Performance Evaluation of Rankers and RRF Techniques for Retrieval Pipelines

Ashwin Mathur Varun Mathur
{ashwinxmathur, varunm500}@gmail.com

February 12, 2024

1 Introduction

A RAG pipeline can be tuned in many ways to give more relevant answers. One important way is to improve the relevance of the retrieved context which is input to the LLM. This ensures that the generated answers are coherent and consistent with the context in the original documents.

In this study, we set out to answer the following questions:

- **Question 1**: How can we optimize the way RAG selects and utilizes information (retrieval) to enhance its performance?
- **Question 2**: How do we fill the LLM's context window to maximize the quality of the answers?
- **Question 3**: How do using rankers and RRF techniques help in improving the quality of the retrieved documents?

LFQA question-answering and RAG demands a context window filled with high-quality, varied, relevant, and non-repetitive paragraphs.

A context window is a textual range around a target token that a large language model (LLM) can process at the time the information is generated. Typically, the LLM manages the context window of a textual sequence, analyzing the passage and interdependence of its words, as well as encoding text as relevant responses.

The context window of LLM's is the number of tokens the model can take as input when generating responses. Larger context windows improve LLM performance and their usefulness across various applications.

By considering the neighboring words within a context window, language generation models can produce coherent and contextually relevant textual outputs, thereby enhancing the quality of generated content for various applications.

In the intricate world of LFQA and RAG, making the most of the LLM's context window is paramount. Any wasted space or repetitive content limits the depth and breadth of the answers we can extract and generate. It's a delicate balancing act to lay out the content of the context window appropriately. We present different approaches to solving this problem with the help of different rankers.

LLM's provide the best results when given fewer, more relevant documents in the context, rather than large numbers of unfiltered documents.

The Diversity Ranker (Carbonell and Goldstein 1998) enhances the diversity of the paragraphs selected for the context window. Lost In The Middle Ranker, usually positioned after Diversity Ranker in the pipeline, helps to mitigate the LLM performance degradation observed when models must access relevant information in the middle of a long context window.

The Lost In The Middle Ranker (Liu et al. 2023) optimizes the layout of the selected documents in the LLM's context window. It optimizes the placement of the most relevant paragraphs in the context window, making it easier for the model to access and utilize the best-supporting documents.

Transformers Similarity Ranker (Reimers and Gurevych 2019) ranks Documents based on how similar they are to the query. It uses a pre-trained cross-encoder model to embed both the query and the Documents. It then compares the embeddings to determine how similar they are. The result is a list of Document objects in ranked order, with the Documents most similar to the query appearing first.

In our study, we consider the following cases:

- Dense Retrieval
- Hybrid Retrieval
- Dense + Rankers Retrieval
- Hybrid + Rankers Retrieval

With the addition of three rankers, viz., DiversityRanker, LostInTheMiddleRanker, Similarity rankers and RRF techniques, we aim to address these challenges and improve the answers generated by the LFQA/RAG pipelines. We have done a comparative study of adding different combinations of rankers in a Retrieval pipeline and evaluated the results on four metrics, viz., Normalized Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen 2000), Mean Average Precision (MAP)(Cormack and Lynam 2006), Recall and Precision (Hull 1993).

2 FIQA dataset

The FiQA (Financial Opinion Mining and Question Answering) dataset has a corpus, queries and qrels (relevance judgments file). The FiQA dataset has roughly 6,000 questions and 57,000 answers. They are in the following format:

- Corpus file: a .jsonl file (jsonlines) that contains a list of dictionaries, each with three fields _id with unique document identifier, title with document title (optional) and text with document paragraph or passage. For example: "_id": "doc1", "title": "Albert Einstein", "text": "Albert Einstein was a German born....".
- Queries file: a .jsonl file (jsonlines) that contains a list of dictionaries, each with two fields _id with unique query identifier and text with query text. For example: "_id": "q1", "text": "Who developed the mass-energy equivalence formula?".
- Qrels file: a .tsv file (tab-seperated) that contains three columns, i.e. the query-id, corpus-id and score in this order.

3 Choice of Embedding Model

The large embedding models like INSTRUCTOR and bge-large perform better and give the best Retrieval. Below is a table which the comparative analysis of embedding models based on their size, embedding dimension, max sequence length and Retrieval average and STS average. The choice of the embdding model was based on the ranking in the MTEB Leaderboard.

Model Name	Size	Embedding Dimension	Max Sequence Length	Average	Retrieval Average	STS Average
INSTRUCTOR-xl	4.96GB	768	512	61.59	83.15	31.84
BAAI/bge-large-en-v1.5	1.34GB	1024	512	75.53	82.4	31.07
BAAI/bge-small-en-v1.5	0.13GB	384	512	62.17	51.68	81.59
intfloat/e5-small-v2	0.13GB	384	512	59.93	49.04	80.39
jina-embeddings-v2-base-en	0.27GB	768	8192	60.38	80.7	31.6
jinaai/jina-embeddings-v2-small-en	0.07GB	512	8192	58	45.14	80
all-mpnet-base-v2	0.44GB	768	514	57.78	43.81	80.28

Table 1: Embedding Model Comparison

4 Metrics Used in the Analysis of the Rankers and RRF

- **Precision**: Precision is a measure of how many of the retrieved or predicted items are relevant to a query or task. It is calculated as the ratio of relevant items retrieved to the total number of items retrieved. High precision means the system is good at returning relevant results while minimizing irrelevant ones.

  Precision = (Number of Relevant Items Retrieved) / (Total Number of Items Retrieved)

- **Recall**: Recall assesses how effectively a system retrieves all the relevant items from a dataset. It is calculated as the ratio of relevant items retrieved to the total number of relevant items in the dataset. High recall indicates that the system can find most of the relevant items.

  Recall = (Number of Relevant Items Retrieved) / (Total Number of Relevant Items in the Dataset)

- **Average Precision**: AP is a more refined metric that considers how well relevant items are ranked within the list of retrieved items. It quantifies the area under the precision-recall curve for a single query or task. In other words, it evaluates the precision at different recall levels and averages these values. AP varies from 0 (worst) to 1 (best).

  AP is calculated as the average of precision values at each relevant position in the ranked list of retrieved items. For instance, if you have five relevant items retrieved at positions 2, 5, 7, 8, and 10 in the ranked list, you calculate precision at each position and average them to get the AP.

- **Mean Average Precision (MAP)**: Mean Average Precision is a measure of the precision of a ranking system, taking into account the number of relevant items in the ranked list. It is calculated by averaging the precision at each position in the ranked list, where precision is defined as the number of relevant items in the list up to that position divided by the total number of items in the list up to that position. MAP ranges from 0 to 1, with higher values indicating better performance.

  MAP takes AP further by calculating the average AP across multiple queries or tasks. It provides an overall measure of the system's performance across diverse queries or tasks. MAP is particularly useful when you want to evaluate how well a system performs across various information needs.

  MAP = (1 / Total Number of Queries or Tasks) * $\sum$ (AP for each Query or Task)

- **NDCG Metric** NDCG (normalized discounted cumulative gain): NDCG is a measure of the effectiveness of a ranking system, taking into account the position of relevant items in the ranked list. It is based on the idea that items that are higher in the ranking should be given more credit than items that are lower in the ranking. NDCG is calculated by dividing the discounted cumulative gain (DCG) of the ranked list by the DCG of the ideal ranked list, which is the list with the relevant items ranked in the most optimal order. NDCG ranges from 0 to 1, with higher values indicating better performance.

  NDCG provides the ability to fine-tune which ranks are more valuable than others, and account for a scale of relevancy scores (graded relevance). While NDCG overcomes the shortcomings of MAP, it is limited by actual data and partial feedback and thus requires a more manual data-cleaning process for an accurate calculation.

  Normalized Discounted Cumulative Gain or NDCG is a metric of ranking quality or the relevance of the top listed products. The principle of NDCG is that the more relevant products must be ranked better than the irrelevant products. The higher NDCG indicates that the relevant products are ranked higher.

- **Mean Reciprocal Rank:** MRR (mean reciprocal rank): MRR is a measure of the rank of the first relevant item in a ranked list. It is calculated by taking the reciprocal of the rank of the first relevant item, and averaging this value across all queries or users. For example, if the first relevant item for a given query has a rank of 3, the MRR for that query would be 1/3. MRR ranges from 0 to 1, with higher values indicating better performance.

5 Rankers Used in the Analysis of the Retrieval Pipelines

Lost in the Middle Ranker: The Lost In The Middle Ranker (Liu et al. 2023) optimizes the layout of the selected documents in the LLM's context window. This component is a way to work around a problem identified in recent research that suggests LLMs struggle to focus on relevant passages in the middle of a long context. The Lost In The Middle Ranker alternates placing the best documents at the beginning and end of the context window, making it easy for the LLM's attention mechanism to access and use them. This ranker ranks documents based on the "Lost in the Middle" order, designed to position "the best" documents (low index in the given list of documents) at the beginning and the end of the resulting list while placing "the worst" documents (high index in the given list of documents) in the middle.

Diversity Ranker: The Diversity Ranker (Carbonell and Goldstein 1998) is designed to maximize the variety of given documents. It does so by selecting the most semantically similar document to the query, then selecting the least similar one, and continuing this process with the remaining documents until a diverse set is formed. It operates on the principle that a diverse set of documents can increase the LLM's ability to generate answers with more breadth and depth. The DiversityRanker uses sentence transformers to calculate the similarity between documents. The sentence transformers library offers powerful embedding models for creating meaningful representations of sentences, paragraphs, and even whole documents. These representations, or embeddings, capture the semantic content of the text, allowing us to measure how similar two pieces of text are.

Similarity Ranker: The Similarity Ranker (Reimers and Gurevych 2019) ranks Documents based on how similar they are to the query. It uses a pre-trained cross-encoder model from the Hugging Face Hub to embed both the query and the Documents. It then compares the embeddings to determine how similar they are. The result is a list of Document objects in ranked order, with the Documents most similar to the query appearing first.

Reciprocal Rank Fusion: The Reciprocal Rank Fusion (RRF) (Cormack, Clarke, and Buettcher 2009) is a technique for combining the ranks of multiple search result lists to produce a single, unified ranking. Developed in collaboration with the University of Waterloo (CAN) and Google, RRF, in the words of its authors, "yields better results than any individual system, and better results than standard" reranking methods.

By combining ranks from different queries, we increase the chances that the most relevant documents will appear at the top of the final list. RRF is particularly effective because it does not rely on the absolute scores assigned by the search engine but rather on the relative ranks, making it well-suited for combining results from queries that might have different scales or distributions of scores. Typically, RRF has been used to blend lexical and vector results.

6 Adding Rankers to Improve Performance for Dense Retrieval

The Diversity Ranker enhances the diversity of the paragraphs selected for the context window. Lost In The Middle Ranker, usually positioned after Diversity Ranker in the pipeline, helps to mitigate the LLM performance degradation observed when models must access relevant information in the middle of a long context window.

Diversity Ranker ensures that the generated answer is based on diverse documents. It uses a sentence transformer model to calculate the semantic representation (embedding) for each document. Then, it ranks the documents so that each subsequent document is the least similar to the ones it already selected. This results in a diverse set of documents.

It can be used in combination with other rankers. It is to be placed after the similarity ranker, like Sentence Transformers Ranker, but before the Lost In The Middle Ranker. Such a setup is typical for the long form question answering task.

The Diversity Ranker uses a greedy local approach to select the next document in order, which might not find the most optimal overall order for the documents. Diversity Ranker focuses on diversity more than relevance, so it should be placed in the pipeline after another component like TopPSampler or another similarity ranker that focuses more on relevance. By using it after a component that selects the most relevant documents, we ensure that we select diverse documents from a pool of already relevant documents.

The Lost In The Middle Ranker optimizes the layout of the selected documents in the LLM's context window. This component is a way to work around a problem identified in recent research that suggests LLM's struggle to focus on relevant passages in the middle of a long context. The Lost In The Middle Ranker alternates placing the best documents at the beginning and end of the context window, making it easy for the LLM's attention mechanism to access and use them.

The Lost In The Middle Ranker is best positioned as the last ranker in a RAG pipeline as the given documents are already selected based on similarity (relevance) and ordered by diversity.

First each ranker (Diversity, Lost in The Middle and Similarity) is added individually and evaluated on the NDCG, MAP, Recall and Precision scores.

Then we add the pairs of rankers, first the Similarity Ranker with the Diversity Ranker, and then the Similarity Ranker with the Lost in the Middle Ranker.
Finally, all the three rankers are added and evaluated. The Lost in The Middle Ranker is added at at the end since it orders the most relevant documents for Retrieval.

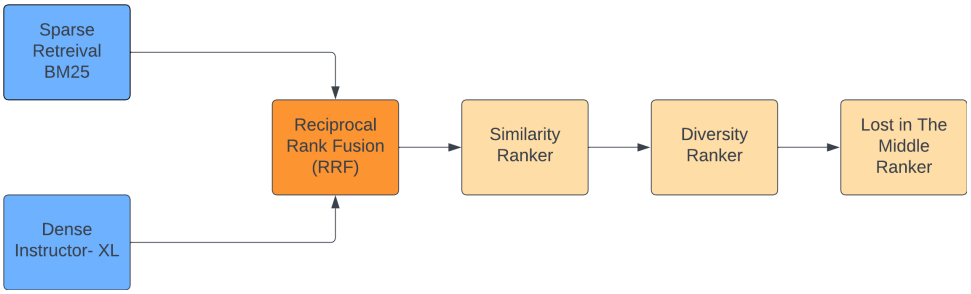

Figure 1: Summary of Techniques

7 Adding Rankers to Improve Performance for Dense Retrieval for Instructor-XL

7.1 Retrieval Pipeline for Dense + Diversity Ranker

Figure 2: Dense + Diversity Ranker Pipeline

Adding a Diversity Ranker results in a NDCG score of 0.425 which is the highest and gives the best performance.

7.2 Retrieval Pipeline for Dense + Lost in The Middle Ranker

Figure 3: Dense + Lost in The Middle Ranker Pipeline

Adding a Lost in The Middle Ranker results in a NDCG score of 0.425 which is the highest and gives the best performance.

7.3 Retrieval Pipeline for Dense + Similarity Ranker

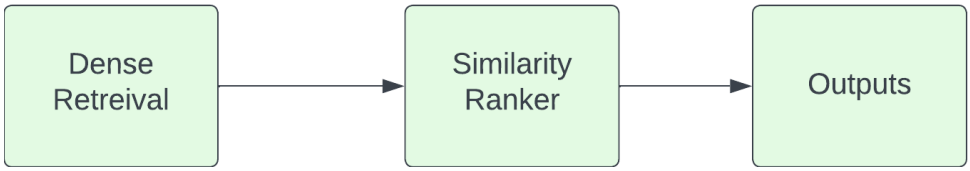

Figure 4: Dense + Similarity Ranker Pipeline

Adding a Similarity Ranker results in a NDCG score of 0.4031 which is lower than that of adding a Diversity and Lost in The Middle Ranker.

7.4 Retrieval Pipeline for Dense + Similarity Ranker + Diversity Ranker

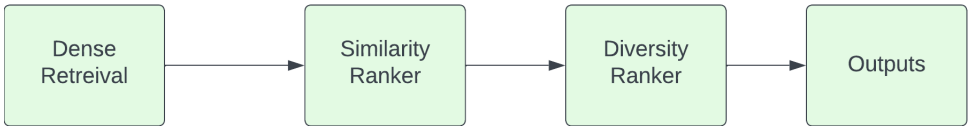

Figure 5: Dense + Similarity Ranker + Diversity Ranker Pipeline

Adding a Similarity and Diversity ranker results in a NDCG score of 0.3861. It performs lower than adding a Diversity or Lost in The Middle Ranker individually to the pipeline.

7.5 Retrieval Pipeline for Dense + Similarity Ranker + Lost in The Middle

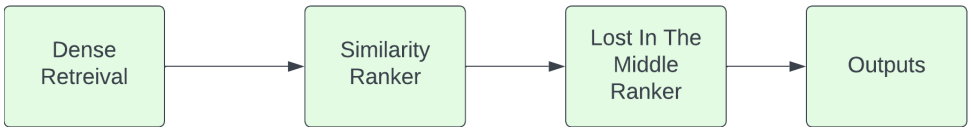

Figure 6: Dense + Similarity Ranker + Lost in The Middle Pipeline

Adding a Similarity and Lost in The Middle ranker results in a NDCG score of 0.4031. It performs lower than adding a Diversity or Lost in The Middle Ranker individually to the pipeline.

7.6 Retrieval Pipeline for Dense + Similarity Ranker + Diversity Ranker + Lost in The Middle

Figure 7: Dense + Similarity Ranker + Diversity Ranker + Lost in The Middle Ranker Pipeline

Adding a Similarity, Diversity and Lost in The Middle ranker results in a NDCG score of 0.3861. It performs lower than adding a Diversity or Lost in The Middle Ranker individually to the pipeline. Adding all the three rankers to the pipeline does not result in a improvement in performance.

8 Adding Rankers to Improve Performance for Hybrid Retrieval for Instructor-XL

First each ranker (Diversity, Lost in The Middle and Similarity) is added individually and evaluated on the NDCG, MAP, Recall and Precision scores. Then we add the pairs of rankers. First the Similarity

Ranker with the Diversity Ranker and then the Similarity Ranker with the Lost in the Middle Ranker. Finally all the three rankers are added and evaluated. The Lost in The Middle Ranker is added at at the end since orders the most relevant documents for Retrieval.

