including *rule-based retrieval methods* (TF-IDF, BM25), *classical machine learning classifiers* (logistic regression, SVM), *deep neural networks*, and *LLM-based methods*, both in zero-shot and few-shot settings. Our final pipeline leverages instruction-tuned LLMs to act as strict binary classifiers, with a focus on maximizing **precision** over recall, thereby ensuring that only the most relevant and high-quality documents are passed to the generation module. Experiments are conducted on a Reddit-based query-document dataset tailored to subjective and opinion-heavy queries. Our evaluations suggest that LLMs, even without fine-tuning, can outperform traditional methods in this setting, offering a strong foundation for further enhancement through supervised fine-tuning.

Keywords: Retrieval-Augmented Generation, Binary Relevance Classification, Document Filtering, Large Language Models, Precision-Oriented Retrieval, Reddit Dataset, Zero-Shot Inference

Table of Contents

Introduction

Large Language Models (LLMs) such as GPT-4, Claude, and LLaMA have demonstrated impressive capabilities in language understanding and generation. However, they are inherently limited by the static nature of their pretraining data and are unable to reason over or incorporate real-time information, current events, or domain-specific knowledge not seen during training. This limitation severely affects their utility in dynamic applications such as personal assistants, customer support, and social media monitoring.

A simple yet powerful solution to this challenge is the Retrieval-Augmented Generation (RAG) framework, which augments LLMs with a retrieval component. Given a user query, a set of potentially relevant documents is first retrieved from a knowledge base (e.g., a web index, corporate database, or social media corpus). These retrieved documents are then provided as additional context to the LLM, enabling it to generate more informed, accurate, and up-to-date responses.

However, the success of RAG depends critically on the quality of the retrieved context. LLMs have limited context windows and may generate hallucinated or misleading answers if fed irrelevant or noisy documents. Thus, it becomes crucial to filter or rank the retrieved documents to ensure that only the most relevant ones are passed into the LLM.

While first-stage retrievers (e.g., BM25 or dense retrieval using embeddings) can identify loosely relevant documents, they often fail to make nuanced judgments about whether a document is actually helpful in answering a specific query. This motivates a second-stage filtering or reranking component focused on **precision over recall**-we only want to pass documents that are high-quality, accurate, and directly helpful in answering the query.

In this project, we reformulate the second-stage reranking task as a strict **binary relevance classification** problem: for each retrieved document, the model must decide whether it is relevant enough to be included in the context or not. We experiment with a wide range of approaches for this binary decision task:

- **Rule-based methods:** TF-IDF and BM25 are evaluated as baselines for filtering based on lexical overlap.
- **Classical ML methods:** We train logistic regression and support vector machine (SVM) classifiers on hand-engineered features from the query and document.
- **Neural networks:** We use simple feedforward models trained on concatenated embeddings of queries and documents.
- **LLM-based approaches:** Finally, we leverage instruction-tuned LLMs like LLaMA-3 via the Groq API to serve as zero-shot or few-shot binary relevance classifiers. These models are prompted to act as strict judges that prioritize signal over noise and precision over recall.

We evaluate these methods on a Reddit-based dataset containing subjective queries and associated candidate documents. Our results show that LLMs, even in zero-shot settings, can significantly outperform traditional methods on precision and overall downstream quality. While fine-tuning an LLM for this task remains a promising direction for future work, we focus in this project on zero-shot and few-shot instruction prompting due to time and resource constraints.

Ultimately, our goal is to create a high-precision filtering pipeline that improves the reliability and relevance of generated outputs in RAG-based systems, particularly for open-ended or opinion-heavy queries sourced from social media.

Related Work

Recent advancements in Retrieval-Augmented Generation (RAG) have highlighted the critical role of document reranking in improving end-to-end performance. In most modern RAG systems, a retriever (typically dense or hybrid) retrieves a large set of candidate passages, from which a smaller subset is selected for generation. To bridge the gap between retrieval and generation, several recent papers have proposed using Large Language Models (LLMs) as intelligent rerankers to improve the precision of selected documents.

