Decoding Latent Spaces: LLM-based Embeddings vs Encoder-Only Architectures

David N. Juboor
Dr. John MacLaren Walsh
Dr. Christopher Weyant

December 9, 2024

Abstract

Text embeddings are foundational to retrieval tasks, enabling applications such as search and recommendation systems. While encoder-only architectures, such as BERT-based models, are well-established for generating robust embeddings, decoder-only architectures, optimized for autoregressive tasks, remain underexplored in this context. This study evaluates the comparative performance of decoder-only and encoder-only models in creating text embeddings for retrieval tasks, leveraging the Massive Text Embedding Benchmark (MTEB) for evaluation across diverse datasets and domains. We empirically validate that decoder-only models, such as `mistralai/Mistral-7B-v0.1`, are poorly suited for retrieval, achieving significantly lower performance compared to encoder-only architectures. Notably, embeddings from intermediate and penultimate layers in decoder models outperformed final-layer embeddings, highlighting potential layer-wise trends in feature representation. Additionally, task-specific fine-tuning of encoder-only models resulted in performance trade-offs, with slight improvements in some domains but overall degradation across general retrieval tasks, underscoring the complexities of fine-tuning for domain-specific use cases. These findings reinforce the superiority of encoder-only models for retrieval and contribute to the understanding of architectural trade-offs in text embedding models, offering actionable insights for practitioners optimizing embeddings for retrieval-based applications.

1 Introduction

Text embeddings are a cornerstone of modern natural language processing (NLP), enabling tasks such as information retrieval, semantic search, and recommendation systems. Encoder-only architectures, such as BERT-based models [1], are well-established for generating robust embeddings due to their ability to effectively model contextual relationships in text. However, the potential utility of decoder-only architectures, which are optimized for autoregressive tasks like next-word prediction, remains largely unexplored in the context of embedding-based retrieval apart from training data generation tasks [2] [3] [4] and "LLM as a Judge" evaluation frameworks [5]. This distinction between encoder- and decoder-based embeddings presents an intriguing question: how do these latent spaces differ when applied to the same retrieval tasks?

This work investigates the comparative performance of decoder-only and encoder-only architectures in creating text embeddings for retrieval tasks. Specifically, decoder-only models, such as `mistralai/Mistral-7B-v0.1` [6], are evaluated alongside encoder-only models, including both task-tuned and untuned variants, to examine whether the autoregressive training objectives of decoder models can offer any utility in retrieval applications. By leveraging the Massive Text Embedding Benchmark (MTEB) [7], a widely adopted framework for evaluating embedding models, we provide a robust comparison of these architectures across diverse domains and datasets.

The motivation for this study lies in understanding the philosophical and practical overlap between the tasks these models are optimized for. Encoder-only models excel at representing semantic relationships in text, while decoder-only models focus on sequential coherence. By comparing their embeddings, we aim to uncover whether the "understanding" encoded by these architectures exhibits any meaningful overlap or divergence in retrieval tasks. In addition, this work evaluates the effects of task-specific fine-tuning on encoder-only models, shedding light on its impact on both domain-specific and general retrieval performance.

This study makes two primary contributions:

- **Empirical Validation of Decoder-Only Limitations:** We confirm that decoder-only models, such as `Mistral-7B`, are poorly suited for retrieval tasks, achieving significantly lower performance compared to encoder-based architectures. Interestingly, we observe layer-wise trends in decoder embeddings, with intermediate and penultimate layers outperforming final-layer embeddings.
- **Insights into Task-Specific Fine-Tuning:** Our results demonstrate that task-specific fine-tuning of encoder-only models can degrade general retrieval performance, emphasizing the need for careful ablation studies to balance domain-specific optimization and task generalizability.

To evaluate these questions, we compare models across a wide range of retrieval

datasets from the MTEB framework. These datasets span diverse domains, including scientific, medical, and general QA, enabling a robust analysis of model performance across varying contexts. By synthesizing these results, we aim to provide actionable insights for researchers and practitioners seeking to optimize text embedding models for retrieval tasks.

The remainder of this paper is organized as follows: Section 2 reviews related work on embedding-based retrieval and model fine-tuning. Section 3 describes the experimental methodology, including model selection, datasets, and evaluation metrics. Section 4 presents the experimental results, and Section 5 discusses the key findings and their implications. Finally, the paper concludes with a summary of contributions and future directions.

2 Related Work

The effectiveness of text embeddings in retrieval tasks has been extensively studied, with a primary focus on encoder-only architectures. Decoder-only models, in contrast, are optimized for autoregressive generation and remain underexplored in retrieval contexts. This section situates the present study within the broader landscape of embedding research, task-specific fine-tuning, and evaluation frameworks.

2.1 Embedding Models for Retrieval

Encoder-only architectures, such as BERT [1] and Sentence-BERT [8], are well-established for retrieval tasks due to their ability to capture semantic relationships in text. Recent advances, such as task-tuned LLMs like E5 [3], have further demonstrated the potential of scaling and domain-specific optimization for embedding quality. These models leverage bidirectional context, making them ideal for generating robust embeddings.

In contrast, decoder-only architectures like Mistral-7B [6] are primarily designed for autoregressive tasks, such as next-word prediction. While decoder embeddings have been explored in tasks such as summarization or text generation, their utility in retrieval remains unclear. This study aims to bridge this gap by empirically evaluating decoder embeddings across diverse retrieval tasks.

2.2 Task-Specific Fine-Tuning

Fine-tuning pre-trained models for specific tasks or domains is a common strategy for improving embedding quality. Methods such as triplet-loss-based training have been shown to enhance retrieval performance by optimizing query-positive-negative relationships [4]. However, fine-tuning may introduce trade-offs, such as reduced generalizability to other tasks or domains. This study contributes to this area by highlighting the risks of task-specific fine-tuning,

particularly its potential to degrade performance across both target and non-target domains.

2.3 Evaluation Frameworks

Evaluation benchmarks like the Massive Text Embedding Benchmark (MTEB) [7] have standardized the assessment of embedding models across diverse tasks and datasets. MTEB includes retrieval-focused datasets spanning domains such as science, healthcare, and general QA, providing a robust framework for comparing models. By leveraging MTEB, this study ensures consistency and comparability with existing literature.