8.1 Dense + BM25 Retrieval with RRF

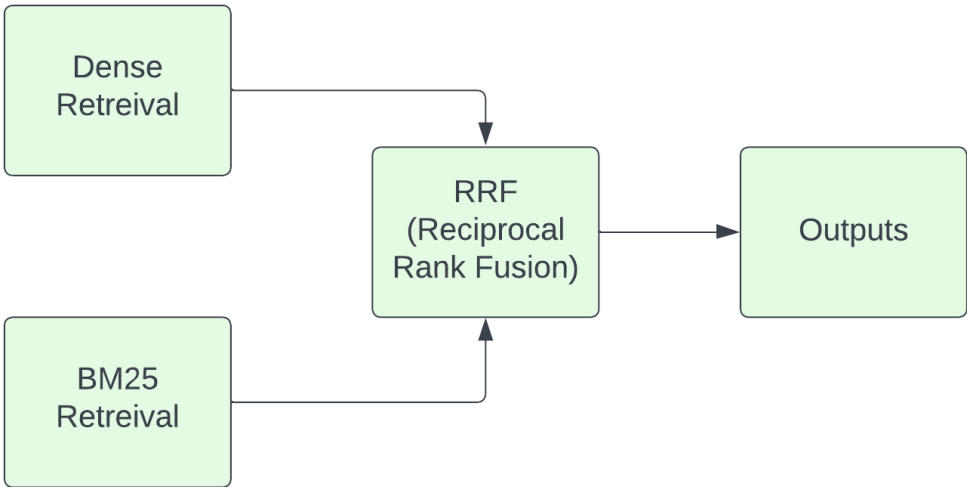

Figure 8: Dense + BM25 Retrieval with RRF

In the BM25 Retrieval combined with RRF pipeline the NDCG score obtained is 0.3437.

8.2 Dense + BM25 Retrieval with RRF + Similarity Ranker

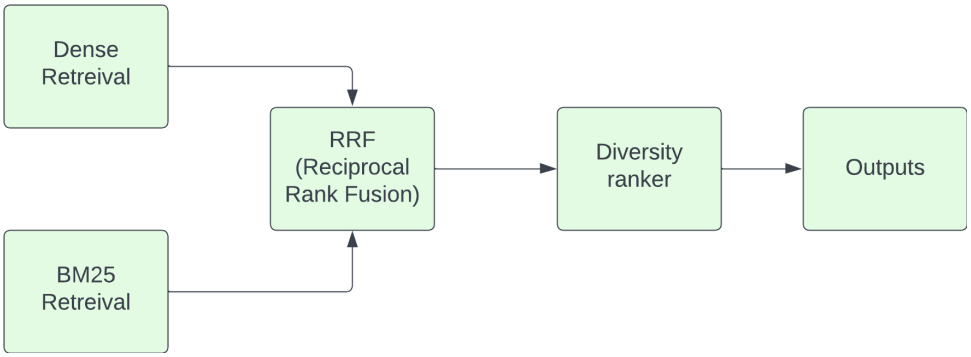

Figure 9: Dense + BM25 Retrieval with RRF + Similarity Ranker

In the BM25 Retrieval combined with RRF pipeline with a Similarity Ranker the NDCG score obtained is 0.3861. This is the highest score obtained and has the best performance.

8.3 Dense + BM25 Retrieval with RRF + Diversity Ranker

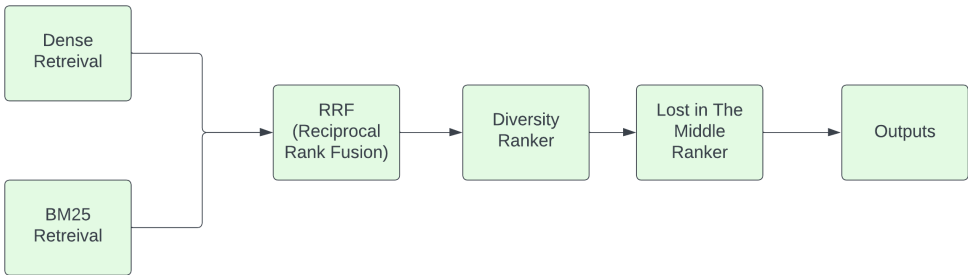

Figure 10: Dense + BM25 Retrieval with RRF + Diversity Ranker

In the BM25 Retrieval combined with RRF pipeline adding a Diversity Ranker gives a NDCG score of 0.2626. This performers lower than adding the Similarity and Lost in The Middle ranker sequentially to the pipeline.

8.4 Dense + BM25 Retrieval with RRF + Lost in The Middle Ranker

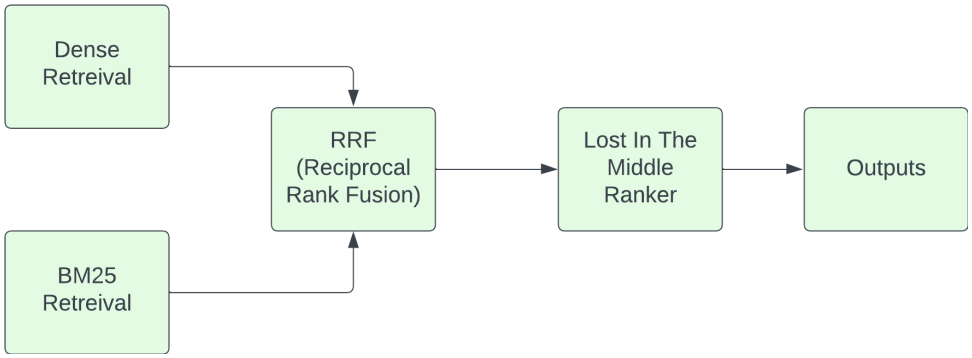

Figure 11: Dense + BM25 Retrieval with RRF + Lost in The Middle Ranker

In the BM25 Retrieval combined with RRF pipeline adding a Lost in The Middle ranker gives an NDCG score of 0.3437.

8.5 Dense + BM25 Retrieval with RRF + Similartiy Ranker + Diversity Ranker

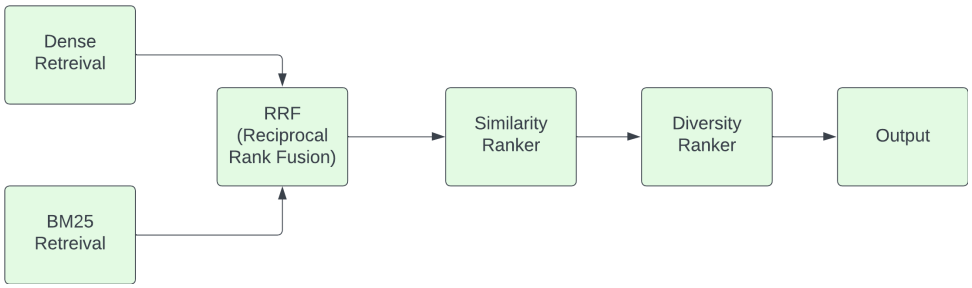

Figure 12: Dense + BM25 Retrieval with RRF + Similartiy Ranker + Diversity Ranker:

In the BM25 Retrieval combined with RRF pipeline adding a Lost in The Middle ranker and Diversity Ranker gives an NDCG score of 0.3861. This gives similar performance to adding the Similarity and Lost in The Middle Ranker sequentially to the pipeline.

8.6 Dense + BM25 Retrieval with RRF + Similarity Ranker + Lost in The Middle Ranker

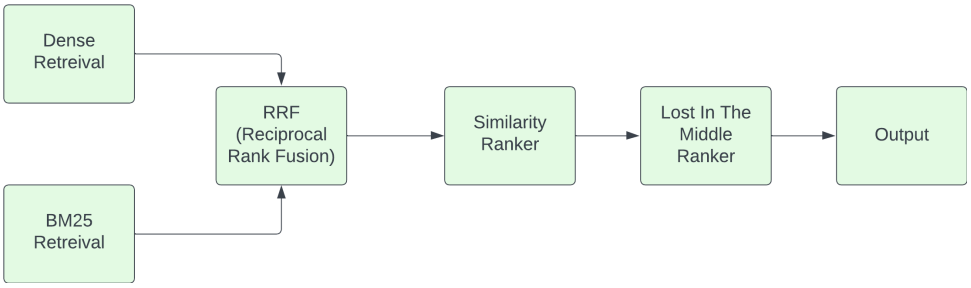

Figure 13: Dense + BM25 Retrieval with RRF + Similarity Ranker + Lost in The Middle Ranker:

In the BM25 Retrieval combined with RRF pipeline adding a Lost in The Middle ranker and Similarity Ranker gives a NDCG score obtained is 0.3861. This gives similar performance to adding the Similarity and Lost in The Middle Ranker sequentially to the pipeline.

8.7 Dense + BM25 Retrieval with RRF + Similarity Ranker + Diversity Ranker + Lost in The Middle Ranker

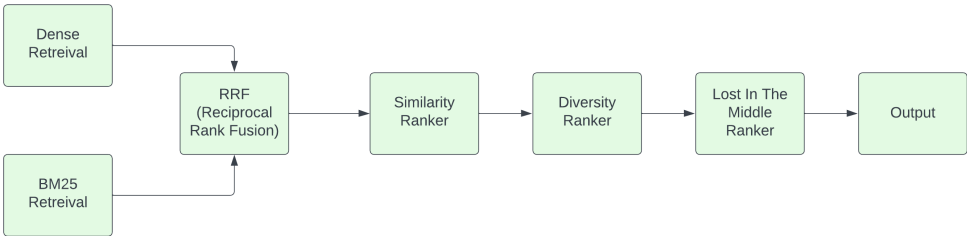

Figure 14: Pipeline for Dense + BM25 Retrieval with RRF + Similarity Ranker + Diversity Ranker + Lost in The Middle Ranker:

In the BM25 Retrieval combined with RRF pipeline with a Lost in The Middle ranker and Similarity Ranker and Diversity Ranker the NDCG score obtained is 0.3861. Adding all the three rankers to the pipeline does not improve performance but performs similarly to that of adding each ranker individually to the pipeline.

9 NDCG Scores for the Top Retrieved Results using Instructor-XL

9.1 NDCG Score for the Top 3 Retrieved Results

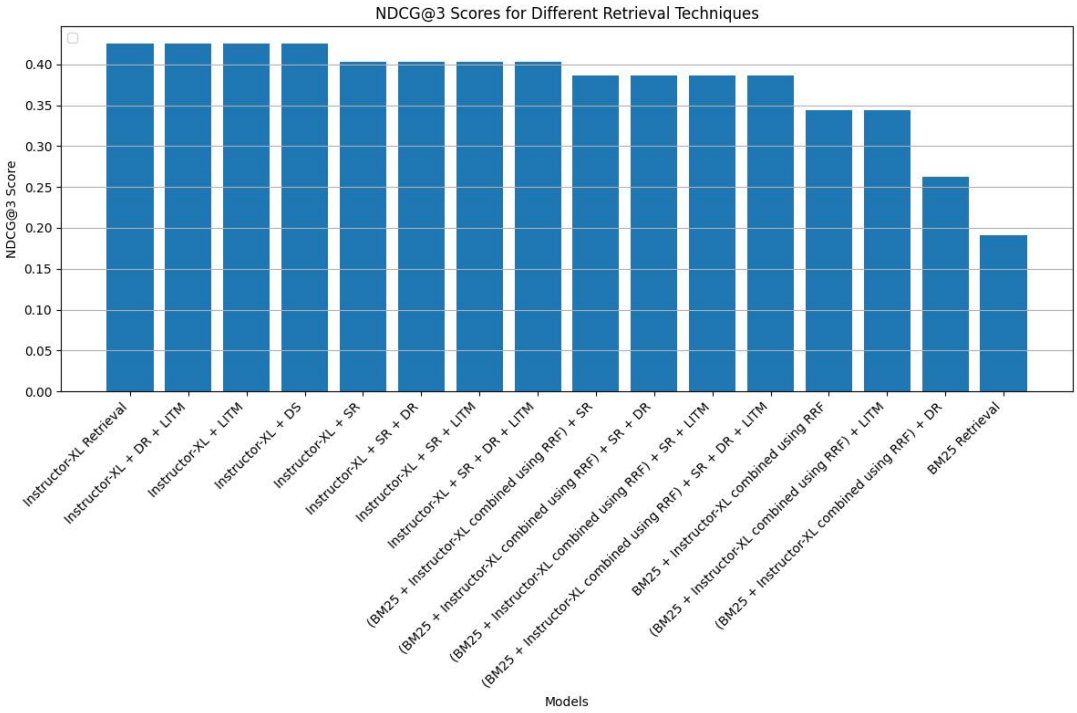

Figure 15: NDCG Score for the Top 3 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest NDCG score of 0.425.
- Adding the Diversity, Similarity and Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave an NDCG score of 0.4031.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave an NDCG score of 0.4031.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

9.2 NDCG Score for the Top 5 Retrieved Results

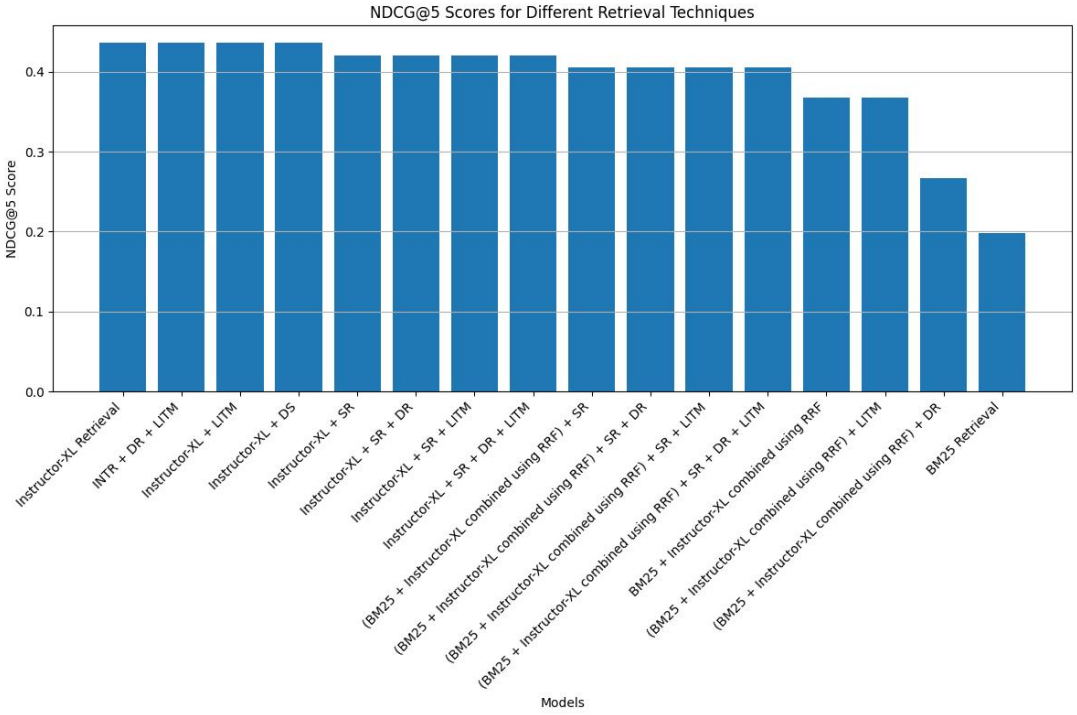

Figure 16: NDCG Score for the Top 5 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest NDCG score of 0.436.
- Adding the Diversity, Similarity and Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave an NDCG score of 0.4203.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave an NDCG score of 0.4203.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

9.3 NDCG Score for the Top 7 Retrieved Results

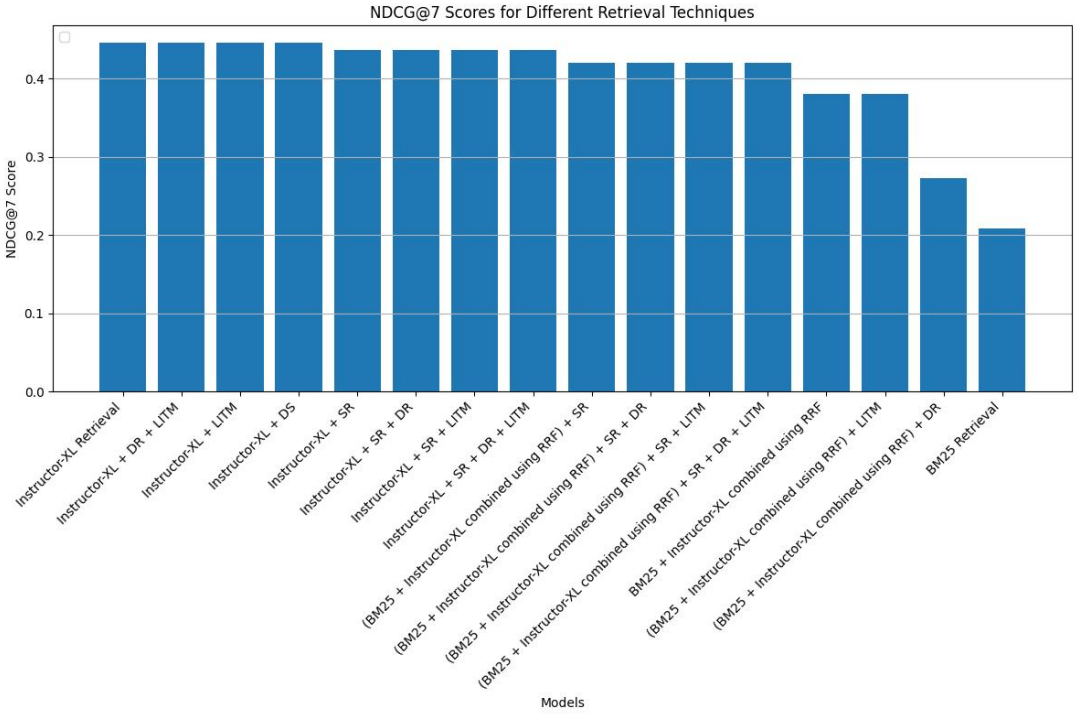

Figure 17: NDCG Score for the Top 7 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest NDCG score of 0.4456.
- Adding the Diversity, Similarity and Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave an NDCG score of 0.4372.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave an NDCG score of 0.4372.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