DynamicRAG introduced a retrieval-feedback loop where the LLM dynamically adjusts the number and order of retrieved documents during inference based on internal uncertainty. This dynamic reranking was shown to be more effective than static top-k strategies, especially when generating long-form answers to complex queries.

RankCoT explored the use of Chain-of-Thought reasoning as a reranking mechanism, demonstrating that LLM-generated rationales can serve as effective implicit signals for document relevance. By ranking reasoning chains associated with different passages, RankCoT improved document selection in a way that is interpretable and competitive with traditional rerankers.

RankLLM proposed using zero-shot LLMs to directly score (query, document) pairs for relevance, effectively replacing cross-encoders with LLM prompts. This approach showed that LLMs can be surprisingly competitive as out-of-the-box rerankers when carefully prompted.

Perhaps the most relevant to our work is **RankRAG**, which proposes instruction-tuning a single LLM to jointly perform context ranking and answer generation. RankRAG demonstrates that adding a small fraction of ranking data into the instruction tuning process can outperform traditional expert rerankers and even compete with models fine-tuned on much larger ranking datasets. It operates in a retrieve-rerank-generate loop, where the same LLM scores candidate contexts for relevance and then generates the final response. RankRAG's success strongly supports the dual capability of LLMs to understand relevance and generate coherent answers in a unified framework.

However, while RankRAG provides an elegant unified solution, it still relies on learning from large-scale supervision. Our work diverges in two key ways. First, we frame the problem not as a ranking task but as a binary filtering problem—deciding whether a document should be included at all, independent of relative order. Second, we extensively compare this binary classification approach across multiple model families, including TF-IDF, BM25, classical machine learning classifiers, neural networks, zero-shot and few-shot LLMs. This breadth of comparison is largely absent in the reranking-focused literature.

In many real-world RAG pipelines, especially those constrained by small context windows, the exact order of included documents is often less important than their individual relevance. As long as highly relevant documents are included in the context window, generation quality typically remains stable. This makes binary inclusion - rather than fine-grained ordering - a more natural and practical formulation for high-precision document selection.

In addition, much of the recent work has been concentrated on knowledge-intensive QA datasets like Natural Questions, HotpotQA, and biomedical corpora. Our focus is instead on subjective, user-generated Reddit-style corpora, which pose distinct challenges such as opinion diversity, informal language, and noisy document boundaries. Although some prior work has explored Reddit in RAG pipelines—for example, a two-stage Reddit summarization+RAG setup in medical QA—the task was not framed as document-level filtering and did not benchmark classical or LLM-based methods.

To the best of our knowledge, ours is the first study to (1) frame Reddit-style RAG as a binary document filtering problem, and (2) evaluate the effectiveness of zero-shot and few-shot LLM prompts as high-precision relevance filters in such a subjective and noisy domain. By releasing a small synthetic benchmark dataset and exploring models from traditional IR to LLMs, we aim to provide a practical recipe for lightweight yet effective filtering in real-world low-context RAG applications.

Dataset

To evaluate our document filtering approaches under realistic, subjective, and noisy retrieval conditions, we constructed a custom dataset of query-document pairs specifically tailored to Reddit-style discourse. Since no existing benchmark captures the nuance and opinion diversity of user-generated social content in a RAG setting, we designed our dataset to reflect these challenges while aligning closely with downstream generation behavior.

Query Collection: We curated a set of 50 open-ended natural language queries, reflecting subjective user needs such as lifestyle advice, ethical dilemmas, or social opinions. These queries were selected from a pool of past user interactions within production systems. Care was taken to focus on queries where answers are unlikely to be factual or canonical, but instead benefit from a diversity of perspectives and community-sourced wisdom - the kind of content typically found on platforms like Reddit.

First-Stage Retrieval (L1 Ranker): For each query, we used the SerpAPI search engine interface as our first-stage retriever. It serves as a strong lexical and semantic ranker, surfacing the top 20 Reddit URLs that are likely to be broadly relevant to the query. These search-based candidates form the initial document pool from which our second-stage classifier must select the truly useful entries.