2.4 Addressing Gaps in Existing Research

While encoder-only architectures dominate retrieval-focused research, decoder-only models remain under explored in this context. This study addresses this gap by empirically validating the limitations of decoder embeddings. Additionally, the results highlight subtleties in fine-tuning strategies, emphasizing the need for ablation studies to balance domain-specific optimization with general retrieval performance.

3 Methodology

This study evaluates the effectiveness of embeddings generated by a range of models across diverse retrieval tasks. As a structured approach to compare decoder-only and encoder-only architecture embedding performance, our experimental design relies heavily on the use of consistent, 3rd party evaluation frameworks across a consistent set of datasets and specific evaluation metrics [9] in the Retrieval space. Leveraging well accepted evaluation frameworks in our experimental design, systemically addresses our core experimental questions around embedding and retrieval performance between varying embedding architectures in a reasonably reproducible fashion.

3.1 Models Evaluated

The models analyzed are categorized into two primary architectural groups: decoder-only and encoder-only. As decoder-only architectures are primarily utilized in for auto-regressive text generation, we choose this architecture as a base-line for unidirectional encoding not tuned for any particular retrieval task. In addition, we select three models to represent reasonable variants of encoder-only architectures. These distinctions provide a foundation for analyzing performance across retrieval tasks.

Decoder-Only Model The `mistralai/Mistral-7B-v0.1` model [6] represents a decoder-only architecture optimized for text generation. While not

inherently designed for retrieval tasks, embeddings were extracted from the model's hidden states to explore its potential for such applications. The following strategies were employed to extract embeddings:

- **Fused Average**: Combines hidden states from layers 15, 16, and the penultimate layer by averaging their representations.
- **Intermediate Layer**: Extracts embeddings directly from the 15th hidden state.
- **Penultimate Layer**: Uses the second-to-last hidden state for embeddings.
- **Final Layer Norm**: Relies on the normalized final hidden state for generating representations.

Encoder-Only Models Encoder-based architectures, optimized for high-quality text embeddings, were analyzed through three configurations each serving a unique embedding optimization strategy:

- **E5-Mistral-7B-Instruct**: This bi-encoder model [3] represents a more modern retrieval-optimized encoding architecture that has been fine-tuned on multi-domain data utilizing decoder-only model(s) to generate significant portions of its training data. Its embeddings were evaluated in their pre-trained form without additional modifications.
- **DistilBERT Baseline**: The `sentence-transformers/msmarco-distilbert-base-v4` model [1], widely used for retrieval tasks, represents a computationally efficient baseline in its pre-trained configuration. While used heavily in industry for its latency-performance tradeoff, this represents an "older" retrieval-optimized encoding architecture that has not been optimized explicitly for any of these particular retrieval domains, nor was it trained with LLM-generated data. It was selected primarily due to its retrieval-specific metric performance on general tasks with short queries and long answer context.
- **Task-Tuned DistilBERT**: A fine-tuned version of `msmarco-distilbert-base-v4`, trained over five epochs utilizing triplet loss and query-positive-negative pairs derived from selected in-domain yet out-of-distribution datasets. Training datasets included SciFact, toughdata/quora-question-answer-dataset, kroshan/BioASQ, and deepset/covid_qa_deepset. This architecture represents an "off-the-shelf retrieval optimized encoder" like our previous model, with the addition of being finetuned on data in the scientific domain. With this we aim to observe model performance gains or losses on a specific tasked domain while assessing generalization across other datasets both in and out of domain.

3.2 Datasets

The evaluation leveraged datasets from the Massive Text Embedding Benchmark (MTEB) Retrieval task [7], a widely accepted framework for benchmarking embedding models. The datasets span a broad range of domains and task types, ensuring a comprehensive evaluation. The full list includes:

- **Scientific and Academic Domains**: SciFact, SCIDOCS
- **General QA**: ArguAna
- **Finance**: FiQA2018
- **Healthcare and Environmental Domains**: NFCorpus
- **Community QA (CQADupStack)**: CQADupstackAndroidRetrieval, CQADupstackEnglishRetrieval, CQADupstackGamingRetrieval, CQADupstackGisRetrieval, CQADupstackMathematicaRetrieval, CQADupstackPhysicsRetrieval, CQADupstackProgrammersRetrieval, CQADupstackStatsRetrieval, CQADupstackTexRetrieval, CQADupstackUnixRetrieval, CQADupstackWebmastersRetrieval, CQADupstackWordpressRetrieval

Additionally, these datasets have good variation in overall relevant queries per document and are consistent with our "short queries long documents" retrieval paradigm as seen in our dataset breakdowns in Appendix A.4 .

The inclusion of datasets from scientific, finance, healthcare, general QA, and community QA contexts provides a robust evaluation set that captures both domain-specific and generalized retrieval challenges.

3.3 Evaluation Metrics

All datasets were evaluated on the "test set" as described by their respective dataset. In addition, by using the MTEB framework for evaluation, all models and datasets were evaluated with identical code and roughly 100 individual retrieval specific metrics were accumulated for all models on all datasets. That said, all datasets contained a "main score" that described what specific evaluation metric is most relevant for the dataset. As our datasets were Retrieval specific, each dataset noted a "main score" of Normalized Discounted Cumulative Gain at top 10 (NDCG@10), a widely used metric that considers both relevance and ranking position.

NDCG@k, a metric that relies heavily on Discounted Cumulative Gain, rewards systems that rank highly relevant documents closer to the top of the list, aligning with practical retrieval goals on a search results page.

The Discounted Cumulative Gain (DCG) is computed as:

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)} \tag{1}$$

where rel_i represents the relevance score of the retrieved item at position i. To normalize DCG, the Ideal Discounted Cumulative Gain (IDCG) is calculated for the best possible ranking:

$$\text{IDCG@k} = \sum_{i=1}^{k} \frac{2^{\text{rel}_i^{\text{ideal}}} - 1}{\log_2(i+1)} \tag{2}$$

Finally, NDCG is obtained as:

$$\text{NDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}} \tag{3}$$

This normalization ensures that NDCG ranges between 0 and 1, with higher scores indicating better performance [9]. While NDCG@10 was prioritized due to its alignment with both our evaluation datasets and ranking-focused retrieval tasks; auxiliary metrics, such as Mean Average Precision (MAP@k), Mean Reciprocal Rank (MRR@k), Precision@k, and Recall@k, were also tracked for additional insights. Additionally, k was evaluated at 20 different positions including 1, 2, 3, 5, 10, 20, 30, 100, etc and all metrics were evaluated as Area Under Curve (AUC) of a typical Receiver Operating Characteristic (ROC) curve. Where applicable, embeddings were precomputed or cached and evaluation metrics were computed against these values for operational efficiency.