9.4 NDCG Score for the Top 10 Retrieved Results

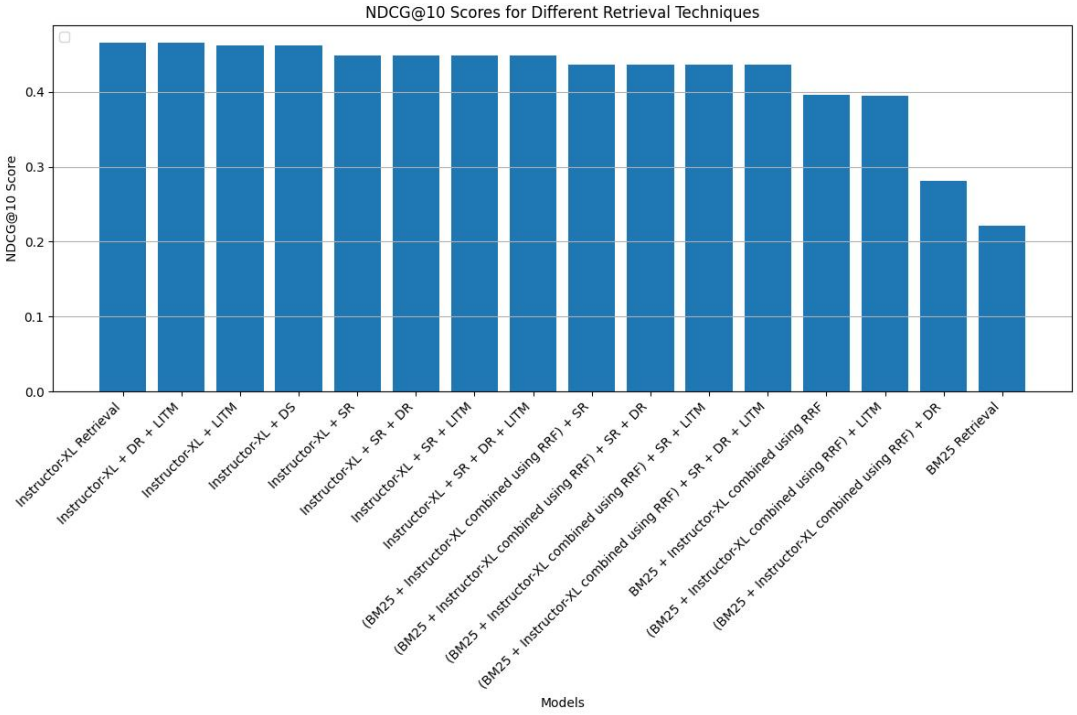

Figure 18: NDCG Score for the Top 10 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest NDCG score of 0.4652.
- Adding the Diversity, Similarity and Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave an NDCG score of 0.4491.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave an NDCG score of 0.4491.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

10 MAP Score for the Top Retrieved Results using Instructor-XL

10.1 MAP Score for the Top 3 Retrieved Results

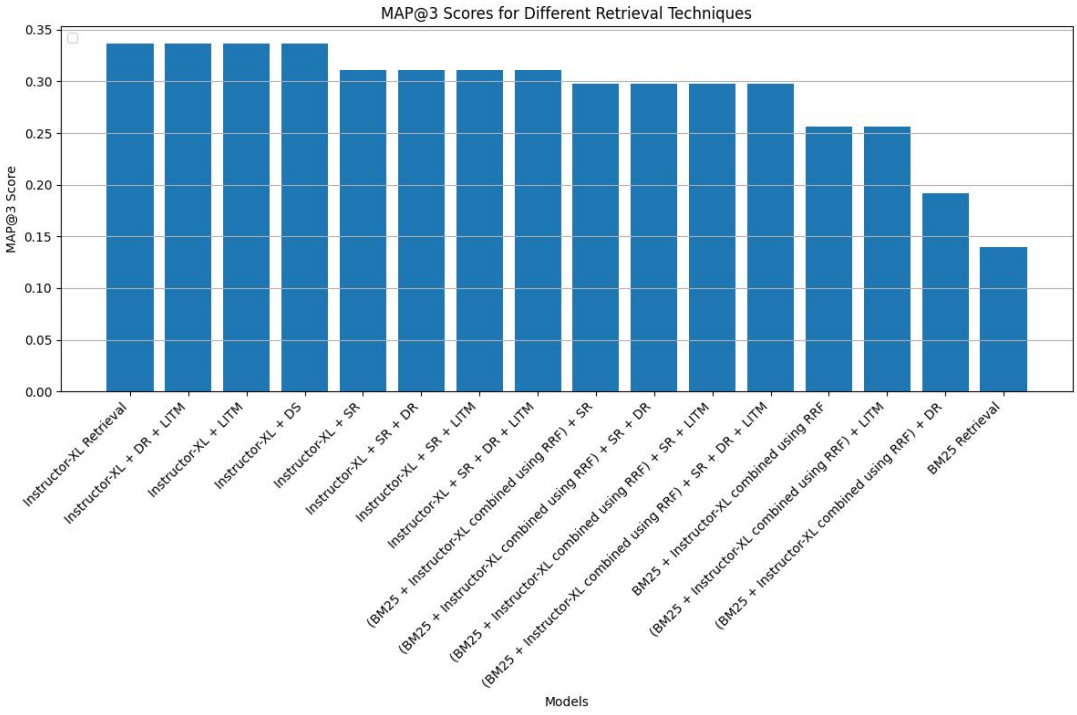

Figure 19: MAP Score for the Top 3 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest MAP score of 0.3362.
- Adding the Diversity, Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker to the Dense Retrieval, resulted in a MAP score of 0.3109, which is lower than the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave an MAP score of 0.3109.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave an MAP score of 0.3109.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

10.2 MAP Score for the Top 5 Retrieved Results

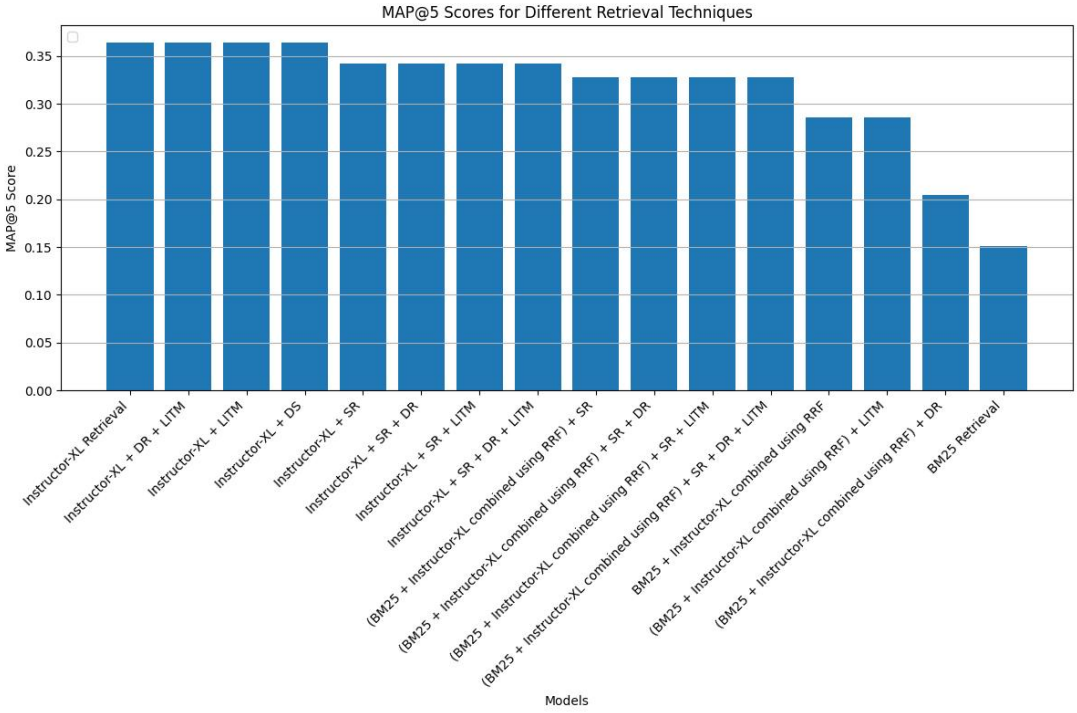

Figure 20: MAP Score For The Top 5 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest MAP score of 0.3637.
- Adding the Diversity, Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker to the Dense Retrieval, resulted in a MAP score of 0.3422, which is lower than the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave an MAP score of 0.3422.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave an MAP score of 0.3422.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

10.3 MAP Score for the Top 7 Retrieved Results

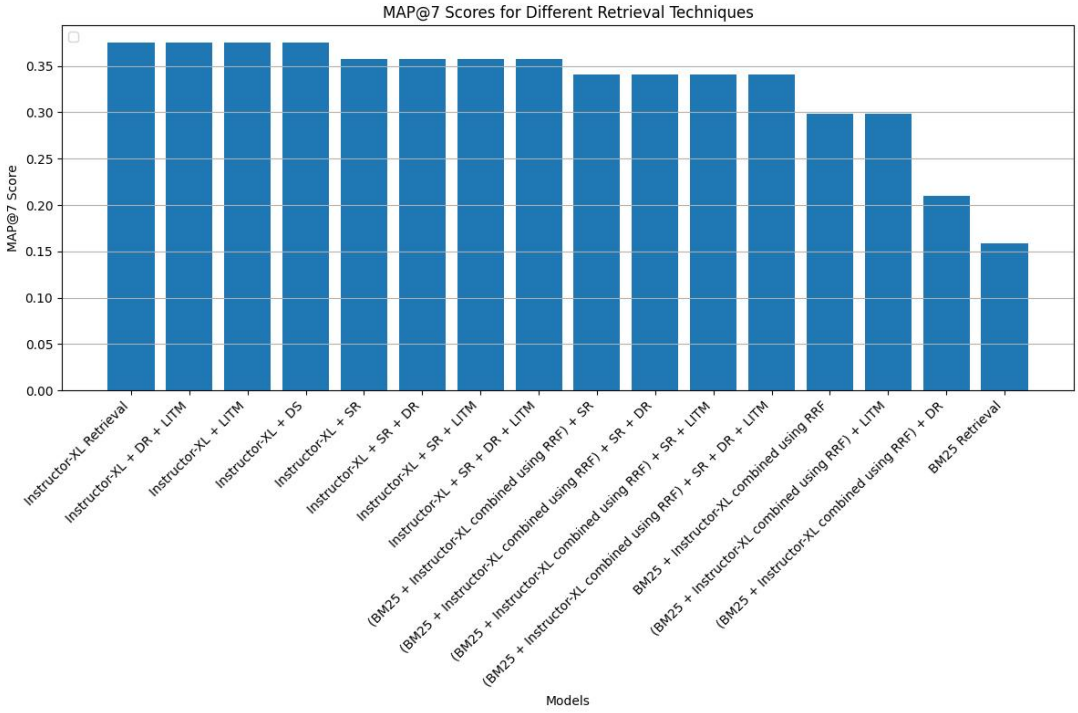

Figure 21: MAP score for the Top 7 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest MAP score of 0.3749.
- Adding the Diversity, Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker to the Dense Retrieval, resulted in a MAP score of 0.3571, which is lower than the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave an MAP score of 0.3571.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave an MAP score of 0.3571.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

10.4 MAP Score for the Top 10 Retrieved Results

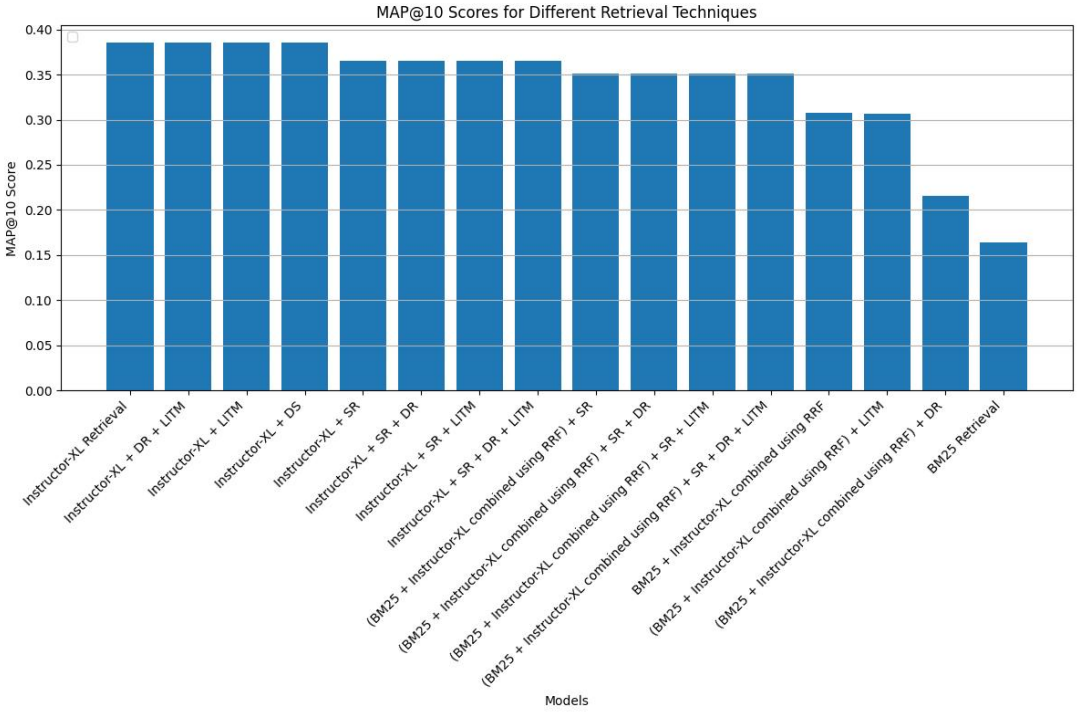

Figure 22: MAP Score for the Top 10 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest MAP score of 0.3851.
- Adding the Diversity, Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker to the Dense Retrieval, resulted in a MAP score of 0.3654, which is lower than the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave a MAP score of 0.3654.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave a MAP score of 0.3654.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

11 Recall Scores for the Top Retrieved Results using Instructor-XL

11.1 Recall Score for the Top 3 Retrieved Results

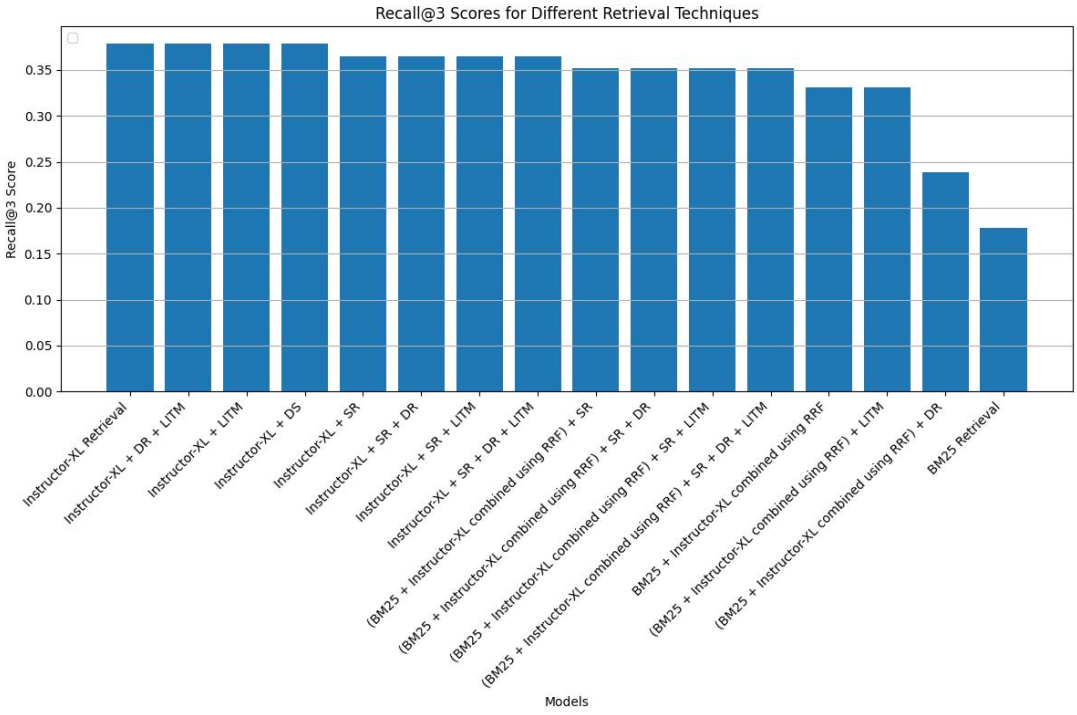

Figure 23: Recall Score for the Top 3 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest Recall score of 0.3783.
- Adding the Diversity, Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker to the Dense Retrieval, resulted in a Recall score of 0.3649, which is lower than the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave a Recall score of 0.3649.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave a Recall score of 0.3649.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