Document Construction: Each Reddit URL retrieved was parsed using PRAW (Python Reddit API Wrapper) to extract key content fields: the title, main body text of the post, and the top five comments by upvote score. This holistic inclusion of community responses, in addition to the original post, helps reflect the full conversational context and mimics the type of passages typically retrieved in user-facing assistant systems.

This process yielded a dataset of 1,000 query-document pairs (50 queries $\times$ 20 documents per query), with a wide variety of document lengths, tones, and writing styles — all characteristics that make subjective relevance classification particularly challenging.

Annotation Process: Rather than relying on generic similarity heuristics, our annotation pipeline was centered around downstream performance. A group of trained human annotators evaluated each document for its task-specific relevance to the query by simulating its use in a RAG pipeline. Specifically, each document was individually provided as context to a generation LLM, and annotators assessed whether the resulting answer meaningfully addressed the query.

Documents that led to informative, accurate, or useful generations were labelled as `RELEVANT`; those that resulted in vague, off-topic, or misleading outputs were labelled as `IRRELEVANT`. This generation-aware methodology ensures that labels are tightly coupled

to actual utility in a RAG setting, rather than abstract topicality.

Label Balance: The final dataset comprises an approximately even split between relevant and irrelevant documents. This balance avoids the pitfalls of class imbalance and enables effective training and evaluation of binary classifiers without additional resampling or weighting.

Motivation for L2 Filtering: While the SerpAPI-based first-stage retrieval typically returns documents that are topically related, it lacks the precision needed for strict relevance filtering. In many cases, documents may match keywords or be loosely associated with the query theme, yet offer little value when passed into a language model for generation. This makes our second-stage binary classifier (L2 ranker) essential — its goal is to act as a high-precision gatekeeper, admitting only those documents that contribute meaningfully to downstream quality.

This dataset — combining real-world style queries, noisy social content, and relevance labels grounded in generation outcomes — serves as a robust testbed for evaluating binary document filtering approaches in realistic RAG workflows.

Methodology

We explore a range of methods for filtering query-document pairs in our RAG pipeline, starting from simple rule-based approaches and progressing to more sophisticated neural techniques. Each method aims to classify whether a given document is contextually useful for a downstream generation model, based on a natural language query. Our evaluation is focused on task-specific binary relevance prediction, with all models trained and tested using the custom dataset described previously.

Rule-Based Baselines: Our first set of methods leverages traditional information retrieval techniques - TF-IDF and BM25 - due to their computational efficiency, interpretability, and speed. These methods inherently produce a real-valued relevance score for each query-document pair. To adapt them for binary classification, we introduce a score threshold: if the similarity score exceeds this threshold, the document is classified as `RELEVANT`; otherwise, it is deemed `IRRELEVANT`.

Threshold Selection Strategy: To determine the optimal threshold, we employ an oracle-style tuning strategy: a small validation set is used to exhaustively search across possible thresholds, selecting the one that maximizes F1 score. This ensures each method is evaluated under its best-case scenario, enabling a fair baseline comparison.

TF-IDF Scoring: TF-IDF (Term Frequency–Inverse Document Frequency) constructs sparse vector representations of queries and documents based on word occurrence statistics. Similarity is computed via cosine distance. While fast and conceptually simple, TF-IDF fails to capture semantic similarity, making it particularly ill-suited for our subjective, opinion-based dataset where exact word overlap is rare. Empirically, TF-IDF achieves only mid-60s F1 scores, reflecting its limitations in understanding meaning beyond surface forms.

BM25 Ranking: BM25 builds on TF-IDF by incorporating term frequency saturation and document length normalization, making it more robust for variable-length inputs. However, our queries are short, and Reddit documents have relatively consistent lengths, limiting the effectiveness of BM25's enhancements. In practice, BM25 yields only marginal improvements over TF-IDF. This is consistent with prior work, where BM25's gains typically emerge in verbose query settings or heterogeneous corpora.