3.4 Experimental Design

The experiments were structured to address three central research questions:

- How does task-specific fine-tuning improve retrieval performance compared to pre-trained models?
- How do embeddings from decoder-only models compare to those of encoder-only models in terms of retrieval effectiveness?
- Does fine-tuning on domain-specific datasets enhance retrieval within the domain, and how does it impact general performance across other tasks?

Embeddings for encoder-only models were extracted using standard library functions, while embeddings from the decoder-only model were derived from intermediate and final hidden states, as described earlier. The task-tuned DistilBERT model was fine-tuned using triplet loss to optimize retrieval for query-positive-negative pairs.

Baseline comparisons included:

- **Task-Tuned Baseline**: The E5-Mistral model served as a task-tuned baseline for retrieval tasks.
- **Non-Tuned Baseline**: The pre-trained DistilBERT model provided a computationally efficient, non-domain-specific baseline for broader comparisons.

4 Experiments

The experiments were designed to evaluate the performance of a decoder-only and encoder-only architectures of various classes across a wide range of retrieval datasets for the retrieval task. Each experiment addresses specific research questions, leveraging metrics such as NDCG@k, MAP@k, and MRR@k for evaluation. The experimental setups ensure consistency across datasets and metrics, while focusing on the nuances of model architectures and task-specific fine-tuning to allow a direct comparison between various embedding implementations for retrieval.

4.1 Experiment 1: Decoder-Only Model Evaluation

This experiment investigates the suitability of decoder-only architectures for retrieval tasks. The `mistralai/Mistral-7B-v0.1` model [6] was evaluated by extracting embeddings from four different encoding stages: fused average, intermediate layer, penultimate layer, and final layer normalization. Decoder-only models, such as Mistral-7B, are primarily optimized for autoregressive tasks, and their embeddings lack the bidirectional context present in encoder-based architectures. While atypical for usage in retrieval, this experiment serves as a baseline for utilizing using embeddings outside of their intended task (generation vs retrieval).

The primary objectives were:

- Quantify the performance gap between decoder-only and encoder-only models in retrieval tasks.
- Explore whether embeddings from specific layers exhibit better retrieval performance than others.

All evaluations were conducted across the full suite of evaluation datasets and metrics (described above), providing a robust assessment of decoder-only performance.

4.2 Experiment 2: Encoder-Only SLM vs LLM Comparison

This experiment compares an untuned encoder-only small language model (SLM) (`msmarco-distilbert-base-v4`) with the task-tuned large language model (LLM) encoder (`e5-mistral-7b-instruct`) [3]. The evaluation focuses on the trade-offs between embedding performance between relatively small and large encoder-only architectures. While the E5-Mistral model represents a high-performing large encoder-only baseline, DistilBERT provides a smaller, more lightweight, alternative for computationally constrained scenarios.

The goals of this experiment include:

- Evaluate the retrieval performance of the simpler SLM architecture relative to the more complex LLM encoder.
- Assess whether the increased performance of the LLM encoder justifies its higher computational cost in practical applications.

Metrics such as NDCG@1, 3, 5, and 10, as well as MAP, MRR, and others were used to quantify general performance while NDCG@10 was used as the primary score for all evaluation datasets.

4.3 Experiment 3: Task-Tuned SLM vs Non-Task-Tuned SLM

This experiment examines the impact of task-specific fine-tuning on an encoder only SLM. The `msmarco-distilbert-base-v4` model was fine tuned using out of distribution domain-specific training datasets from the scientific domain, including SciFact and BioASQ [5]. It is important to note that we chose traing data in a domain within our evaluation set without utilizing training sets from the evaluation datasets themselves. This was a strategic choice intended to steer our finetuning in favor of a specific domain within our task without explicitly biasing our model towards in-distribution data. Fine-tuning was performed with triplet loss, which optimizes embeddings for query-positive-negative relationships, improving retrieval performance.

Dataset Formatting for Fine-Tuning For fine-tuning, training datasets were transformed into a query-positive-negative format:

- Queries and positives were directly derived from the datasets.
- Hard negatives were sampled randomly using in-distribution data (data within the pardicular dataset).

This in-distribution domain-specific hard-negative sampling method ensures that negative examples are challenging yet relevant, which is critical for effective triplet-loss training [8].

The experiment evaluated the fine-tuned SLM on the full set of retrieval datasets, focusing on:

- Quantifying improvements in task performance resulting from fine-tuning in the scientific domain.
- Assessing whether fine-tuning impacts the model's generalization performance outside the scientific domain.

4.4 Experiment 4: Task-Tuned SLM vs Task-Tuned LLM

The final experiment compares the fine-tuned SLM from Experiment 3 with the task-tuned LLM encoder (`e5-mistral-7b-instruct`) [3]. Both models were

evaluated on all retrieval datasets, with particular attention to their performance on scientific datasets [5].

The objectives of this experiment are to determine whether task-specific fine-tuning allows the SLM to match or exceed the performance of the task-tuned LLM encoder within the scientific domain.

Metrics such as NDCG@k, MAP, and MRR were prioritized, and latency measurements were used to contextualize deployment considerations but not analyzed in depth for this study. For all experiments, NDCG@10 was used as the primary score under evaluation.

5 Key Results and Observations

The experiments provided insights into the performance of decoder-only and encoder-only architectures across a variety of retrieval tasks. This section summarizes the most critical findings, focusing on trends and anomalies observed during evaluation. Detailed numerical results and dataset-specific analyses are provided in the Appendix.

5.1 Decoder-Only Model Evaluation

The decoder-only architecture, represented by `mistralai/Mistral-7B-v0.1` [6], underperformed significantly across all retrieval tasks. This aligns with common industry knowledge that decoder-only models, optimized for autoregressive generation, struggle with retrieval tasks due to their unidirectional encoding and alternative training objectives. Key observations include:

- **Overall Performance**: Across all datasets, the decoder-only model achieved an average NDCG@10 score below 2%. Notable exceptions include SciFact and ArguAna, where performance increased to approximately 14–16%, likely due to alignment with specific pretraining data or objectives. Overall, these results show abysmal performance across all metrics and datasets when utilizing generative decoder-only architectures for retrieval tasks.
- **Layer-Wise Performance**: Embeddings from the penultimate layer consistently outperformed other layers, with the final layer performing worse than intermediate layers. This suggests that later layers, while still unsuitable for retrieval tasks, may contain more complex encodings better suited for transfer learning tasks.
- **Domain-Specific Trends**: The model performed relatively better on medical and scientific datasets compared to other domains (e.g., QA, programming). This trend may reflect pretraining biases, though further analysis of the model's (closed source) training data is needed.