11.2 Recall Score for the Top 5 Retrieved Results

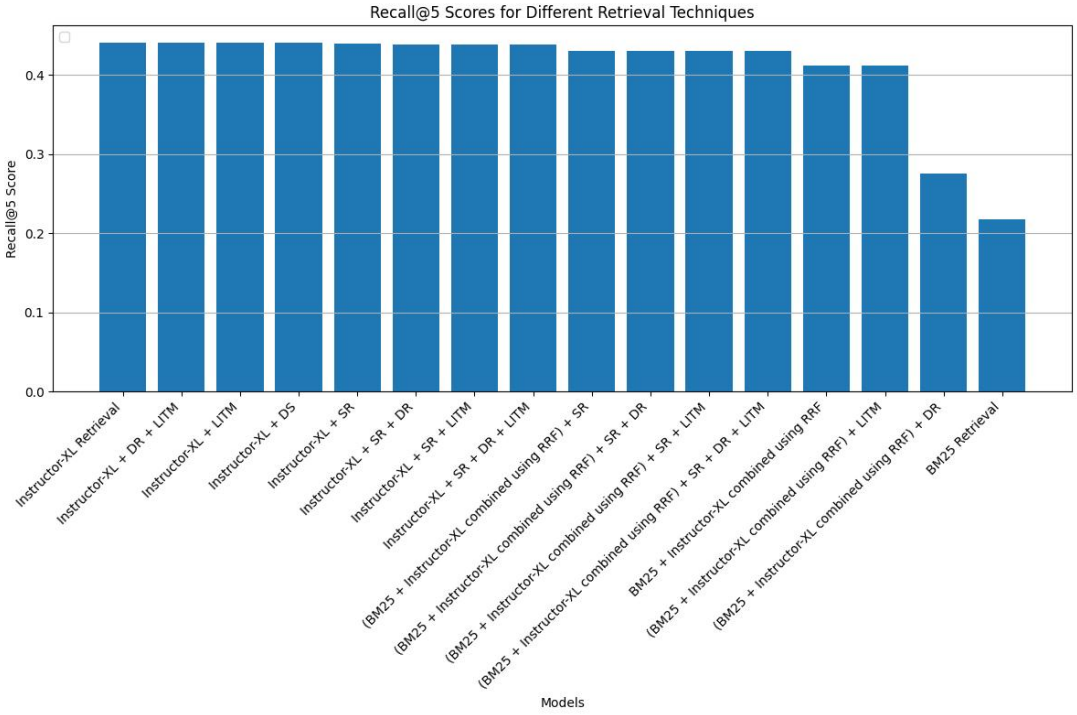

Figure 24: Recall Score for the Top 5 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives a Recall score of 0.4402.
- Adding the Diversity, Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker to the Dense Retrieval, resulted in a Recall score of 0.4387, which is lower than the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave a Recall score of 0.4387.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave a Recall score of 0.4387. The combination of adding the (Similarity + Diversity) and (Similarity + Lost In The Middle) gives the best retrieval.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

11.3 Recall Score for the Top 7 Retrieved Results

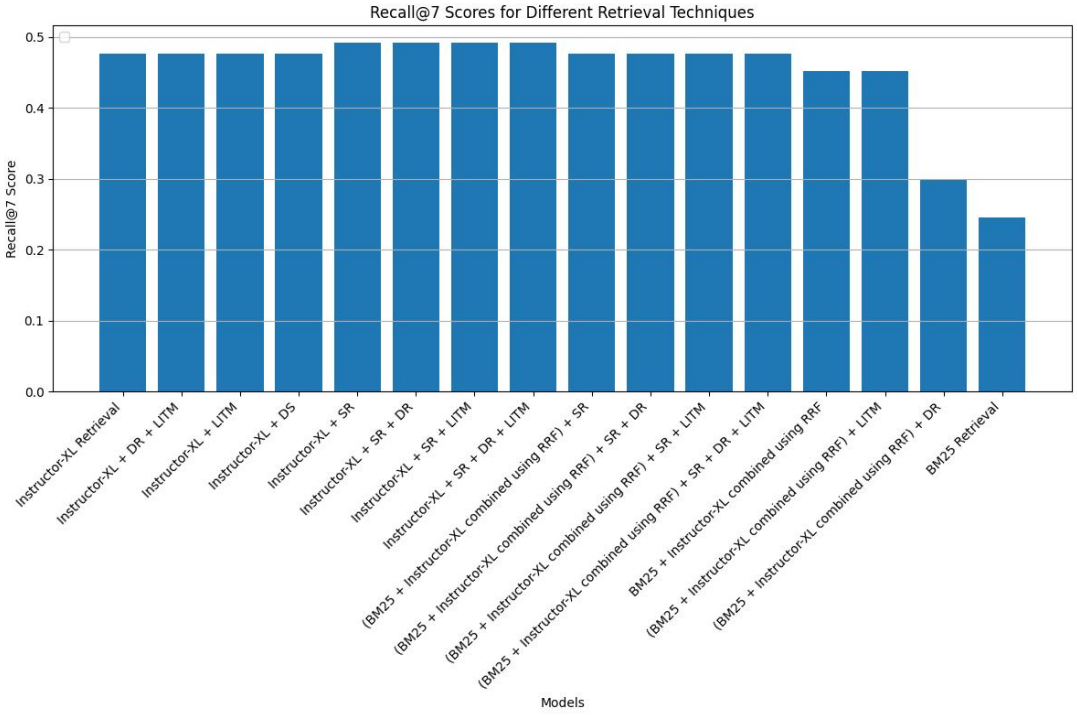

Figure 25: Recall Score for the Top 7 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives a Recall score of 0.4766.
- Adding the Diversity, Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker resulted in a lower score to that the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave a Recall score of 0.4915.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave a Recall score of 0.4915. The combination of adding the (Similarity + Diversity) and (Similarity + Lost In The Middle) gives the best retrieval.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

11.4 Recall Score for the Top 10 Retrieved Results

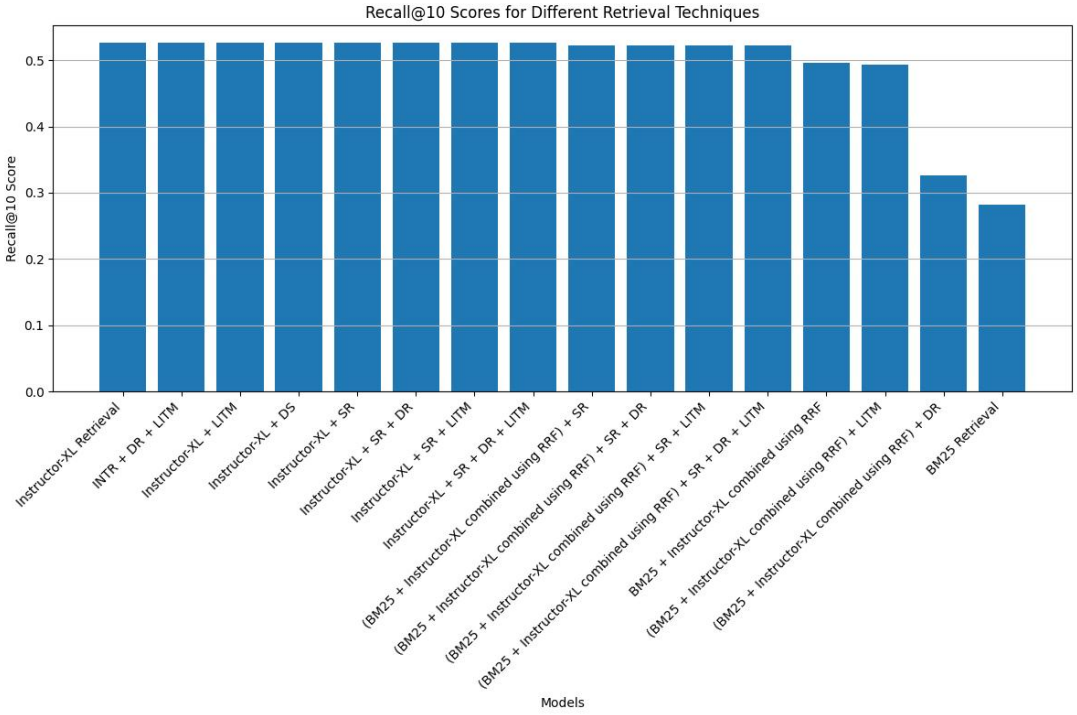

Figure 26: Recall Score For The Top 10 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives a Recall score of 0.526.
- Adding the Diversity, Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave an Recall score of 0.526.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave an Recall score of 0.526.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

12 Precision Score for the Top Retrieved Results using Instructor-XL

12.1 Precision Score for the Top 3 Retrieved Results

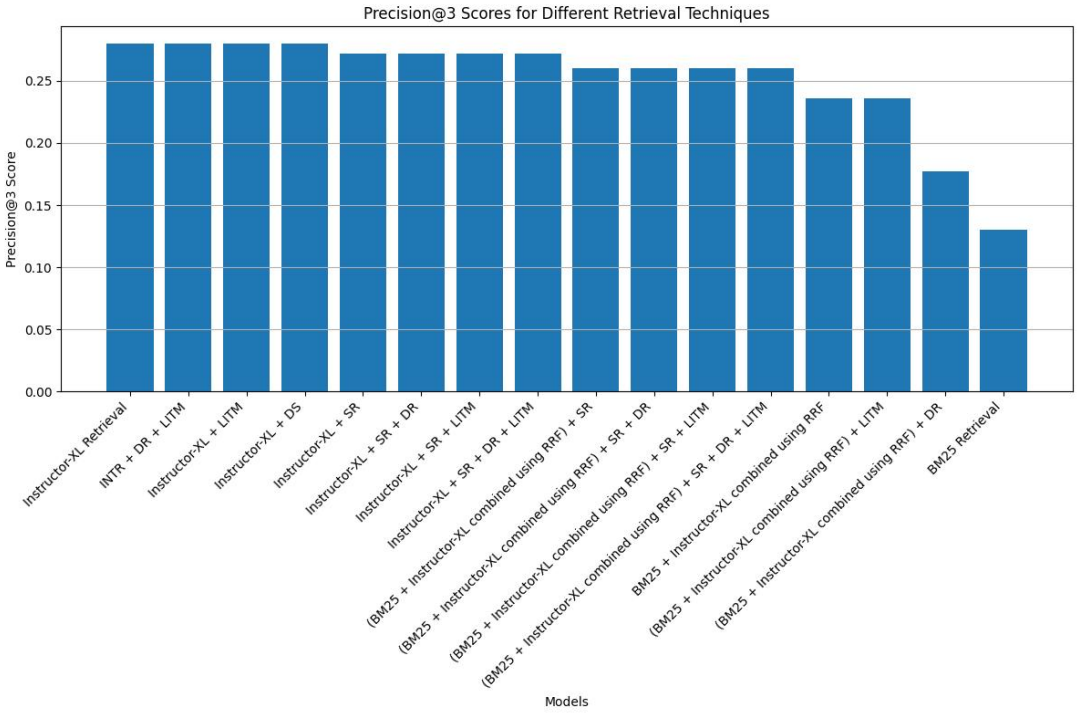

Figure 27: Precision Score for the Top 3 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest Precision score of 0.2798.
- Adding the Diversity, Similarity and Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave an Precision score of 0.2721.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave an Precision score of 0.2721.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

12.2 Precision Score for the Top 5 Retrieved Results

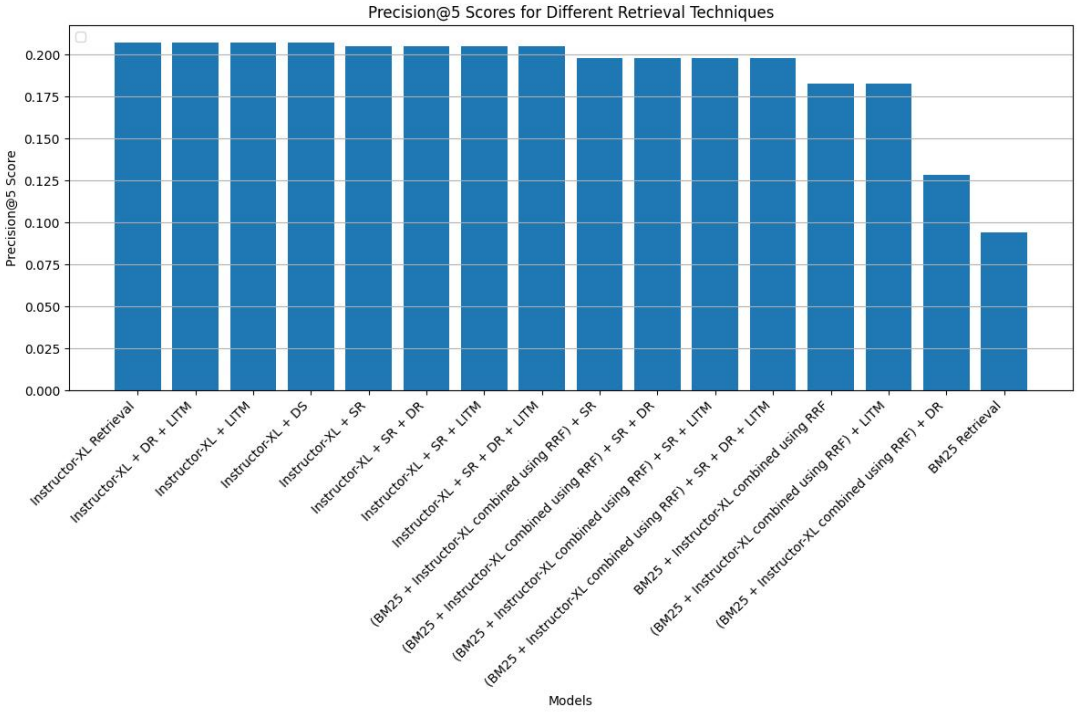

Figure 28: Precision Score for the Top 5 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest Precision score of 0.2071.
- Adding the Diversity, Similarity and Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave an Precision score of 0.2052.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave an Precision score of 0.2052.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

12.3 Precision Score for the Top 7 Retrieved Results

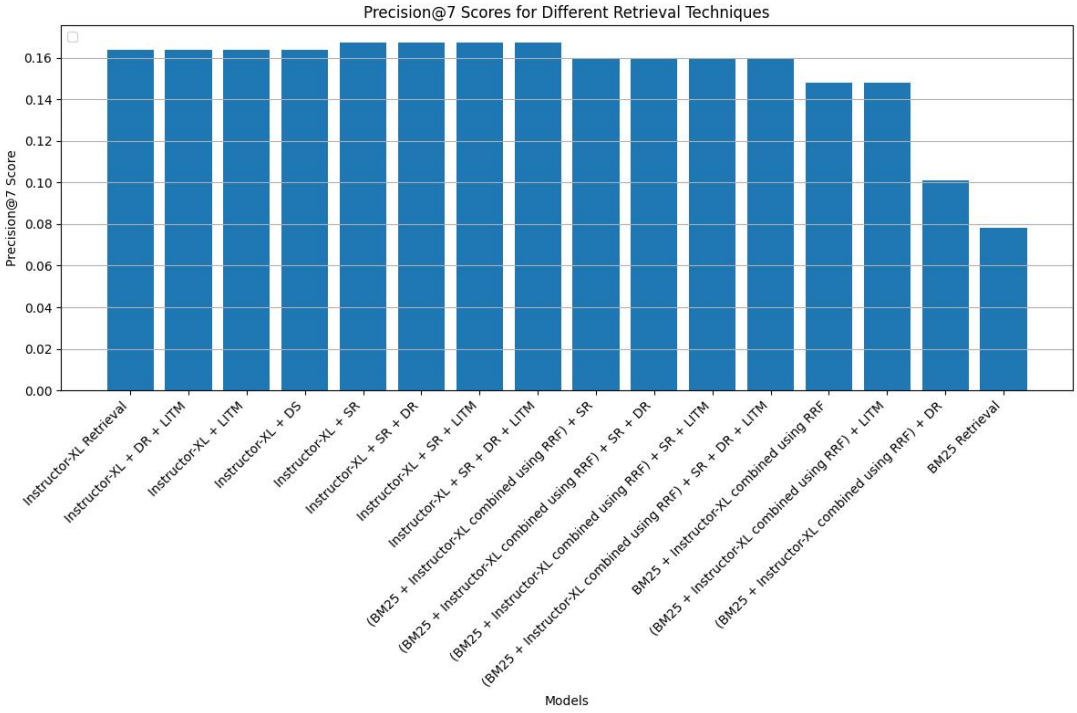

Figure 29: Precision Score for the Top 7 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest Precision score of 0.1638.
- Adding the Diversity, Similarity and Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave an Precision score of 0.1598.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave an Precision score of 0.1598.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