Despite their weaknesses, these lexical baselines are valuable. They establish a performance floor, highlight the importance of semantic reasoning in our task, and underscore the need for more powerful filtering mechanisms in noisy retrieval contexts.

Embedding-Based Similarity

Following the limitations of sparse lexical representations, we transitioned to dense semantic embeddings using Sentence Transformers. These models map text into high-dimensional vector spaces where semantically similar sentences are placed closer together, allowing for better alignment between the intent of a query and the meaning of a document — especially in subjectively worded scenarios.

Model Choice: We utilized the `all-MiniLM-L6-v2` model from the `sentence-transformers` family, a lightweight and efficient transformer-based encoder pre-trained for semantic textual similarity. Each query and document was independently encoded into a fixed-length dense vector, capturing contextual and paraphrastic meanings that traditional bag-of-words models miss.

Scoring Mechanism: Once embedded, the similarity between a query and a document was computed using cosine similarity - a natural choice for comparing unit-normalized vectors in high-dimensional space. This approach yields a scalar score for each pair, indicating semantic proximity.

Threshold Tuning: As in the rule-based models, we again applied our oracle-style thresholding technique to convert similarity scores into binary relevance labels. Using a small validation set, we selected the threshold that maximized F1 score, thereby ensuring optimal separation between relevant and irrelevant pairs.

Results and Insights: The move to embedding-based similarity produced modest improvements in classification performance, with F1 scores rising to the high 60s. This validates the intuition that sentence-level semantic information is crucial for our task, where relevance depends less on exact keyword matches and more on shared intent and discourse themes.

However, while these embeddings capture richer representations, the final decision is still based on a single similarity score - effectively compressing two high-dimensional inputs into a one-dimensional metric. This projection discards potentially valuable structural information embedded in the latent space. In essence, we are still underutilizing the representational capacity of the model by reducing the interaction to a scalar.

Thus, while embedding similarity methods offer better generalization than lexical techniques, they remain limited by the simplicity of their decision function. To truly leverage the expressivity of dense vectors, we turn next to learned classifiers that can reason directly over the full query-document embedding pair.

Traditional ML on SBERT Embeddings

Having established the value of dense semantic representations using Sentence Transformers, we next explored ways to fully exploit the expressive power of these embeddings. Rather than collapsing query-document pairs into a single scalar similarity score, we adopted a richer representation by directly feeding the high-dimensional embeddings into supervised learning models.

Feature Representation: We encoded each query and document using the `all-MiniLM-L6-v2` Sentence Transformer, as before. However, instead of computing similarity between the two embeddings, we concatenated them to form a single feature vector for each query-document pair. This resulted in a fixed-length input that jointly represents both query intent and document content in a format amenable to traditional machine learning algorithms.

Classifier Models: We experimented with lightweight and interpretable models that perform well on moderate-sized, high-dimensional datasets - namely, **Logistic Regression** and **Support Vector Machines (SVM)**. These models are well-suited to linearly or quasi-linearly separable problems, and offer fast training and evaluation times, making them attractive for real-world deployment.

Training Protocol: We performed a random train-test split using the `scikit-learn` library to ensure a representative sampling of the data. Classifiers were trained on the concatenated query-document embeddings using the ground-truth relevance labels as targets. Standard classification metrics were computed on the held-out test set.

Results and Analysis: This approach yielded a substantial boost in performance over similarity-based methods. F1 scores rose to the mid-70s, with **Logistic Regression** slightly outperforming SVM. This improvement is attributable to the model's ability to consider all dimensions of the query and document embeddings simultaneously, capturing more subtle interactions between them than a single similarity score could represent.

The results validate our hypothesis that dense embeddings hold more actionable information than what is revealed by pairwise comparisons alone. By feeding these embeddings directly into classifiers, we begin to unlock their potential for fine-grained decision boundaries.

These models serve as an important bridge between traditional IR-style scoring functions and modern neural approaches, offering a strong baseline that is both performant and computationally efficient.