5.2 Encoder-Only SLM vs LLM Comparisons

The task-tuned LLM encoder (`e5-mistral-7b-instruct`) [3] consistently outperformed the untuned SLM (`msmarco-distilbert-base-v4`) across all retrieval tasks. These findings highlight the advantages of task-specific tuning in large models (even with generated data), particularly for complex retrieval tasks:

- **Performance Gap**: The LLM encoder demonstrated a significant performance advantage, achieving an average main-score improvement of approximately 22.3% over the untuned SLM across all dataset domains. Scientific and medical datasets showed the largest margins.
- **Fine-Tuned SLM vs Task-Tuned LLM**: Despite fine-tuning the SLM on domain-specific datasets, the LLM encoder maintained a consistent advantage, with an average performance improvement of 28.5% across all domains. In scientific datasets, the LLM achieved an NDCG@10 score of 37.16%, compared to 24.99% for the fine-tuned SLM.
- [**Note**] **Computational Trade-Offs**: While the LLM encoder demonstrated higher performance, latency measurements indicate that the SLM may be a more reasonable option in scenarios where embedding latency is a critical factor.

5.3 Effects of Task-Specific Fine-Tuning

While finetuning hyperparameter optimization is an active area of research, fine-tuning the encoder-only SLM (`msmarco-distilbert-base-v4`) yielded mixed results. Using standard tuning hyperparameters for this task [8], our fine-tuned SLM showed a notable decrease in overall performance across most domains. These findings challenge the assumption that domain-specific fine-tuning universally improves retrieval performance:

- **Overall Performance**: Fine-tuning resulted in an average NDCG@10 decrease of 6.2% across all datasets. Surprisingly, the scientific domain, which was the target of the fine-tuning, performance decreased by 5.3%.
- **Domain-Specific Trends**: General QA and programming datasets experienced the largest decreases (-9.8% and -6.7%, respectively). Medical datasets showed a slight improvement (+0.38%), though this result may fall within the bounds of statistical variability and warrants further investigation.
- **Dataset-Level Insights**: SciFact, a dataset included in the fine-tuning process, experienced a performance decrease of 4.3%. Conversely, ArguAna, which was not included in the fine-tuning data, showed an unexpected performance increase of 3.3%, suggesting potential generalization effects specific to this dataset.

These results highlight the complexity of fine-tuning strategies and underscore the need for further investigation into optimizing domain-specific training processes for retrieval tasks. Additionally, our results confirm that, while computationally less efficient, more modern LLM-based encoder-only embeddings significantly outperform their alternative counterparts for both in and out of domain retrieval tasks.

6 Conclusion

This study evaluated the performance of decoder-only and encoder-only architectures for text retrieval tasks, providing practical insights into their effectiveness across diverse datasets and task-specific fine-tuning scenarios. Several key takeaways emerged from the experiments:

- Decoder-only models, such as `mistralai/Mistral-7B-v0.1`, are poorly suited for retrieval tasks, achieving an average NDCG@10 score of less than 2% across most datasets. While intermediate and penultimate layers outperformed the final layer, our results suggest that embeddings at any stage of decoder-only models are not suited for retrieval objectives.
- Task-tuned LLM encoders (`e5-mistral-7b-instruct`) consistently outperformed both untuned and fine-tuned SLMs (`msmarco-distilbert-base-v4`). The LLM encoder demonstrated an average performance improvement of 28.5% over the fine-tuned SLM, reinforcing the value of task-specific pretraining at scale.
- Domain-specific fine-tuning of SLMs showed mixed results, with a 6.2% overall performance decrease and degradation even in the fine-tuned domain (scientific datasets). These findings highlight the importance of careful ablation studies when fine-tuning for specific retrieval tasks, as unoptimized fine-tuning can substantially degrade both generalizability and task performance.

While the findings largely confirm existing assumptions about decoder-only and encoder-only architectures, they also offer empirical validation across a wide range of datasets. This study serves as a practical guide for researchers and engineers in choosing models and tuning strategies for retrieval tasks, emphasizing the importance of balancing task performance with computational constraints.

Several limitations of this work warrant consideration. First, small performance changes, such as the slight improvement observed in medical datasets, may fall within statistical variability and should be interpreted cautiously. Second, the experiments focused on a subset of models and datasets, and other architectures or fine-tuning methods may yield different results. Finally, biases in pretraining data were not explicitly analyzed, which could explain some domain-specific trends observed during evaluation.

Future research could explore improved fine-tuning strategies, such as con-

trastive learning, alternative sampling methods, or more structured curriculum-based approaches, to enhance generalizability while improving domain-specific performance. Additionally, expanding the scope of evaluated models to include other encoder-only architectures or LLM variants could provide a more comprehensive understanding of trade-offs in retrieval-focused embeddings.

In conclusion, this study underscores the importance of architectural choice and tuning strategies in embedding models for retrieval tasks. Decoder-only models remain unsuitable for retrieval, and task-tuning must be approached with care to ensure alignment with desired task objectives. These findings provide actionable guidance for the development of efficient and effective retrieval systems.