12.4 Precision Score for the Top 10 Retrieved Results

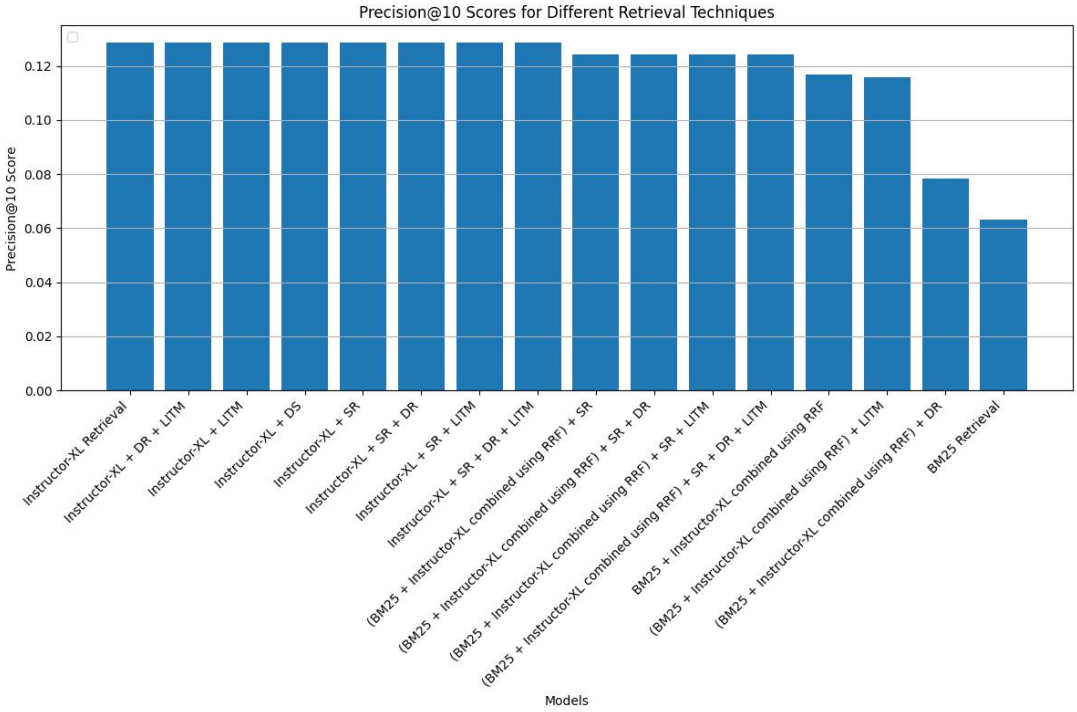

Figure 30: Precision Score for the Top 7 Retrieved Results

- The Dense Retrieval with the Instructor-XL gives the best retrieval, it has the highest Precision score of 0.1285.
- Adding the Diversity, Similarity and Lost in the Middle Rankers individually to the Dense Retrieval resulted in equivalent scores to that of the Dense Retrieval.
- Adding the Similarity Ranker and Diversity Ranker together to the Dense Retrieval gave an Precision score of 0.1285.
- Adding the Similarity and Lost In The Middle Ranker together to the Dense Retrieval gave an Precision score of 0.1285.
- All combinations of Hybrid Retrieval perform worse than the Dense Retrieval.

13 Summary of Results for Dense Retrieval with the Instructor-XL Embedding Model

We analysed the performance of adding rankers (Similarity, Lost in The Middle, Diversity) in a dense Retrieval pipeline using the Instructor-XL embedding model.

- The Instructor-XL gives the best NDCG score 0.425 with Dense Retrieval.
- The Instructor-XL with the Diversity Ranker, Lost in the Middle Ranker gives the same NDCG score with Dense Retrieval.
- The Instructor-XL with the Lost in the Middle Ranker gives the same NDCG score with with Dense Retrieval.
- The Instructor-XL + Diversity Ranker (ms-marco-MiniLM-L-12-v2) gives a lower NDCG score than the dense Retrieval with no ranker.
- The Instructor-XL + Similarity Ranker (bge-reranker-large) + Diversity Ranker (ms-marco-MiniLM-L-12-v2) gives a NDCG score of 0.4031 lower than that of NDCG with no rankers.

- The Instructor-XL + Similarity Ranker (bge-reranker-large) + Lost In the Middle Ranker gives a NDCG score of 0.4031 lower than that of NDCG with no rankers.
- The Instructor-XL + Similarity Ranker (bge-reranker-large) + Diversity Ranker (ms-marco-MiniLM-L-12-v2) + Lost In the Middle Ranker gives a NDCG score of 0.4031 lower than that of NDCG with no rankers.
- For the FIQA dataset adding rankers gives equivalent performance in the case of adding the Diversity, Lost in the middle, and Similarity rankers individually with no combinations.
- Whereas adding two or more rankers with different combinations gives a lower NDCG score for the Retrieval.

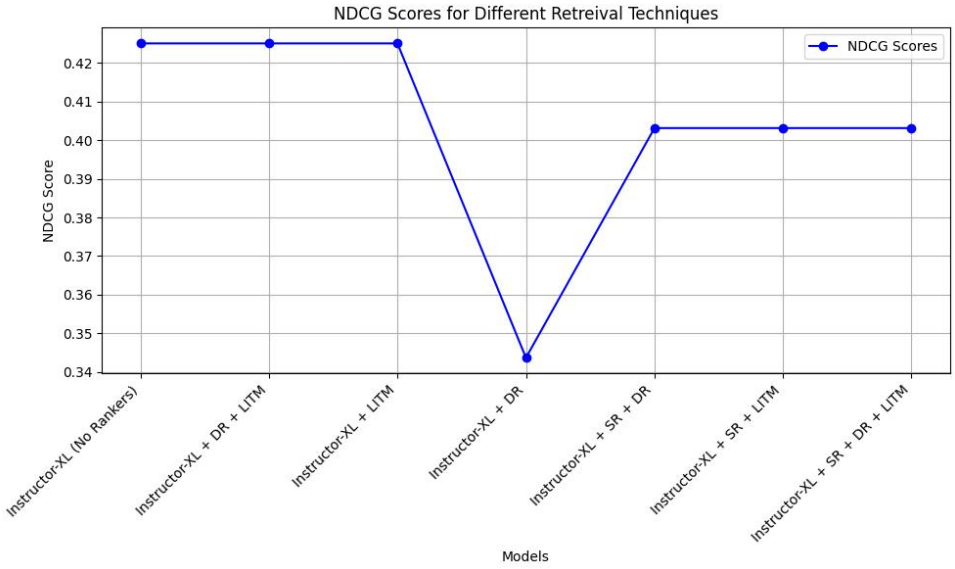

Figure 31: Summary of Results for Rankers in Dense Retrieval Pipelines

Name	NDCG@3	NDCG@5	NDCG@7	NDCG@10
Instructor-XL Retrieval	0.425	0.436	0.4456	0.4652
Instructor-XL + Diversity Ranker + Lost In the Middle Ranker	0.425	0.436	0.4456	0.4652
Instructor-XL + Lost In the Middle Ranker	0.425	0.436	0.4456	0.4625
Instructor-XL + Diversity Ranker	0.425	0.436	0.4456	0.4625
Instructor-XL + Similarity Ranker	0.4031	0.4203	0.4372	0.4491
Instructor-XL + Similarity Ranker + Diversity Ranker	0.4031	0.4203	0.4372	0.4491
Instructor-XL + Similarity Ranker + Lost In the Middle Ranker	0.4031	0.4203	0.4372	0.4491
Instructor-XL + Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker	0.4031	0.4203	0.4372	0.4491
(BM25 + Instructor-XL combined using RRF) + Similarity Ranker	0.3861	0.4057	0.4204	0.436
(BM25 + Instructor-XL combined using RRF) + Similarity Ranker + Diversity Ranker	0.3861	0.4057	0.4204	0.436
(BM25 + Instructor-XL combined using RRF) + Similarity Ranker + Lost In the Middle Ranker	0.3861	0.4057	0.4204	0.436
(BM25 + Instructor-XL combined using RRF) + Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker	0.3861	0.4057	0.4204	0.436
BM25 + Instructor-XL combined using RRF	0.3437	0.3672	0.3806	0.3962
(BM25 + Instructor-XL combined using RRF) + Lost In the Middle Ranker	0.3437	0.3672	0.3806	0.395
(BM25 + Instructor-XL combined using RRF) + Diversity Ranker	0.2626	0.2664	0.2724	0.2812
BM25 Retrieval	0.1905	0.1988	0.2089	0.2212

Table 2: NDCG for Different retrieval techniques

13.1 Summary Of Results for Rankers In A Hybrid Retrieval Pipeline With The Instructor-XL Embedding Model

- The (BM25 + Instructor-XL combined using RRF) + Similarity Ranker (bge-reranker-large) gives a NDCG score of 0.3861.
- (BM25 + Instructor-XL combined using RRF) + Similarity Ranker (bge-reranker-large) + Diversity Ranker (ms-marco-MiniLM-L-12-v2) gives a NDCG score of 0.3861.
- (BM25 + Instructor-XL combined using RRF) + Similarity Ranker (bge-reranker-large) + Lost In the Middle Ranker gives a NDCG score of 0.3861.

Name	MAP@3	MAP@5	MAP@7	MAP@10
Instructor-XL Retrieval	0.3362	0.3637	0.3749	0.3851
Instructor-XL + DR + LITM	0.3362	0.3637	0.3749	0.3851
Instructor-XL + LITM	0.3362	0.3637	0.3749	0.3851
Instructor-XL + DR	0.3362	0.3637	0.3749	0.3851
Instructor-XL + SR	0.3109	0.3422	0.3571	0.3654
Instructor-XL + SR + DR	0.3109	0.3422	0.3571	0.3654
Instructor-XL + SR + LITM	0.3109	0.3422	0.3571	0.3654
Instructor-XL + SR + DR + LITM	0.3109	0.3422	0.3571	0.3654
(BM25 + Instructor-XL combined using RRF) + SR	0.2978	0.3273	0.3409	0.3506
(BM25 + Instructor-XL combined using RRF) + SR + DR	0.2978	0.3273	0.3409	0.3506
(BM25 + Instructor-XL combined using RRF) + SR + LITM	0.2978	0.3273	0.3409	0.3506
(BM25 + Instructor-XL combined using RRF) + SR + DR + LITM	0.2978	0.3273	0.3409	0.3506
BM25 + Instructor-XL combined using RRF	0.256	0.2856	0.298	0.3073
(BM25 + Instructor-XL combined using RRF) + LITM	0.256	0.2856	0.298	0.30668
(BM25 + Instructor-XL combined using RRF) + DR	0.1914	0.2049	0.2103	0.2151
BM25 Retrieval	0.14	0.1511	0.1583	0.1636

Table 3: MAP Scores for Different Retrieval Techniques

Name	Recall@3	Recall@5	Recall@7	Recall@10
Instructor-XL Retrieval	0.3783	0.4402	0.4766	0.526
Instructor-XL + DR + LITM	0.3783	0.4402	0.4766	0.526
Instructor-XL + LITM	0.3783	0.4402	0.4766	0.526
Instructor-XL + DR	0.3783	0.4402	0.4766	0.526
Instructor-XL + SR	0.3649	0.4397	0.4915	0.526
Instructor-XL + SR + DR	0.3649	0.4387	0.4915	0.526
Instructor-XL + SR + LITM	0.3649	0.4387	0.4915	0.526
Instructor-XL + SR + DR + LITM	0.3649	0.4387	0.4915	0.526
(BM25 + Instructor-XL combined using RRF) + SR	0.3522	0.4301	0.4769	0.5226
(BM25 + Instructor-XL combined using RRF) + SR + DR	0.3522	0.4301	0.4769	0.5226
(BM25 + Instructor-XL combined using RRF) + SR + LITM	0.3522	0.4301	0.4769	0.5226
(BM25 + Instructor-XL combined using RRF) + SR + DR + LITM	0.3522	0.4301	0.4769	0.5226
BM25 + Instructor-XL combined using RRF	0.3311	0.4115	0.4519	0.4961
(BM25 + Instructor-XL combined using RRF) + LITM	0.3311	0.4115	0.4519	0.4934
(BM25 + Instructor-XL combined using RRF) + DR	0.2383	0.2757	0.2998	0.3262
BM25 Retrieval	0.1779	0.2173	0.246	0.2818

Table 4: Recall Scores for Different Retrieval Techniques

- (BM25 + Instructor-XL combined using RRF) + Similarity Ranker (bge-reranker-large) + Diversity Ranker (ms-marco-MiniLM-L-12-v2) + Lost In the Middle Ranker gives a NDCG score of 0.3861.
- The BM25 + Instructor-XL combined using RRF gives a NDCG score of 0.3437
- (BM25 + Instructor-XL combined using RRF) + Lost In the Middle Ranker gives a NDCG score of 0.3437.
- Adding two rankers in the combination of Similarity Ranker and Diversity Ranker gives better performance.
- Adding two rankers in the combination of Similarity Ranker and Lost in the Middle gives better performance.
- Adding all the three rankers also gives the equivalent performance as the combination of two rankers.
- Whereas adding single rankers sequentially like Lost in the middle and Diversity ranker does not give an improvement in performance.

Name	P@3	P@5	P@7	P@10
Instructor-XL Retrieval	0.2798	0.2071	0.1638	0.1285
Instructor-XL + DR + LITM	0.2798	0.2071	0.1638	0.1285
Instructor-XL + LITM	0.2798	0.2071	0.1638	0.1285
Instructor-XL + DR	0.2798	0.2071	0.1638	0.1285
Instructor-XL + SR	0.2721	0.2052	0.1671	0.1285
Instructor-XL + SR + DR	0.2721	0.2052	0.1671	0.1285
Instructor-XL + SR + LITM	0.2721	0.2052	0.1671	0.1285
Instructor-XL + SR + DR + LITM	0.2721	0.2052	0.1671	0.1285
(BM25 + Instructor-XL combined using RRF) + SR	0.2603	0.1978	0.1598	0.1241
(BM25 + Instructor-XL combined using RRF) + SR + DR	0.2603	0.1978	0.1598	0.1241
(BM25 + Instructor-XL combined using RRF) + SR + LITM	0.2603	0.1978	0.1598	0.1241
(BM25 + Instructor-XL combined using RRF) + SR + DR + LITM	0.2603	0.1978	0.1598	0.1241
BM25 + Instructor-XL combined using RRF	0.2361	0.1827	0.1479	0.1167
(BM25 + Instructor-XL combined using RRF) + LITM	0.2361	0.1827	0.1479	0.1157
(BM25 + Instructor-XL combined using RRF) + DR	0.1775	0.1284	0.101	0.0785
BM25 Retrieval	0.1301	0.0941	0.0783	0.063

Table 5: Precision Scores for Different Retrieval Techniques

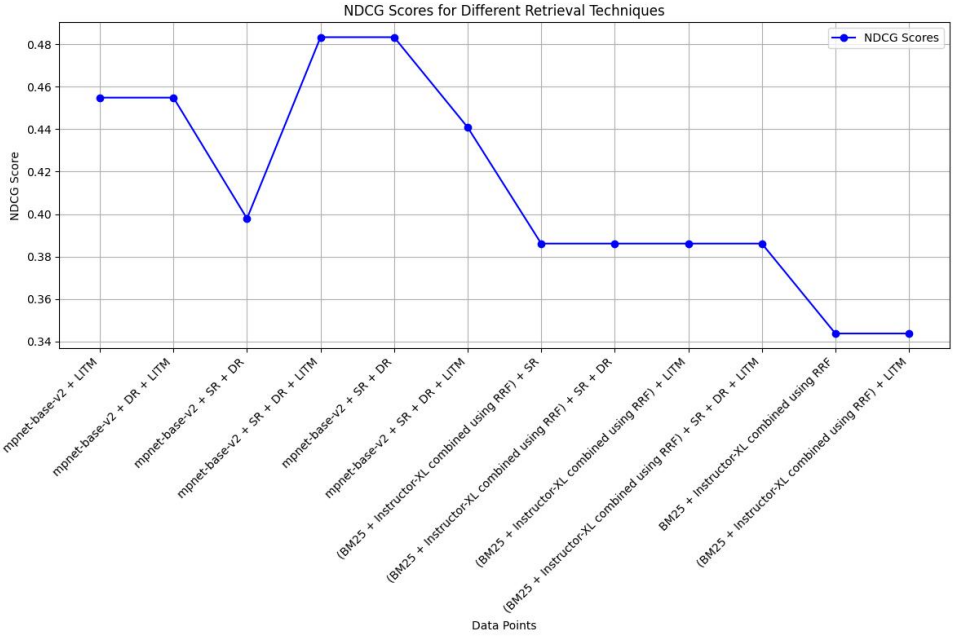

Figure 32: Comparing Performance of Hybrid Retrieval for Top 3 Retrieved Results

14 Adding Rankers to Improve Performance for Dense Retrieval with the mpnet-v2 Embedding Model

We analyse the performance of adding rankers (Similarity, Lost in The Middle, Diversity) and RRF in a dense Retrieval pipeline using the all-mpnet-base-v2 embedding model. First each ranker (Diversity, Lost in The Middle and Similarity) is added individually and the evaluated on the NDCG, MAP, Recall and Precision scores. Then we add the pairs of rankers first the Similarity Ranker with the Diversity Ranker and then the Similarity Ranker with the Lost in the Middle Ranker. Finally all the three rankers are the Similarity, Diversity and Lost in The Middle ranker are added and evaluated. The Lost in The Middle Ranker is added at at the end since orders the most relevant documents for Retrieval.