Neural Classification on SBERT Feature Embeddings

Building on the success of traditional classifiers operating on high-dimensional Sentence Transformer embeddings, we next explored a more expressive model class: neural networks. These models are capable of learning complex, non-linear decision boundaries and allow us to incorporate richer feature engineering to exploit the semantic representations more effectively.

Embedding and Feature Construction: We first upgraded the underlying sentence encoder to a more powerful variant - `all-mpnet-base-v2` - known for its improved semantic alignment and representation quality. For each query-document pair, we computed their respective embedding vectors and engineered a set of composite features by combining them in multiple ways:

- Raw query and document embeddings
- Absolute difference of embeddings
- Element-wise product of embeddings

This yielded a 3072-dimensional feature vector per query-document pair, encapsulating both the raw signals and their interactive relationships.

Model Architecture: We designed a simple yet powerful multi-layer perceptron (MLP) architecture composed of two hidden layers with ReLU activations and dropout for regularization. The network was optimized using binary cross-entropy loss and the Adam optimizer, with a batch size of 32 and training over 20 epochs. Training was conducted using PyTorch on GPU where available.

Training and Evaluation: As before, we performed a random train-test split and trained the model on the concatenated feature vectors. Thanks to the neural network model's capacity to model non-linear relationships and the addition of interaction-aware features, we observed a further lift in performance - pushing F1 scores to the high 70s on the test set.

Analysis: The neural network's improved performance validates the hypothesis that richer interactions between query and document embeddings are critical to identifying true relevance. By leveraging not just the semantic encodings but also their mutual alignment, divergence, and interaction, the model is able to discern patterns missed by linear classifiers.

From Neural to LLM-Based Classification: While the neural MLP architecture effectively captured more of the high-dimensional structure in the embedding space, it still operates on fixed feature vectors derived from precomputed encoders. Our next logical step was to explore *end-to-end* large language model (LLM)-based classification — where the model directly reasons over raw text in context, with full awareness of both syntax and semantics.

In the following section, we describe how we reframe the classification problem as a text-pair inference task and leverage the capabilities of powerful LLMs to achieve even better relevance discrimination in a generation-aware setting.

LLM-Based Relevance Classification

Having extracted most of the predictive power from sentence transformer embeddings and traditional classifiers, we turned to the most powerful tool in our arsenal: instruction-tuned large language models (LLMs). These models offer the unique advantage of reasoning directly over natural language inputs, allowing for nuanced understanding and classification without the need for explicit feature engineering.

Zero-Shot and Few-Shot Prompting: We approached the classification task using both zero-shot and few-shot paradigms. In the zero-shot setting, we provided the model with a carefully crafted system prompt that established clear behavioral expectations. This included the context of acting as a strict second-stage ranker in a retrieval pipeline, with an emphasis on precision and high-quality signal detection. The prompt enforced strict binary classification - only outputting `true` or `false` with no room for uncertainty or elaboration.

In the few-shot variant, we augmented this prompt with handpicked examples from both the positive and negative classes. These examples served as demonstrations of ideal labeling behavior, leveraging the model's instruction-following capabilities to generalize better on the task.

Model and Implementation: We used the `LLaMA 3.1 70B Instruct` model for all experiments reported in this section. The model was deployed using the HuggingFace `transformers` library and configured for fully deterministic decoding (`temperature = 0.0`, `do_sample = False`) to ensure stable binary predictions. Each query-document pair was embedded into a structured chat prompt that included detailed classification instructions, with the model producing a strict `true` or `false` output. This large-scale instruction-tuned LLM served as a high-capacity few-shot and zero-shot relevance filter in our second-stage ranking pipeline.

Each query-document pair was passed to the model as part of a chat-style dialogue, with the system prompt acting as an instruction and the user message containing the inputs. The output was parsed strictly for the keywords `true` or `false`.

Results and Observations: The zero-shot setup already yielded strong performance, with F1 scores reaching the mid-80s. When a few-shot strategy was employed - adding illustrative examples to the prompt - the model achieved even better results, soaring into the high 80s.