References

[1] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[2] C. Li, M. Qin, S. Xiao, J. Chen, K. Luo, Y. Shao, D. Lian, and Z. Liu, "Making text embedders few-shot learners," 2024. [Online]. Available: https://arxiv.org/abs/2409.15700

[3] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Improving text embeddings with large language models," 2024. [Online]. Available: https://arxiv.org/abs/2401.00368

[4] N. Thakur, N. Reimers, J. Daxenberger, and I. Gurevych, "Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks," 2021. [Online]. Available: https://arxiv.org/abs/2010.08240

[5] R. Friel, M. Belyi, and A. Sanyal, "Ragbench: Explainable benchmark for retrieval-augmented generation systems," 2024. [Online]. Available: https://arxiv.org/abs/2407.11005

[6] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b," 2023. [Online]. Available: https://arxiv.org/abs/2310.06825

[7] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "Mteb: Massive text embedding benchmark," 2023. [Online]. Available: https://arxiv.org/abs/2210.07316

[8] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[9] O. Jeunen, I. Potapov, and A. Ustimenko, "On (normalised) discounted cumulative gain as an off-policy evaluation metric for top-n recommendation," 2024. [Online]. Available: https://arxiv.org/abs/2307.15053

A Appendix

A.1 Detailed Results (Aggregated)

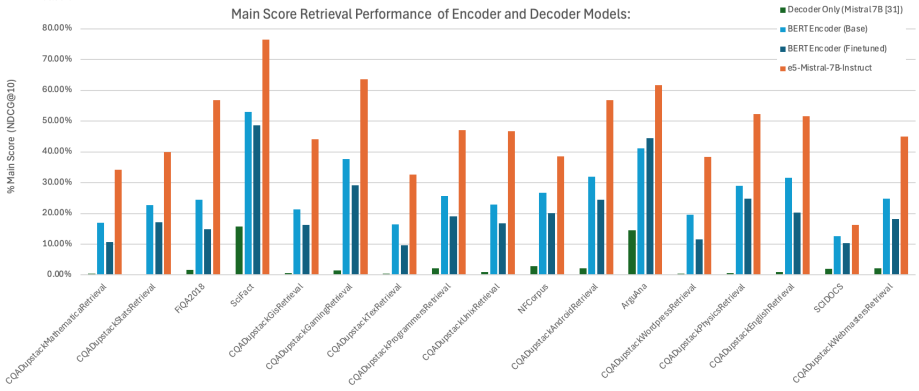


Figure 1: Dataset Specific Performance Metrics for All Models

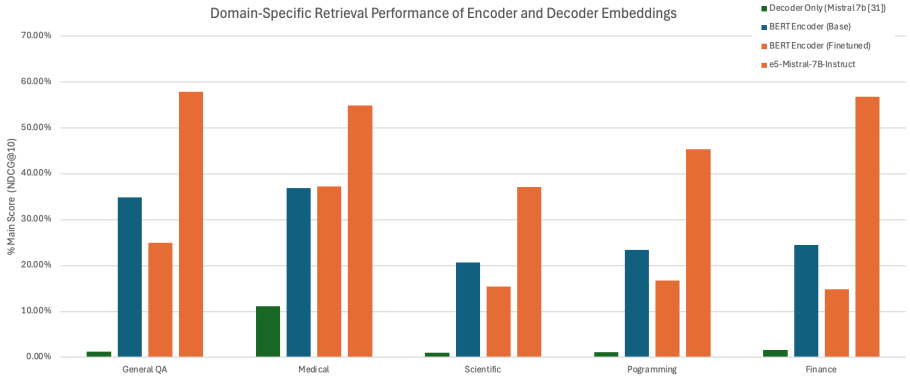


Figure 2: Domain Specific Performance Metrics for All Models

A.2 Decoder-Only Layer-Wise Performance

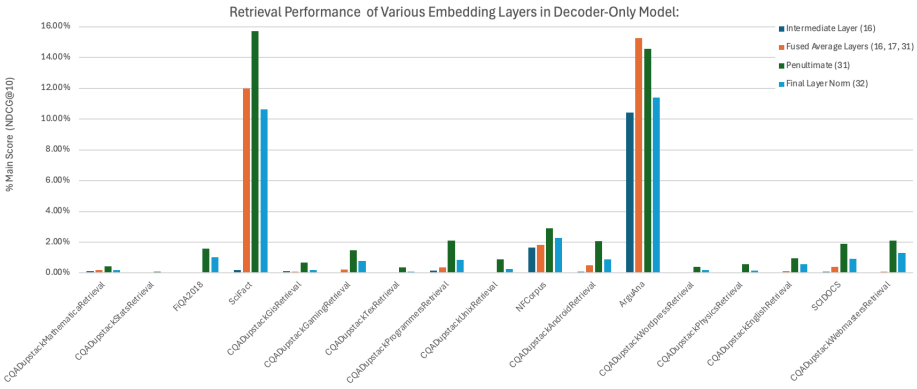


Figure 3: Dataset Specific Decoder-Only Embedding Performance by Layer

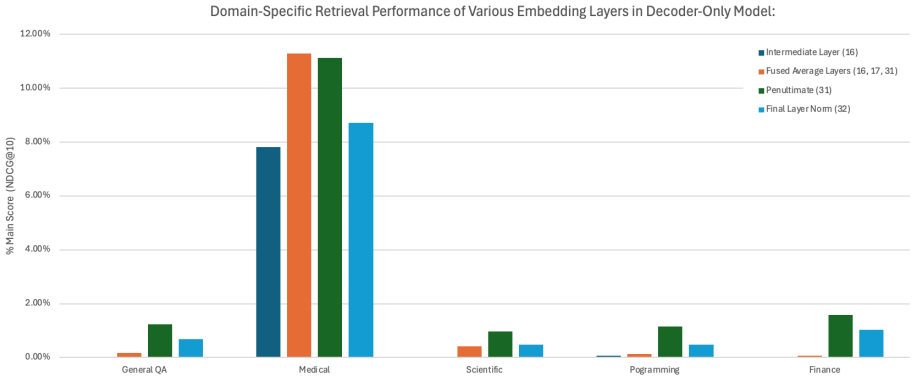
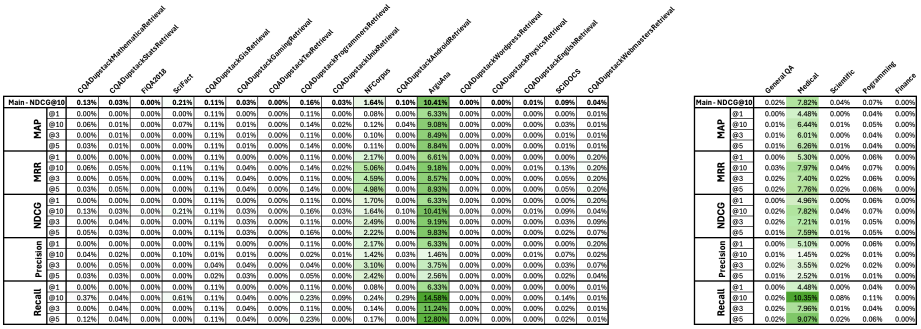