14.1 Dense + Diversity Ranker

Figure 33: Dense + Diversity Ranker

Adding a Diversity Ranker results in a NDCG score of 0.4397.

14.2 Dense + Lost in The Middle Ranker

Figure 34: Retrieval Pipeline for Dense + Lost in The Middle Ranker

Adding a Lost in The Middle Ranker results in a NDCG score of 0.4549 which is the highest and gives the best performance.

14.3 Dense + Similarity Ranker

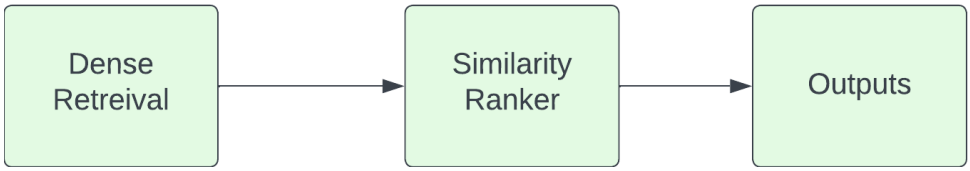

Figure 35: Dense + Similarity Ranker

Adding a Similarity Ranker results in a NDCG score of 0.4397 which is lower than that of adding a and Lost in The Middle Ranker.

14.4 Dense + Similarity Ranker + Diversity Ranker

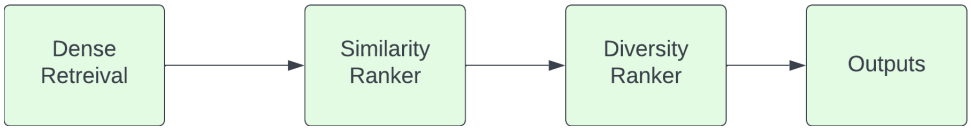

Figure 36: Dense + Similarity Ranker + Diversity Ranker

Adding a Similarity and Diversity ranker results in a NDCG score of 0.3978. It yields a score lower than that of adding a Diversity or Lost in The Middle Ranker individually to the pipeline.

14.5 Dense + Similarity Ranker + Lost in The Middle Ranker

Figure 37: Dense + Similarity Ranker + Lost in The Middle Ranker

Adding a Similarity and Lost in The Middle ranker results in a NDCG score of 0.4391. It performs lower than adding a Diversity or Lost in The Middle Ranker individually to the pipeline.

14.6 Dense + Similarity Ranker + Diversity Ranker + Lost in The Middle Ranker

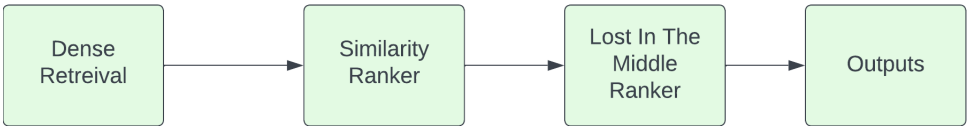

Figure 38: Summary of Techniques

Adding a Similarity, Diversity and Lost in The Middle ranker results in a NDCG score of 0.3978. It performs lower than adding a Diversity or Lost in The Middle Ranker individually to the pipeline. Adding all the three rankers to the pipeline does not result in improvement in performance.

Name	NDCG@3	NDCG@5	NDCG@7	NDCG@10
all-mpnet-base-v2 + SR	0.4397	0.4585	0.4754	0.4883
all-mpnet-base-v2 + LITM	0.4549	0.4715	0.4834	0.4997
all-mpnet-base-v2 + DR	0.4549	0.4715	0.4834	0.4997
all-mpnet-base-v2 + SR + DR	0.3978	0.4233	0.4409	0.4601
all-mpnet-base-v2 + SR + LITM	0.4397	0.4585	0.4754	0.4883
all-mpnet-base-v2 + DR + LITM	0.4549	0.4715	0.4834	0.4997
all-mpnet-base-v2 + SR + DR + LITM	0.3978	0.4233	0.4409	0.4601

Table 6: NDCG Scores for different Retrieval techniques

Name	MAP@3	MAP@5	MAP@7	MAP@10
all-mpnet-base-v2 + SR	0.3381	0.3724	0.3884	0.3968
all-mpnet-base-v2 + LITM	0.3574	0.3872	0.4003	0.4113
all-mpnet-base-v2 + DR	0.3574	0.3872	0.4003	0.4113
all-mpnet-base-v2 + SR + DR	0.3027	0.3387	0.3538	0.3652
all-mpnet-base-v2 + SR + LITM	0.3381	0.3724	0.3884	0.3968
all-mpnet-base-v2 + DR + LITM	0.3574	0.3872	0.4003	0.4113
all-mpnet-base-v2 + SR + DR + LITM	0.3027	0.3387	0.3538	0.3652

Table 7: MAP Scores for different Retrieval techniques

15 Summary of Results for Rankers in Dense Retrieval Pipelines

- The all-mpnet-base-v2 embedding model + Lost In the Middle Ranker and all-mpnet-base-v2 embedding model + Diversity Ranker give the best performance with an NDCG score of 0.4549.
- The all-mpnet-base-v2 embedding model + Diversity Ranker (ms-marco-MiniLM-L-12-v2) + Lost In the Middle Ranker gives an NDCG score of 0.4549.

Name	Recall@3	Recall@5	Recall@7	Recall@10
all-mpnet-base-v2 + SR	0.4054	0.4871	0.5406	0.5814
all-mpnet-base-v2 + LITM	0.4163	0.4925	0.533	0.5814
all-mpnet-base-v2 + DR	0.4163	0.4925	0.533	0.5814
all-mpnet-base-v2 + SR + DR	0.366	0.4598	0.5114	0.566
all-mpnet-base-v2 + SR + LITM	0.4054	0.4871	0.5406	0.5814
all-mpnet-base-v2 + DR + LITM	0.4163	0.4925	0.533	0.5814
all-mpnet-base-v2 + SR + DR + LITM	0.366	0.4598	0.5114	0.5666

Table 8: Recall Scores for different Retrieval techniques

Name	Precision@3	Precision@5	Precision@7	Precision@10
all-mpnet-base-v2 + SR	0.3019	0.2265	0.183	0.139
all-mpnet-base-v2 + LITM	0.3066	0.2262	0.1797	0.139
all-mpnet-base-v2 + DR	0.3066	0.2262	0.1797	0.139
all-mpnet-base-v2 + SR + DR	0.2711	0.2089	0.172	0.139
all-mpnet-base-v2 + SR + LITM	0.3019	0.2265	0.183	0.139
all-mpnet-base-v2 + DR + LITM	0.3066	0.2262	0.1797	0.139
all-mpnet-base-v2 + SR + DR + LITM	0.2711	0.2089	0.172	0.1361

Table 9: Precision Scores for different Retrieval techniques

- The all-mpnet-base-v2 + Similarity Ranker (bge-reranker-large) + Diversity Ranker (ms-marco-MiniLM-L-12-v2) gives an NDCG score of 0.3979.
- The all-mpnet-base-v2 + Similarity Ranker (bge-reranker-large) + Diversity Ranker (ms-marco-MiniLM-L-12-v2) + Lost In the Middle Ranker gives an NDCG score of 0.4834.
- The all-mpnet-base-v2 + Similarity Ranker (bge-reranker-large) + Diversity Ranker (ms-marco-MiniLM-L-12-v2) gives an NDCG score of 0.4834.
- The all-mpnet-base-v2 + Similarity Ranker (bge-reranker-large) + Diversity Ranker (ms-marco-MiniLM-L-12-v2) + Lost In the Middle Ranker gives an NDCG score of 0.4409.
- Adding the pair of rankers that is Lost in the Middle ranker and Diversity Ranker give the best performance.
- The combination of rankers (Diversity + Similarity + Lost In the Middle) and (Diversity + Similarity) gives similar performance. Adding all the rankers together gives the lowest performance.

16 NDCG Scores for the Top Retrieved Results for all-mpnet-base-v2

16.1 NDCG Score for the Top 3 Retrieved Results

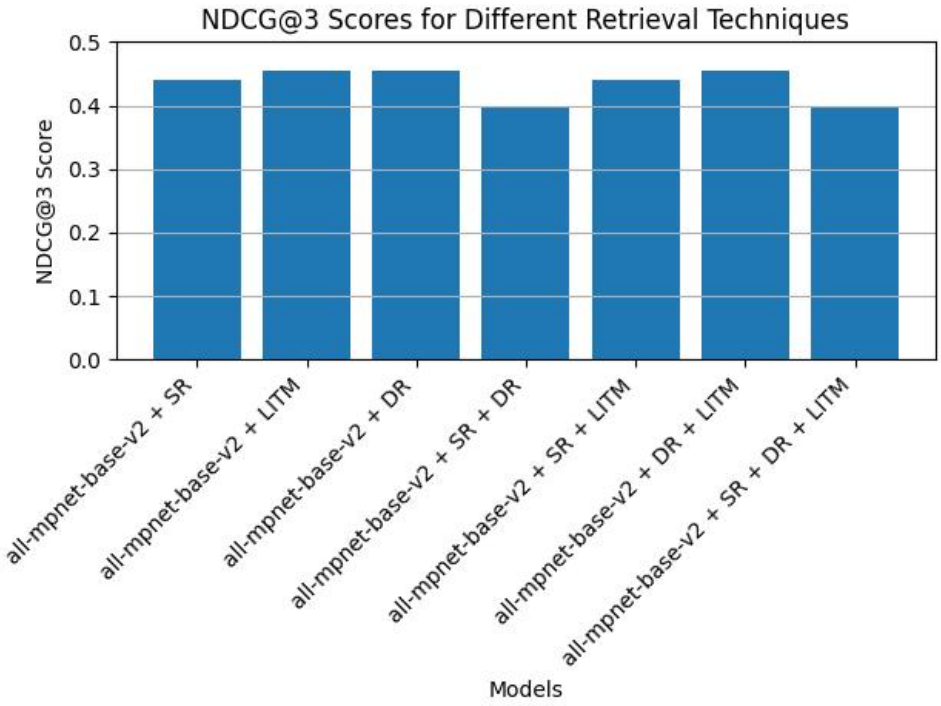

Figure 39: NDCG Score for the Top 3 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an NDCG score of 0.4397.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an NDCG score of 0.4549.
- Adding the rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance in Dense Retrieval, they give NDCG scores of 0.3978 and 0.4379 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an NDCG score of 0.3978.

16.2 NDCG Score for the Top 5 Retrieved Results

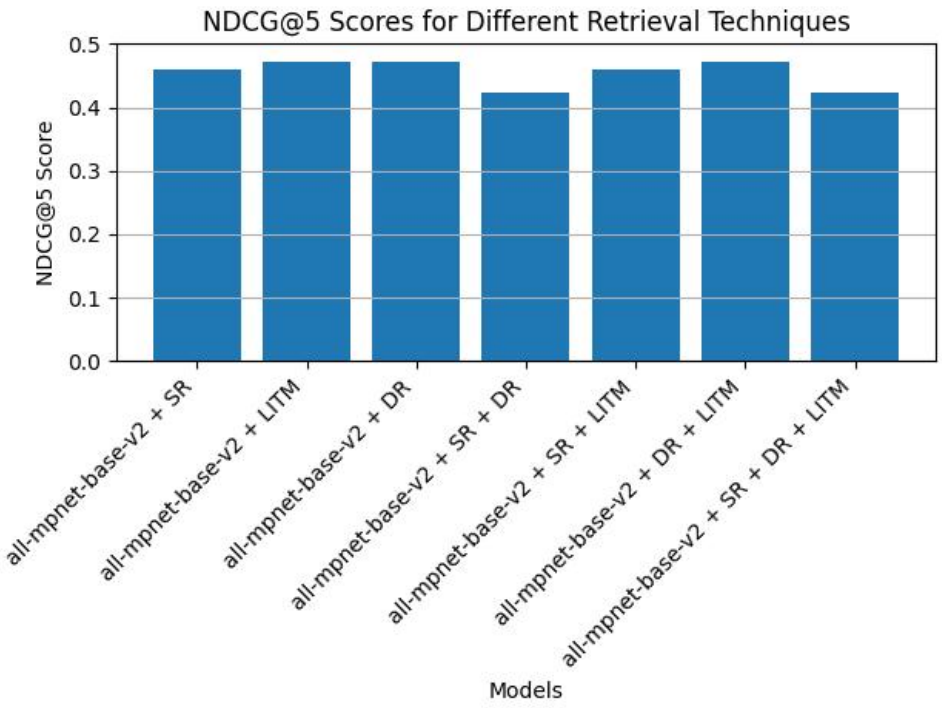

Figure 40: NDCG Score for the Top 5 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an NDCG score of 0.4585.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an NDCG score of 0.4715.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give NDCG scores of 0.4233 and 0.4585 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an NDCG score of 0.4233.

16.3 NDCG Score for the Top 7 Retrieved Results

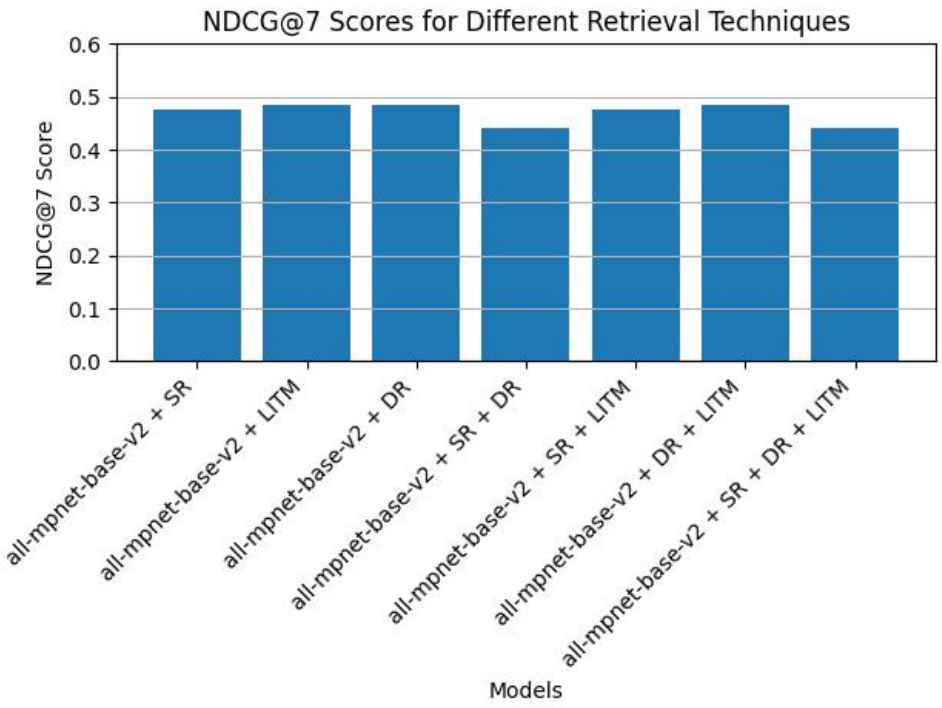

Figure 41: NDCG Score for the Top 7 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an NDCG score of 0.4754.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an NDCG score of 0.4834.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give NDCG scores of 0.4834 and 0.4409 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an NDCG score of 0.4409.

16.4 NDCG Score for the Top 10 Retrieved Results

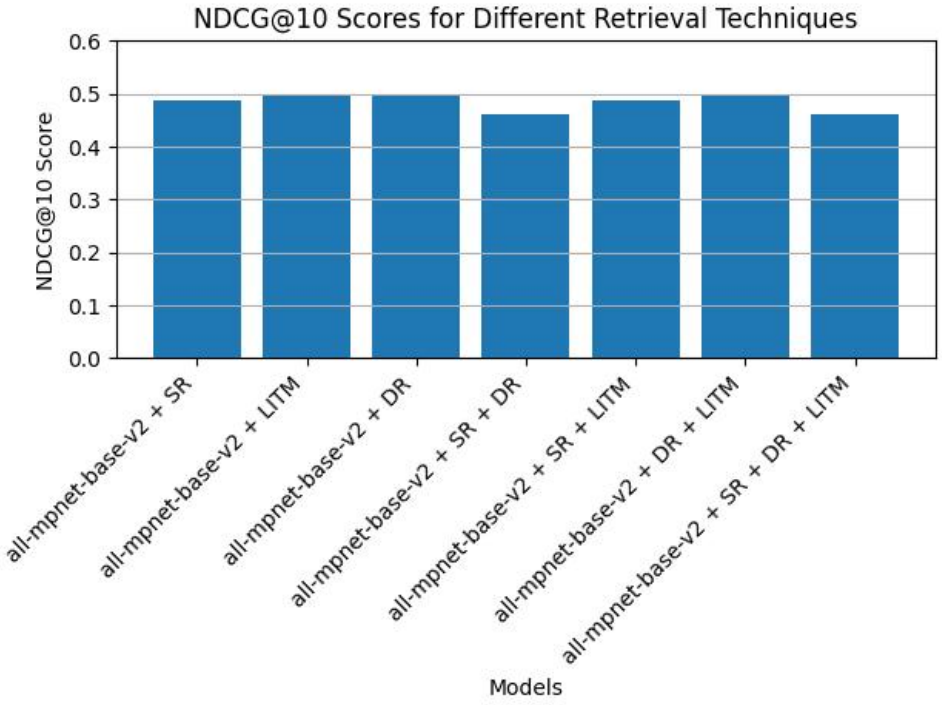

Figure 42: NDCG Score for the Top 10 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an NDCG score of 0.4883.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an NDCG score of 0.4997.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give NDCG scores of 0.4601 and 0.4601 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an NDCG score of 0.4601.