Why This Works: Unlike earlier methods which compressed the query-document relationship into fixed-size vectors or scalar similarity scores, LLMs maintain the full structure, semantics, and contextual flow of the text. This enables a form of reasoning

much closer to human-level judgment. Instruction tuning, especially when combined with well-designed prompts, allows the model to internalize task-specific heuristics such as signal concentration, usefulness, and informativeness.

Path Forward - Finetuning Possibilities: We hypothesize that fine-tuning the model on a modest amount of labeled examples from our specific domain could further close the gap to near-human classification accuracy - possibly pushing F1 scores into the 90s. This would simulate the behavior of an expert human annotator with consistent labeling rigor, making the system ideal as a high-precision L2 reranker in any retrieval-augmented generation (RAG) pipeline.

With that, we conclude our exploration across the scoring spectrum - from traditional heuristics to embedding-based classifiers, all the way to state-of-the-art instruction-tuned LLMs. Each step offered incremental improvements, culminating in a robust, highly accurate relevance filter ready for real-world deployment.

Experimental Results

To evaluate the effectiveness of each filtering method, we applied all approaches to the same set of 1,000 query-document pairs from our custom dataset. Ground-truth labels were derived from human assessments of generation quality, as detailed earlier. Each method predicted binary relevance for the full dataset, and results were compared directly against these gold-standard labels.

Evaluation Metrics:

We report standard classification metrics with a focus on both class-specific and overall performance:

- **Accuracy:** The percentage of total predictions that are correct.
- **Precision (for class 1 / Relevant):** The proportion of documents predicted as relevant that were actually relevant.
- **Recall (for class 1 / Relevant):** The proportion of truly relevant documents correctly identified.
- **F1 Score (for class 1):** Harmonic mean of precision and recall for the relevant class.
- **Precision (for class 0 / Irrelevant):** The proportion of predicted irrelevant documents that were truly irrelevant.
- **Recall (for class 0):** The proportion of irrelevant documents correctly identified.
- **F1 Score (for class 0):** Harmonic mean of precision and recall for the irrelevant class.
- **Weighted Averages:** We also report macro-averaged precision, recall, and F1 score across both classes.

This comprehensive breakdown allows us to assess not just overall correctness, but how well each model handles both relevant and irrelevant documents — crucial for practical deployment in RAG pipelines where false positives and false negatives carry asymmetric costs.

Results Summary:

The following table presents the full set of evaluation metrics for each method:

Evaluation Metrics for All Methods

Method	**Acc**	$\mathbf{P}_1$	$\mathbf{R}_1$	$\mathbf{F1}_1$	$\mathbf{P}_0$	$\mathbf{R}_0$	$\mathbf{F1}_0$	**W-P**	**W-R**	**W-F1**
TF-IDF	0.66	0.58	0.74	0.65	0.76	0.58	0.66	0.67	0.66	0.66
BM25	0.68	0.62	0.78	0.69	0.75	0.59	0.66	0.69	0.68	0.68
SBERT Cosine	0.70	0.76	0.63	0.69	0.65	0.78	0.71	0.70	0.71	0.70
SBERT + SVM	0.73	0.80	0.66	0.72	0.66	0.80	0.72	0.73	0.72	0.73
SBERT + LogReg	0.76	0.71	0.85	0.77	0.83	0.68	0.75	0.77	0.76	0.76
MLP (Composite)	0.79	0.86	0.73	0.79	0.72	0.86	0.78	0.80	0.78	0.79
Zero-Shot LLM	0.84	0.82	0.91	0.86	0.90	0.78	0.84	0.85	0.84	0.84
Few-Shot LLM	0.87	0.91	0.89	0.90	0.88	0.91	0.89	0.89	0.90	0.89

Summary:

Results confirm our hypothesis that progressively richer and more expressive models yield improved performance on the relevance classification task. Rule-based methods offer fast but coarse filtering, embedding-based approaches improve on semantic alignment, and supervised models exploit deeper interactions. The best results come from LLM-based classification, which fully leverages linguistic context and task framing to deliver near-annotator-level relevance judgments.