Figure 4: Domain Specific Decoder-Only Embedding Performance by Layer

A.3 Tabulated Results Across Various Metrics

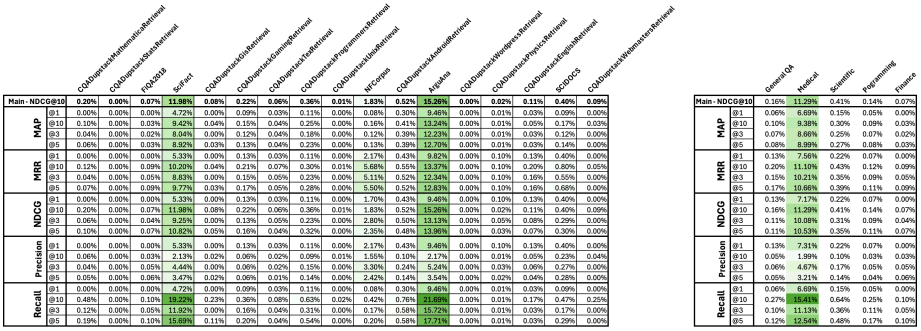
A.3.1 Decoder-Only Intermediate Embedding Layer Performance



Main - NDCG@10		CQADupstackMathematicaRetrieval	CQADupstackStatsRetrieval	FiQA2018	SciFact	CQADupstackGisRetrieval	CQADupstackGamingRetrieval	CQADupstackTexRetrieval	CQADupstackProgrammersRetrieval	CQADupstackUbuntuRetrieval	NFCorpus	CQADupstackAndroidRetrieval	ArguAna	CQADupstackWordpressRetrieval	CQADupstackPhysicsRetrieval	CQADupstackEnglishRetrieval	SCIDOCS	CQADupstackWebmastersRetrieval
		0.13%	0.03%	0.00%	0.21%	0.11%	0.03%	0.00%	0.16%	0.03%	1.64%	0.10%	10.41%	0.00%	0.00%	0.01%	0.09%	0.04%
MAP	@1	0.00%	0.00%	0.00%	0.00%	0.11%	0.00%	0.00%	0.11%	0.00%	0.08%	0.00%	6.33%	0.00%	0.00%	0.00%	0.00%	0.01%
	@10	0.06%	0.01%	0.00%	0.07%	0.11%	0.01%	0.00%	0.14%	0.02%	0.12%	0.04%	9.08%	0.00%	0.00%	0.00%	0.03%	0.01%
	@3	0.00%	0.01%	0.00%	0.00%	0.11%	0.01%	0.00%	0.11%	0.00%	0.10%	0.00%	8.49%	0.00%	0.00%	0.00%	0.01%	0.01%
	@5	0.03%	0.01%	0.00%	0.00%	0.11%	0.01%	0.00%	0.14%	0.00%	0.11%	0.00%	8.84%	0.00%	0.00%	0.00%	0.01%	0.01%
MRR	@1	0.00%	0.00%	0.00%	0.00%	0.11%	0.00%	0.00%	0.11%	0.00%	2.17%	0.00%	6.61%	0.00%	0.00%	0.00%	0.00%	0.20%
	@10	0.06%	0.05%	0.00%	0.11%	0.11%	0.04%	0.00%	0.14%	0.02%	5.06%	0.04%	9.18%	0.00%	0.00%	0.01%	0.13%	0.20%
	@3	0.00%	0.05%	0.00%	0.00%	0.11%	0.04%	0.00%	0.11%	0.00%	4.58%	0.00%	8.57%	0.00%	0.00%	0.00%	0.05%	0.20%
	@5	0.03%	0.05%	0.00%	0.00%	0.11%	0.04%	0.00%	0.14%	0.00%	4.98%	0.00%	8.93%	0.00%	0.00%	0.00%	0.05%	0.20%
NDCG	@1	0.00%	0.00%	0.00%	0.00%	0.11%	0.00%	0.00%	0.11%	0.00%	1.70%	0.00%	6.33%	0.00%	0.00%	0.00%	0.00%	0.20%
	@10	0.13%	0.03%	0.00%	0.21%	0.11%	0.03%	0.00%	0.16%	0.03%	1.64%	0.10%	10.41%	0.00%	0.00%	0.01%	0.09%	0.04%
	@3	0.00%	0.04%	0.00%	0.00%	0.11%	0.03%	0.00%	0.11%	0.00%	2.49%	0.00%	9.19%	0.00%	0.00%	0.00%	0.03%	0.09%
	@5	0.05%	0.03%	0.00%	0.00%	0.11%	0.03%	0.00%	0.16%	0.00%	2.22%	0.00%	9.83%	0.00%	0.00%	0.00%	0.02%	0.07%
Precision	@1	0.00%	0.00%	0.00%	0.00%	0.11%	0.00%	0.00%	0.11%	0.00%	2.17%	0.00%	6.33%	0.00%	0.00%	0.00%	0.00%	0.20%
	@10	0.04%	0.02%	0.00%	0.10%	0.01%	0.01%	0.00%	0.02%	0.01%	1.42%	0.03%	1.46%	0.00%	0.00%	0.01%	0.07%	0.02%
	@3	0.00%	0.05%	0.00%	0.00%	0.04%	0.04%	0.00%	0.04%	0.00%	3.10%	0.00%	3.75%	0.00%	0.00%	0.00%	0.03%	0.07%
	@5	0.03%	0.03%	0.00%	0.00%	0.02%	0.03%	0.00%	0.05%	0.00%	2.42%	0.00%	2.56%	0.00%	0.00%	0.00%	0.02%	0.04%
Recall	@1	0.00%	0.00%	0.00%	0.00%	0.11%	0.00%	0.00%	0.11%	0.00%	0.08%	0.00%	6.33%	0.00%	0.00%	0.00%	0.00%	0.01%
	@10	0.37%	0.04%	0.00%	0.61%	0.11%	0.04%	0.00%	0.23%	0.09%	0.24%	0.29%	14.58%	0.00%	0.00%	0.00%	0.14%	0.01%
	@3	0.00%	0.04%	0.00%	0.00%	0.11%	0.04%	0.00%	0.11%	0.00%	0.14%	0.00%	11.24%	0.00%	0.00%	0.00%	0.02%	0.01%
	@5	0.12%	0.04%	0.00%	0.00%	0.11%	0.04%	0.00%	0.23%	0.00%	0.17%	0.00%	12.80%	0.00%	0.00%	0.00%	0.02%	0.01%