17 MAP Scores for the Top Retrieved Results for all-mpnet-base-v2

17.1 MAP Score for the Top 3 Retrieved Results

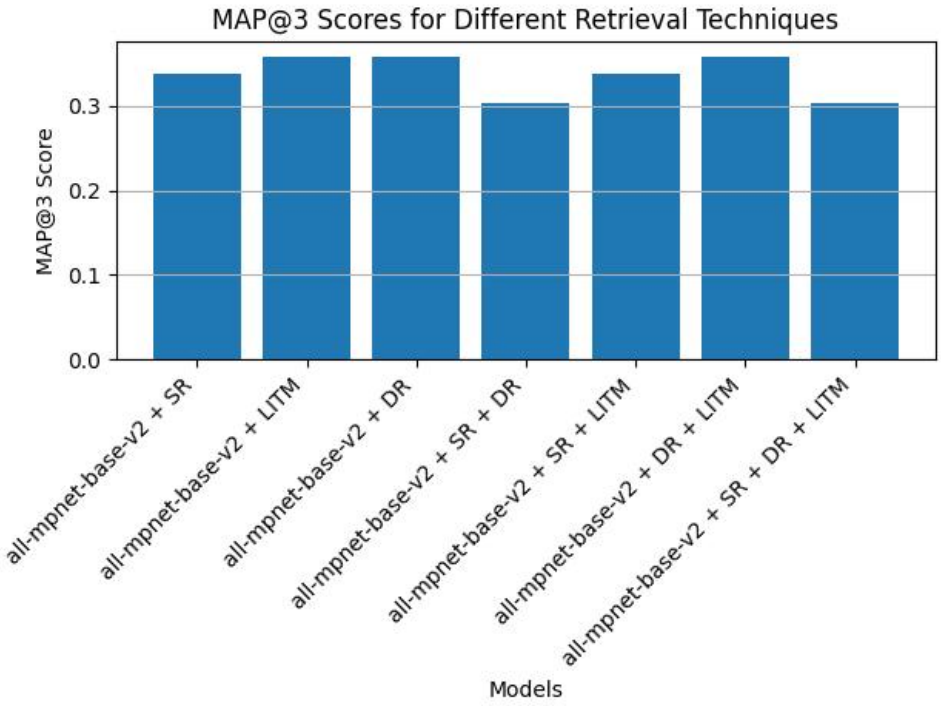

Figure 43: MAP Score for the Top 3 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an MAP score of 0.3381.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an MAP score of 0.3574.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give MAP scores of 0.3027 and 0.3381 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an Recall score of 0.3027.

17.2 MAP Score for the Top 5 Retrieved Results

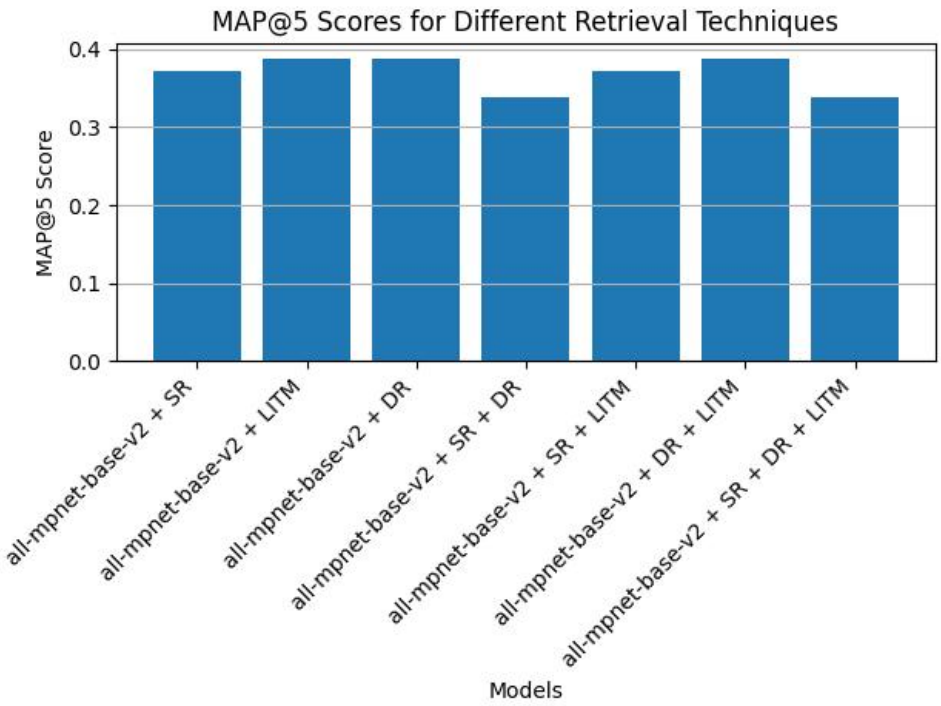

Figure 44: MAP Score for the Top 5 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an MAP score of 0.3724.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an MAP score of 0.3872.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give MAP scores of 0.3387 and 0.3724 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an MAP score of 0.3387.

17.3 MAP Score for the Top 7 Retrieved Results

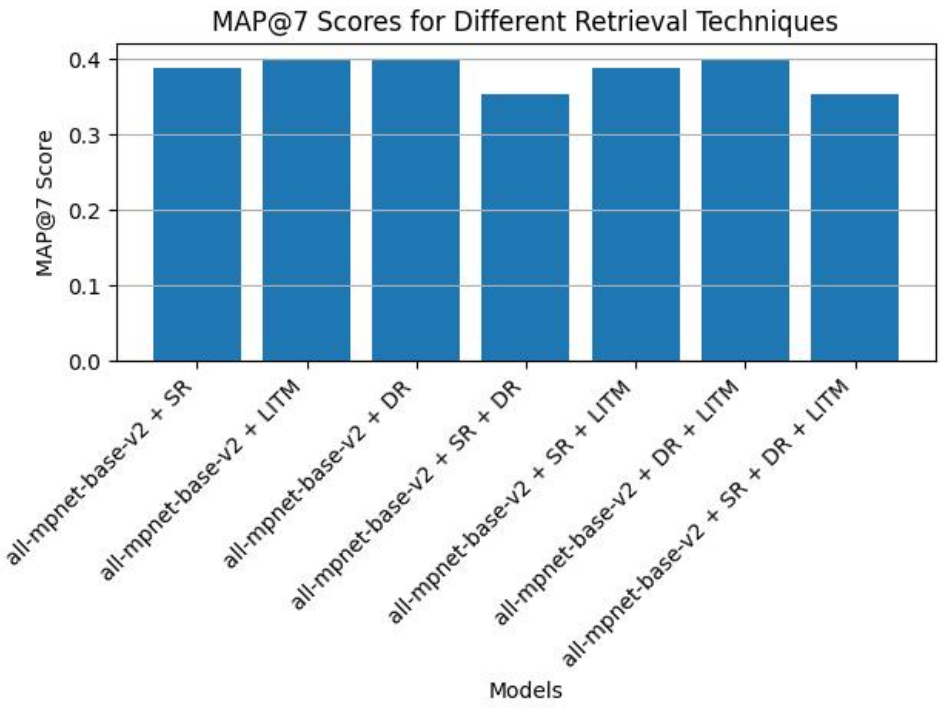

Figure 45: MAP Score for the Top 7 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an MAP score of 0.3884.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an MAP score of 0.4003.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give MAP scores of 0.4003 and 0.3538 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an MAP score of 0.3538.

17.4 MAP Score for the Top 10 Retrieved Results

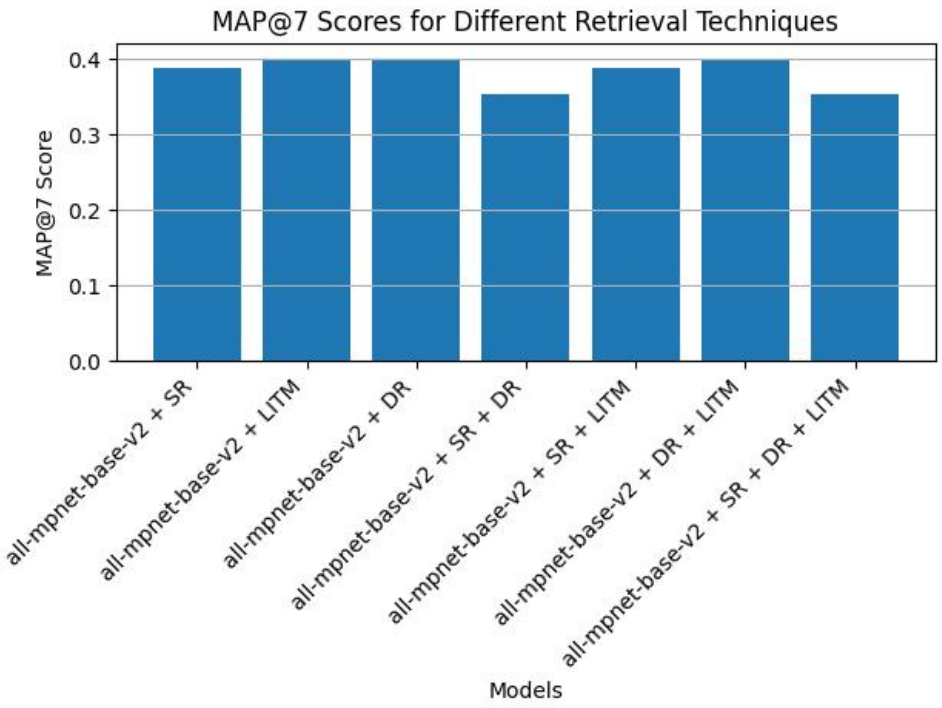

Figure 46: MAP Score for the Top 10 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an MAP score of 0.3968.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an MAP score of 0.4113.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give MAP scores of 0.3652 and 0.3968 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an MAP score of 0.3652.

18 Recall Scores for the Top Retrieved Results for all-mpnet-base-v2

18.1 Recall Score for the Top 3 Retrieved Results

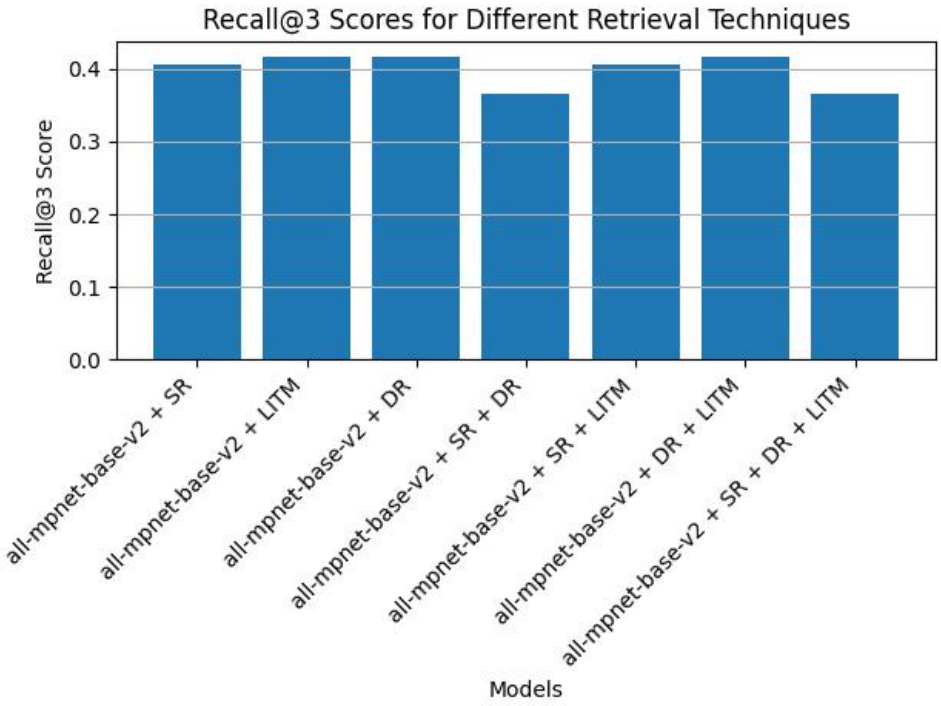

Figure 47: Recall Score for the Top 3 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an Recall score of 0.4054.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an Recall score of 0.4163.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give Recall scores of 0.366 and 0.4054 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an Recall score of 0.366.

18.2 Recall Score for the Top 5 Retrieved Results for all-mpnet-base-v2

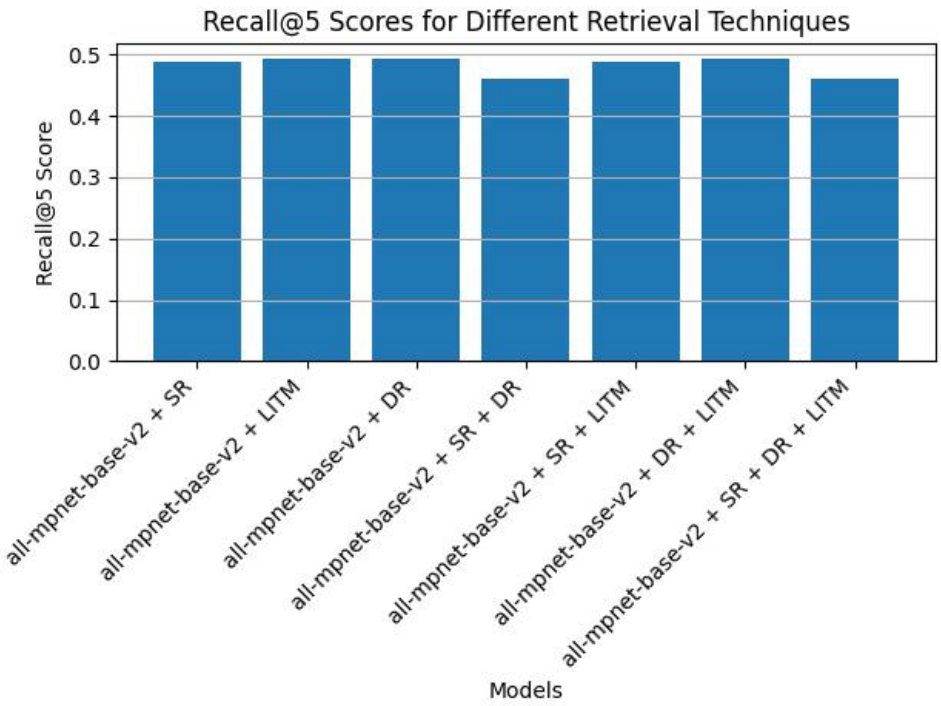

Figure 48: Recall Score for the Top 5 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an Recall score of 0.4871.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an Recall score of 0.4925.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give Recall scores of 0.4598 and 0.4871 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an Recall score of 0.4598.

18.3 Recall Score for the Top 7 Retrieved Results

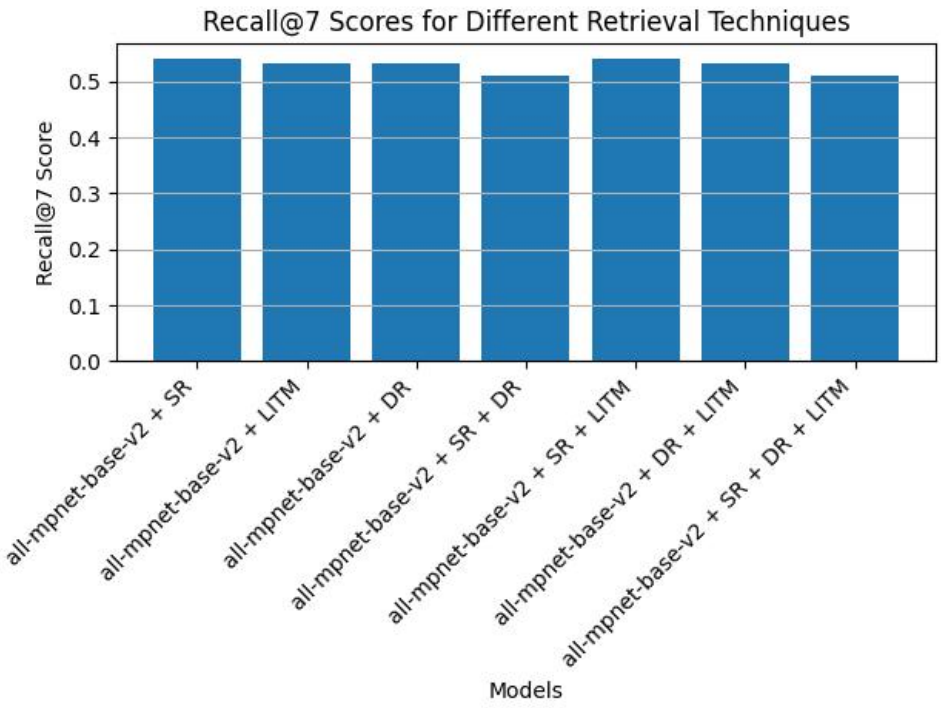

Figure 49: Recall Score for the Top 7 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an MAP score of 0.5406.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an MAP score of 0.533.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give MAP scores of 0.5114 and 0.5406 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an MAP score of 0.5114.