Conclusion and Future Directions

In this report, we investigated a suite of methods for binary document filtering within retrieval-augmented generation (RAG) pipelines, applied to a custom, high-variance Reddit-style dataset. Our goal was to identify approaches capable of reliably filtering irrelevant documents before generation, with an emphasis on high precision and robust semantic alignment.

We began with traditional information retrieval methods such as TF-IDF and BM25, which provided fast, interpretable baselines but lacked the semantic richness needed for deeper relevance assessment. Embedding-based techniques using SBERT improved relevance sensitivity, especially when paired with lightweight classifiers like SVM and logistic regression. We further enhanced this by introducing a neural MLP architecture that fused lexical, semantic, and interaction-based features, yielding substantial gains in both precision and F1.

The most promising results, however, came from large language models (LLMs). Both zero-shot and few-shot LLM-based classification significantly outperformed all other methods, with the few-shot LLM achieving an F1 score of 0.90 for relevant documents—approaching human annotator performance. These results reinforce the value of task framing, prompt design, and contextual understanding inherent in modern LLMs, even without task-specific tuning.

Future Directions:

While our few-shot LLM approach delivered state-of-the-art results in this setting, there remain promising avenues for further enhancement:

- **LLM Fine-Tuning:** We hypothesize that instruction-tuning or supervised fine-tuning on even a modest set of high-quality labeled pairs could yield improvements over few-shot prompting, particularly in edge cases or ambiguous queries where in-context examples may fall short.
- **Cost-Efficiency Tradeoffs:** LLM-based filtering, while effective, is resource-intensive. Future work could explore student-teacher distillation or hybrid cascades (e.g., fast filter then LLM re-ranker) to retain high performance while reducing latency and inference cost.
- **Domain Generalization:** Our methods were evaluated on a Reddit-like dataset. A natural next step is to assess generalization across other domains—e.g., technical support, biomedical QA, or legal reasoning—to validate robustness and adaptability.

- **Multi-Document Filtering:** Current methods treat each document-query pair independently. Extending filtering strategies to consider document sets jointly (e.g., redundancy detection, diversity-aware selection) may further boost downstream generation quality.

Closing Thoughts:

Effective filtering is foundational to scalable and trustworthy RAG systems. By rigorously comparing rule-based, embedding-driven, neural, and LLM-based approaches, we have mapped out a practical accuracy-efficiency frontier and demonstrated the transformative potential of LLMs for fine-grained document relevance classification. As language models become more accessible and customizable, we anticipate that tailored filtering strategies—guided by domain knowledge and empirical rigor—will play an increasingly central role in next-generation information retrieval and generation pipelines.

Bibliography

1. Jiashuo Sun, Xianrui Zhong, Sizhe Zhou, and Jiawei Han. *DynamicRAG: Leveraging Outputs of Large Language Model as Feedback for Dynamic Reranking in Retrieval-Augmented Generation.* arXiv preprint arXiv:2505.07233, 2025.

2. Mingyan Wu, Zhenghao Liu, Yukun Yan, Xinze Li, Shi Yu, Zheni Zeng, Yu Gu, and Ge Yu. *RankCoT: Refining Knowledge for Retrieval-Augmented Generation through Ranking Chain-of-Thoughts.* arXiv preprint arXiv:2502.17888, 2025.

3. Sahel Sharifymoghaddam, Ronak Pradeep, Andre Slavescu, Ryan Nguyen, Andrew Xu, Zijian Chen, Yilin Zhang, Yidi Chen, Jasper Xian, and Jimmy Lin. *RankLLM: A Python Package for Reranking with LLMs.* In Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), Padua, Italy, July 2025.

4. Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. *RankRAG: Unifying Context Ranking with Retrieval-Augmented Generation in LLMs.* arXiv preprint arXiv:2407.02485, 2024.