Main - NDCG@10		General QA	Medical	Scientific	Programming	Finance
		0.02%	7.82%	0.04%	0.07%	0.00%
MAP	@1	0.00%	4.48%	0.00%	0.04%	0.00%
	@10	0.01%	6.44%	0.01%	0.05%	0.00%
	@3	0.01%	6.01%	0.00%	0.04%	0.00%
	@5	0.01%	6.26%	0.01%	0.04%	0.00%
MRR	@1	0.00%	5.30%	0.00%	0.06%	0.00%
	@10	0.03%	7.97%	0.04%	0.07%	0.00%
	@3	0.02%	7.40%	0.02%	0.06%	0.00%
	@5	0.02%	7.76%	0.02%	0.06%	0.00%
NDCG	@1	0.00%	4.96%	0.00%	0.06%	0.00%
	@10	0.02%	7.82%	0.04%	0.07%	0.00%
	@3	0.02%	7.21%	0.01%	0.05%	0.00%
	@5	0.01%	7.59%	0.01%	0.05%	0.00%
Precision	@1	0.00%	5.10%	0.00%	0.06%	0.00%
	@10	0.01%	1.45%	0.02%	0.01%	0.00%
	@3	0.02%	3.55%	0.02%	0.02%	0.00%
	@5	0.01%	2.52%	0.01%	0.01%	0.00%
Recall	@1	0.00%	4.48%	0.00%	0.04%	0.00%
	@10	0.02%	18.35%	0.08%	0.11%	0.00%
	@3	0.02%	7.96%	0.01%	0.04%	0.00%
	@5	0.02%	9.07%	0.02%	0.06%	0.00%

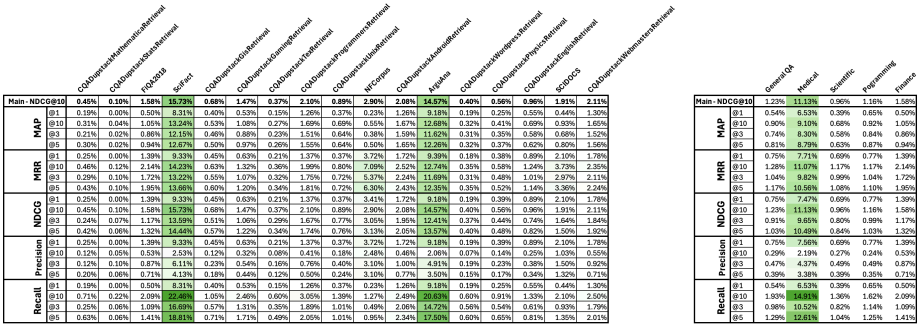
Figure 5: Decoder-Only Intermediate Layer [16] Performance



Main - NDCG@10		CQADupstackMathematicaRetrieval	CQADupstackStatsRetrieval	FiQA2018	SciFact	CQADupstackGisRetrieval	CQADupstackGamingRetrieval	CQADupstackTexRetrieval	CQADupstackProgrammersRetrieval	CQADupstackUbuntuRetrieval	NFCorpus	CQADupstackAndroidRetrieval	ArguAna	CQADupstackWordpressRetrieval	CQADupstackPhysicsRetrieval	CQADupstackEnglishRetrieval	SCIDOCS	CQADupstackWebmastersRetrieval
		0.20%	0.00%	0.07%	11.98%	0.08%	0.22%	0.06%	0.36%	0.01%	1.83%	0.52%	15.26%	0.00%	0.02%	0.11%	0.40%	0.08%
MAP	@1	0.00%	0.00%	0.00%	4.72%	0.00%	0.09%	0.03%	0.11%	0.00%	0.08%	0.30%	9.46%	0.00%	0.01%	0.03%	0.09%	0.00%
	@10	0.10%	0.00%	0.03%	9.42%	0.08%	0.22%	0.06%	0.36%	0.01%	1.83%	0.52%	15.26%	0.00%	0.02%	0.11%	0.40%	0.08%
	@3	0.04%	0.00%	0.02%	8.04%	0.00%	0.12%	0.04%	0.18%	0.00%	0.12%	0.39%	12.23%	0.00%	0.01%	0.03%	0.12%	0.00%
	@5	0.06%	0.00%	0.03%	8.92%	0.03%	0.13%	0.04%	0.23%	0.00%	0.13%	0.39%	12.70%	0.00%	0.01%	0.03%	0.14%	0.00%
MRR	@1	0.00%	0.00%	0.00%	5.33%	0.00%	0.13%	0.03%	0.11%	0.00%	2.17%	0.43%	9.82%	0.00%	0.10%	0.13%	0.40%	0.00%
	@10	0.12%	0.00%	0.09%	10.20%	0.04%	0.21%	0.07%	0.30%	0.01%	5.68%	0.55%	13.37%	0.00%	0.10%	0.20%	0.80%	0.05%
	@3	0.04%	0.00%	0.05%	8.83%	0.00%	0.15%	0.05%	0.23%	0.00%	5.11%	0.52%	12.34%	0.00%	0.10%	0.16%	0.55%	0.00%
	@5	0.07%	0.00%	0.09%	9.77%	0.03%	0.17%	0.05%	0.28%	0.00%	5.50%	0.52%	12.83%	0.00%	0.10%	0.16%	0.68%	0.00%
NDCG	@1	0.00%	0.00%	0.00%	5.33%	0.00%	0.13%	0.03%	0.11%	0.00%	1.70%	0.43%	9.46%	0.00%	0.10%	0.13%	0.40%	0.00%
	@10	0.20%	0.00%	0.07%	11.98%	0.08%	0.22%	0.06%	0.36%	0.01%	1.83%	0.52%	15.26%	0.00%	0.02%	0.11%	0.40%	0.08%
	@3	0.06%	0.00%	0.04%	9.25%	0.00%	0.13%	0.05%	0.23%	0.00%	2.80%	0.50%	13.13%	0.00%	0.05%	0.08%	0.29%	0.00%
	@5	0.10%	0.00%	0.07%	10.82%	0.05%	0.16%	0.04%	0.32%	0.00%	2.35%	0.48%	13.96%	0.00%	0.03%	0.07%	0.30%	0.00%
Precision	@1	0.00%	0.00%	0.00%	5.33%	0.00%	0.13%	0.03%	0.11%	0.00%	2.17%	0.43%	9.46%	0.00%	0.10%	0.13%	0.40%	0.00%
	@10	0.06%	0.00%	0.03%	2.13%	0.02%	0.06%	0.02%	0.09%	0.01%	1.55%	0.10%	2.17%	0.00%	0.01%	0.05%	0.23%	0.04%
	@3	0.04%	0.00%	0.05%	4.44%	0.00%	0.06%	0.02%	0.15%	0.00%	3.30%	0.24%	5.24%	0.00%	0.03%	0.06%	0.27%	0.00%
	@5	0.05%	0.00%	0.06%	3.47%	0.02%	0.06%	0.01%	0.14%	0.00%	2.42%	0.14%	3.54%	0.00%	0.02%	0.04%	0.28%	0.00%
Recall	@1	0.00%	0.00%	0.00%	4.72%	0.00%	0.09%	0.03%	0.11%	0.00%	0.08%	0.30%	9.46%	0.00%	0.01%	0.03%	0.09%	0.00%
	@10	0.48%	0.00%	0.10%	19.22%	0.23%	0.36%	0.08%	0.63%	0.02%	0.42%	0.76%	21.69%	0.00%	0.01%	0.17%	0.47%	0.25%
	@3	0.12%	0.00%	0.05%	11.92%	0.00%	0.16%	0.04%	0.31%	0.00%	0.17%	0.58%	15.72%	0.00%	0.01%	0.03%	0.17%	0.00%
	@5	0.19%	0.00%	0.10%	15.69%	0.11%	0.20%	0.04%	0.54%	0.00%	0.20%	0.58%	17.71%	0.00%	0.01%	0.03%	0.29%	0.00%