18.4 Recall Score for the Top 10 Retrieved Results

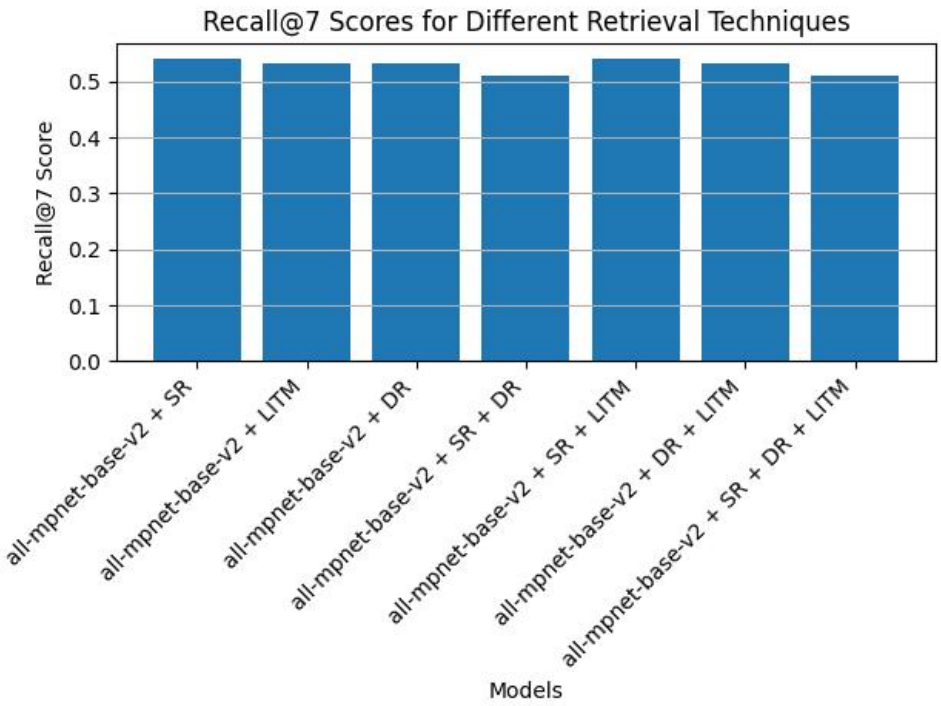

Figure 50: Recall Score for the Top 10 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an Recall score of 0.5814.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an Recall score of 0.5814.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give Recall scores of 0.566 and 0.5814 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an Recall score of 0.5666.

19 Precision Scores for the Top Retrieved Results for all-mpnet-base-v2

19.1 Precision Score for the Top 3 Retrieved Results

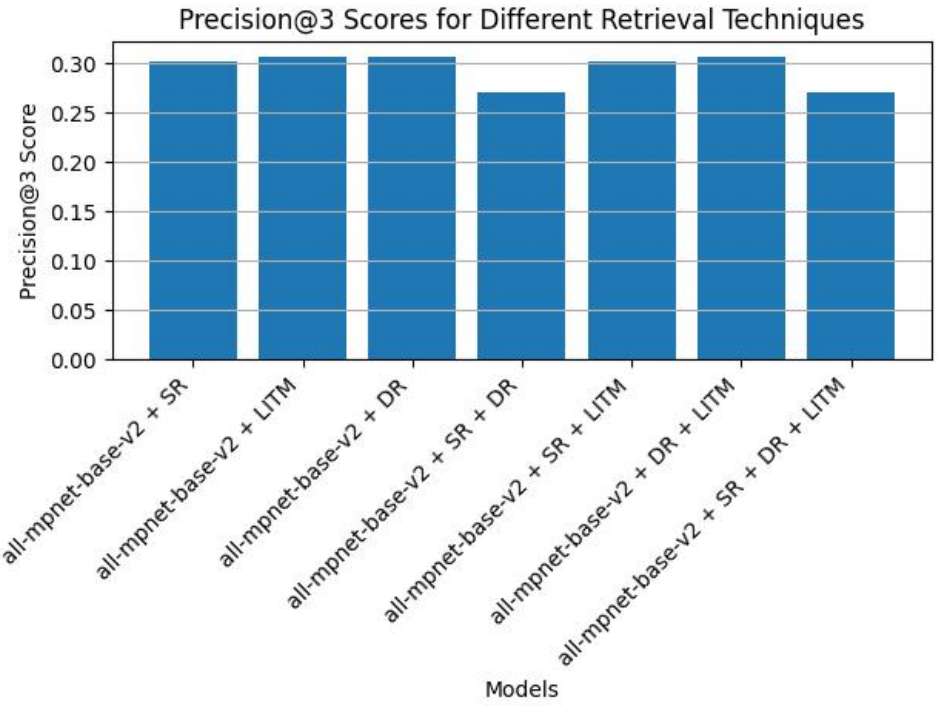

Figure 51: Precision Score for the Top 3 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an Recall score of 0.3019.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an Recall score of 0.3066.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give Recall scores of 0.2711 and 0.3019 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an Recall score of 0.2711.

19.2 Precision Score for the Top 5 Retrieved Results

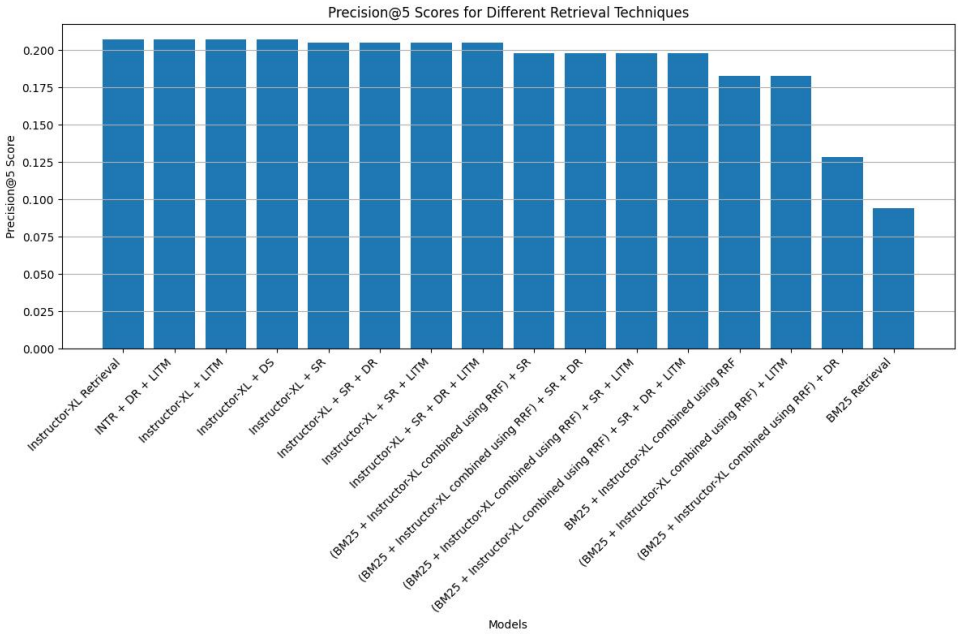

Figure 52: Precision Score for the Top 5 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give an an Recall score of 0.2265.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives an Recall score of 0.2265.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they give Recall scores of 0.2089 and 0.2265 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives an Recall score of 0.2089.

19.3 Precision Score for the Top 7 Retrieved Results

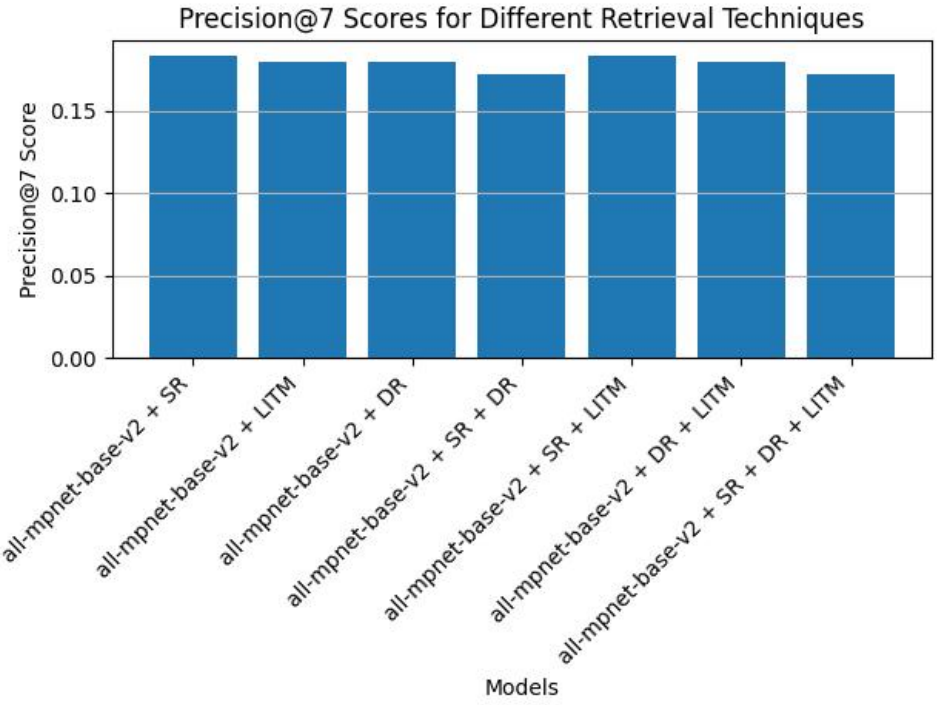

Figure 53: Precision Score for the Top 7 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give a Precision score of 0.183.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives a Precision score of 0.1797.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they gives a Precision scores of 0.172 and 0.183 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives a Precision score of 0.172.

19.4 Precision Score for the Top 10 Retrieved Results

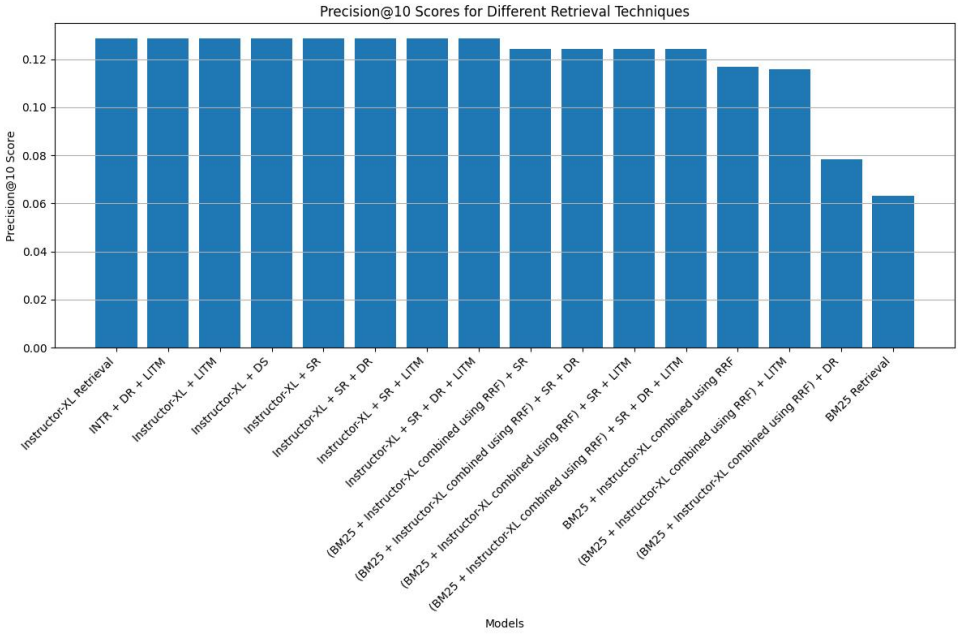

Figure 54: Precision Score for the Top 10 Retrieved Results

- Adding a ranker individually that is the Lost in the Middle and Diversity yield the best performance. They give a Precision score of 0.139.
- Adding the Similarity Ranker + Lost In the Middle Ranker with Dense retrieval gives similar performance to adding the rankers individually. This gives a Precision score of 0.139.
- Whereas adding pairs of rankers (Similarity Ranker + Diversity Ranker) and (Diversity Ranker + Lost In the Middle Ranker) give lower performance, they gives a Precision scores of 0.139 and 0.139 respectively.
- Also adding all the three rankers sequentially (Similarity Ranker + Diversity Ranker + Lost In the Middle Ranker) does not give improvement in performance it gives a Precision score of 0.1361.

20 Results and Conclusions

In our study on the FIQA datasaet, adding rankers individually in a Retrieval pipeline yielded the best results. Adding pair of rankers (2 at a time) in a Retrieval pipeline may result in lower or at par performance compared to adding these rankers sequentially (one at a time) in a pipeline. Adding all the three rankers in a pipeline does not result in improvement in performance it gives lower or at par results compared to adding the rankers sequentially.

- Diversity Ranker ensures that the generated answer is based on diverse documents. It uses a sentence transformer model to calculate the semantic representation (embedding) for each document. Then, it ranks the documents so that each subsequent document is the least similar to the ones it already selected. This results in a diverse set of documents. You can use it in combination with other rankers. If you do so, place it after the similarity ranker, like Sentence Transformers Ranker, but before the Lost In The Middle Ranker. Such setup is typical for the long form question answering task.
- Diversity Ranker ensures that the LLM's context window is filled with diverse, non-repetitive documents, providing a broader range of paragraphs for the LLM to synthesize the answer from. At the same time, the Lost In The Middle Ranker optimizes the placement of the most relevant paragraphs in the context window, making it easier for the model to access and utilize the best-supporting documents.

- The goal of the Lost In The Middle Ranker is to make it easy for an LLM to access the most relevant documents by placing them at the beginning and at the end of the context window. Lost In The Middle Ranker is meant to be used in combination with other rankers. In a RAG pipeline, place it as the last ranker after the relevance and diversity rankers.
- Similarity Ranker is useful in RAG pipeline or a document search pipeline, to ensure the retrieved Documents are ordered by relevance. You can use it after a Retriever to improve the search results. When using Transformers Similarity Ranker with a Retriever, consider setting the Retriever's top_k to a small number. This way the Ranker will have fewer Documents to process which can help make your pipeline faster.
- Reciprocal Rank Fusion (RRF) reranks the documents returned by both retrievers, giving priority to those which appear in both result lists. It's purpose is to push the most relevant documents to the top of the list. It is useful if the order of your results is important, or if you only want to pass on a subset of your results to the next component.

For the FIQA dataset:

- Large embedding models like Instructor-XL which take a instruction as input string regarding the domain tend to perform well in the Dense Retrieval pipeline.
- The Dense retrevial technique for the FIQA dataset performs the best. Adding the BM25 retreiver and subsequent rankers does not improve the performance for the FIQA dataset.
- Adding rankers individually in a Retrieval pipeline yield the best results.
- Adding pair of rankers (2 at a time) in a Retrieval pipeline may result in lower or at par performance compared to adding these rankers sequentially (one at a time) in a pipeline.
- Adding all the three rankers in a pipeline does not result in improvement in performance. It gives lower or at par results compared to adding the rankers sequentially.

References

Carbonell, Jaime and Jade Goldstein (1998). "The use of MMR, diversity-based reranking for reordering documents and producing summaries". In: *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*.

Cormack, Gordon V, Charles LA Clarke, and Stefan Buettcher (2009). "Reciprocal rank fusion outperforms condorcet and individual rank learning methods". In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*.

Cormack, Gordon V and Thomas R Lynam (2006). "Statistical precision of information retrieval evaluation". In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*.

Haldar, Malay, Mustafa Abdool, Liwei He, Dillon Davis, Huiji Gao, and Sanjeev Katariya (2023). "Learning To Rank Diversely At Airbnb". In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*.

Hull, David (1993). "Using statistical testing in the evaluation of retrieval experiments". In: *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*.

Järvelin, Kalervo and Jaana Kekäläinen (2000). "IR evaluation methods for retrieving highly relevant documents". In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. DOI: 10.1145/345508.345545.

Liu, Nelson F, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang (2023). "Lost in the middle: How language models use long contexts". In: *arXiv preprint arXiv:2307.03172*.

Reimers, Nils and Iryna Gurevych (2019). "Sentence-bert: Sentence embeddings using siamese bert-networks". In: *arXiv preprint arXiv:1908.10084*.

Song, Kaitao, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu (2020). "Mpnet: Masked and permuted pre-training for language understanding". In: *Advances in Neural Information Processing Systems* 33.

Su, Hongjin, 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". In: *arXiv preprint arXiv:2212.09741*.

Xiao, Shitao, Zheng Liu, Peitian Zhang, and Niklas Muennighof (2023). "C-pack: Packaged resources to advance general chinese embedding". In: *arXiv preprint arXiv:2309.07597*.