COL764/COL7364: Information Retrieval & Web Search

Assignment 4: Bi-encoders vs. Cross-encoders and Modified Sparse Retrieval

Naveeta Maheshwari (2024AIZ8309)
Sagar Singh (2025SIY7574)

Task 1: Implement the Bi-encoder Retriever with DistilBERT

November 13, 2025

Introduction

In this task, we implemented a dense retrieval model using the bi-encoder architecture with the DistilBERT model. This report presents a comparative analysis between two retrieval models — **Distilbert** and **Mono-bert**. Experiments were conducted across multiple cutoff values ($K = 10, 20, 50, 100$). While the effectiveness metrics remained constant across K, execution time varied, highlighting the trade-off between efficiency and performance.

The DistilBERT-based bi-encoder computes independent embeddings for queries and documents. During retrieval, the query embedding is compared with all pre-computed document embeddings using similarity measures (such as dot product). The FAISS index, an efficient vector search library, is used to quickly retrieve the top-k most relevant documents based on these similarity scores.

This dense retrieval pipeline provides semantic matching capabilities, capturing contextual meaning beyond surface-level keywords. However, bi-encoders trade off some cross-attention power for computational efficiency, making them ideal for large-scale retrieval tasks.

Evaluation Metrics

The performance of the DistilBERT dense retriever was evaluated using standard information retrieval metrics. These metrics assess both ranking quality and retrieval effectiveness:

- **nDCG@k (Normalized Discounted Cumulative Gain):** Measures ranking quality by assigning higher importance to correctly ranked documents appearing earlier in the list.
- **MRR@10 (Mean Reciprocal Rank):** Evaluates how early the first relevant document appears in the ranking.
- **Recall@10:** Proportion of relevant documents retrieved among the top 10 results.
- **Precision@10:** Fraction of retrieved documents in the top 10 that are relevant.
- **F1@10:** Harmonic mean of precision and recall, providing a balanced view of retrieval performance.

The following results were obtained for the DistilBERT run:

Metric	**Score**
nDCG@1	0.2333
nDCG@10	0.2115
MRR@10	0.3989
Recall@10	0.0814
Precision@10	0.2220
F1@10	0.0891

MonoBERT Reranked Results

MonoBERT re-evaluates the top-k documents retrieved by the bm25 using a cross-encoder approach, allowing deeper query-document interaction. The results below show evaluation metrics across various reranking depths (k = 10, 20, 50, 100).

k	nDCG@1	nDCG@10	MRR@10	Recall@10	Precision@10	F1@10
10	0.4600	0.3190	0.6074	0.1415	0.2900	0.1396
20	0.4267	0.3407	0.5824	0.1539	0.3300	0.1552
50	0.4467	0.3718	0.6051	0.1699	0.3640	0.1703
100	0.4600	0.3838	0.6271	0.1765	0.3680	0.1752

The results clearly show that as the reranking depth (k) increases, the overall performance improves across most metrics. The MonoBERT model consistently outperforms the DistilBERT bi-encoder in ranking quality (nDCG and MRR), demonstrating the advantage of cross-encoder architectures in capturing fine-grained relevance relationships between queries and documents.

Visualizations

To better understand the performance difference between the two models, Figure 1 illustrates a comparison between the DistilBERT bi-encoder and the MonoBERT reranker (k=10) across major evaluation metrics. The visualization clearly highlights that MonoBERT outperforms DistilBERT on all metrics, indicating the improved relevance ranking achieved through reranking.

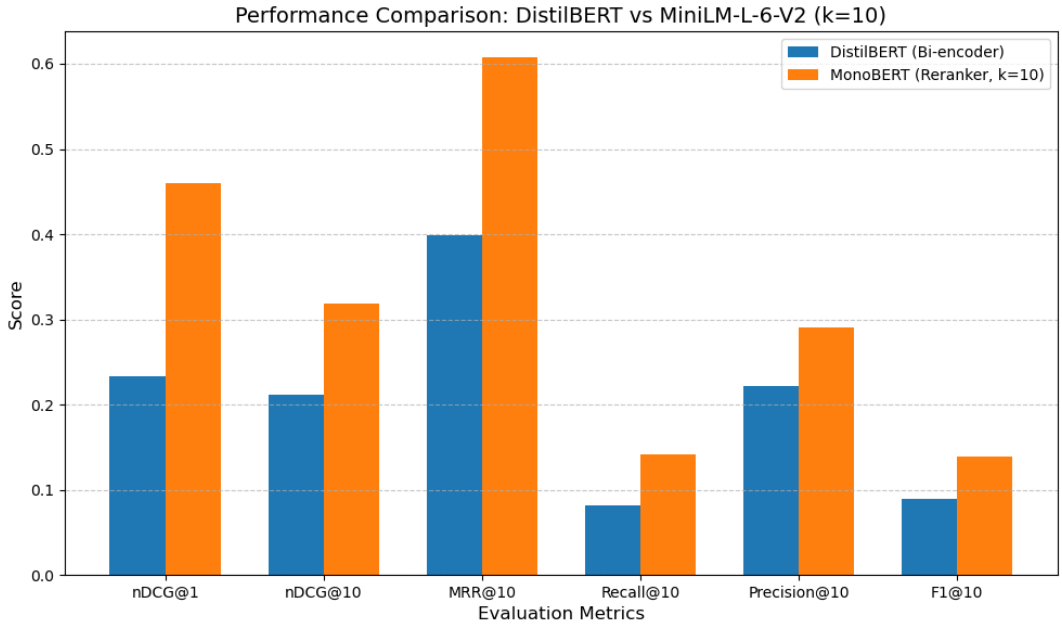

Figure 1: Performance comparison between DistilBERT (Bi-encoder) and MonoBERT (Reranker, k=10) across evaluation metrics.

Efficiency Evaluation

To evaluate computational efficiency, we recorded the total time taken for both Dense Retrieval (DistilBERT bi-encoder) and BM25 + MonoBERT Reranking (cross-encoder) for different top-k retrieval depths. The results are summarized in Table 1.

Table 1: Efficiency comparison of Dense Retrieval and BM25 + MonoBERT Reranking at different k values.

Top-k	**Dense Retrieval Time (s)**	**BM25 + Reranking Time (s)**
10	352.63	12.39
20	336.29	14.52
50	352.68	42.32
100	317.56	72.84

Hybrid Retrieval Evaluation with Varying α

In this experiment, we analyze the impact of varying the fusion weight α in the hybrid retrieval setup, where final document scores are computed as a weighted combination of BM25 (sparse) and MonoBERT (dense) scores:

$$Score_{hybrid} = \alpha \times score_{BM25} + (1 - \alpha) \times score_{Dense}$$

Here, α controls the contribution of BM25 and reranked (MonoBERT) scores to the overall ranking. We evaluate performance for $\alpha = 0.2$, $\alpha = 0.5$, and $\alpha = 0.8$ at $k = 50$.

Table 2: Hybrid Retrieval results for different α values at $k = 50$.

α	**nDCG@1**	**nDCG@10**	**MRR@10**	**Recall@10**	**Precision@10**	**F1@10**
0.2	0.4667	0.3822	0.6226	0.1799	0.3720	0.1791
0.5	0.4800	0.3731	0.6167	0.1794	0.3620	0.1779
0.8	0.4333	0.3391	0.5969	0.1578	0.3360	0.1588

From Table 2 and Figure 2, we observe that as α decreases, the influence of dense (MonoBERT) scores increases, leading to improved semantic ranking performance. The configuration $\alpha = 0.2$ achieves the highest overall performance across nearly all metrics, indicating that combining BM25 with reranked scores in a hybrid fashion allows the model to balance lexical and semantic retrieval strengths effectively.

This experiment demonstrates that hybrid fusion can outperform either pure BM25 or pure reranking, provided that α is tuned appropriately to balance sparse and dense representations.

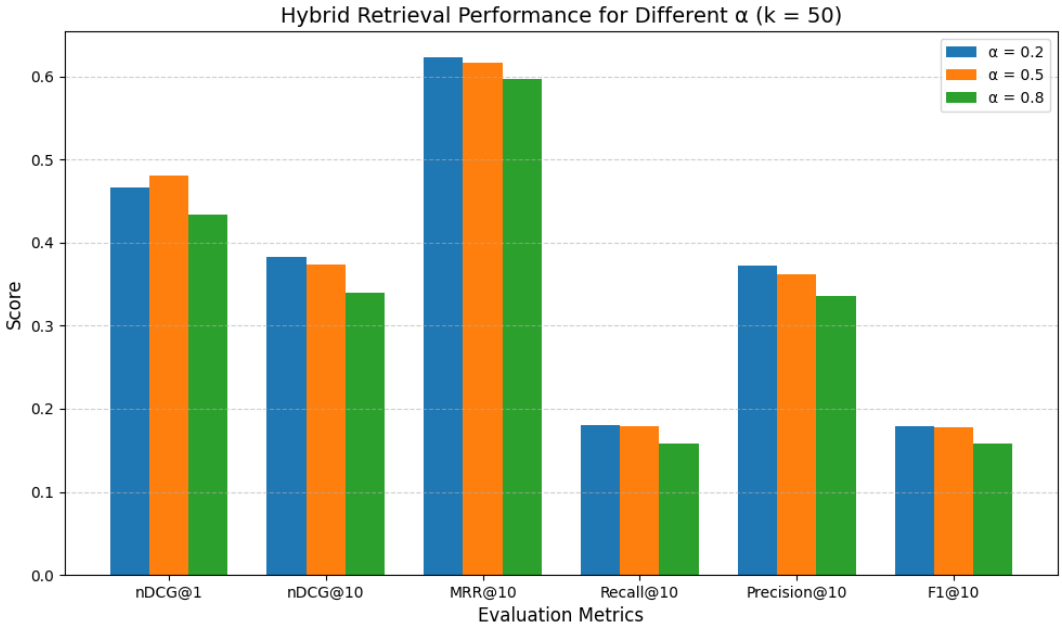

Figure 2: Hybrid Retrieval Performance across different α values ($k = 50$). Lower α values assign higher importance to dense reranking scores, resulting in improved ranking metrics such as nDCG and MRR.