Main - NDCG@10		General QA	Medical	Scientific	Programming	Finance
		0.16%	11.29%	0.41%	0.14%	0.07%
MAP	@1	0.06%	6.69%	0.15%	0.05%	0.00%
	@10	0.10%	9.38%	0.30%	0.09%	0.03%
	@3	0.07%	8.66%	0.25%	0.07%	0.02%
	@5	0.08%	8.99%	0.27%	0.08%	0.03%
MRR	@1	0.13%	7.56%	0.22%	0.07%	0.00%
	@10	0.20%	11.10%	0.43%	0.12%	0.09%
	@3	0.15%	10.21%	0.35%	0.09%	0.05%
	@5	0.17%	10.66%	0.39%	0.11%	0.09%
NDCG	@1	0.13%	7.17%	0.22%	0.07%	0.00%
	@10	0.16%	11.29%	0.41%	0.14%	0.07%
	@3	0.11%	10.08%	0.31%	0.09%	0.04%
	@5	0.11%	10.53%	0.35%	0.11%	0.07%
Precision	@1	0.13%	7.31%	0.22%	0.07%	0.00%
	@10	0.05%	1.99%	0.10%	0.03%	0.03%
	@3	0.06%	4.67%	0.17%	0.05%	0.05%
	@5	0.05%	3.21%	0.14%	0.04%	0.06%
Recall	@1	0.06%	6.69%	0.15%	0.05%	0.00%
	@10	0.27%	15.41%	0.64%	0.25%	0.10%
	@3	0.10%	11.13%	0.36%	0.11%	0.05%
	@5	0.12%	12.54%	0.48%	0.17%	0.10%

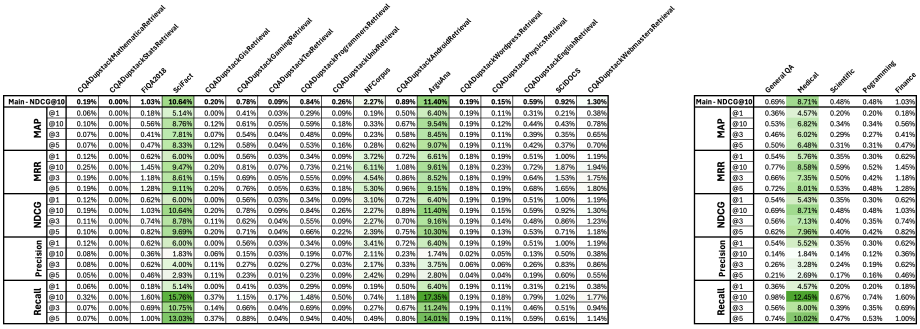
Figure 6: Decoder-Only Fused Average Layers [16, 17, and 31] Performance



Main - NDCG@10		CQADupstackMathematicaRetrieval	CQADupstackStatsRetrieval	FiQA2018	SciFact	CQADupstackGisRetrieval	CQADupstackGamingRetrieval	CQADupstackTexRetrieval	CQADupstackProgrammersRetrieval	CQADupstackUbuntuRetrieval	NFCorpus	CQADupstackAndroidRetrieval	ArguAna	CQADupstackWordpressRetrieval	CQADupstackPhysicsRetrieval	CQADupstackEnglishRetrieval	SCIDOCS	CQADupstackWebmastersRetrieval
		0.45%	0.10%	1.58%	15.73%	0.68%	1.47%	0.37%	2.10%	0.89%	2.90%	2.08%	14.57%	0.40%	0.56%	0.96%	1.91%	2.11%
MAP	@1	0.19%	0.00%	0.50%	8.31%	0.40%	0.53%	0.15%	1.26%	0.37%	0.23%	1.26%	9.18%	0.19%	0.25%	0.55%	0.44%	1.30%
	@10	0.31%	0.04%	1.05%	13.24%	0.53%	1.08%	0.27%	1.69%	0.69%	0.55%	1.67%	12.68%	0.32%	0.41%	0.69%	0.93%	1.65%
	@3	0.21%	0.02%	0.86%	12.15%	0.49%	0.88%	0.23%	1.51%	0.64%	0.38%	1.59%	11.62%	0.31%	0.35%	0.58%	0.68%	1.52%
	@5	0.30%	0.02%	0.94%	12.67%	0.50%	0.97%	0.26%	1.55%	0.64%	0.50%	1.65%	12.26%	0.32%	0.37%	0.62%	0.80%	1.56%
MRR	@1	0.25%	0.00%	1.39%	9.33%	0.45%	0.63%	0.21%	1.37%	0.37%	3.72%	1.72%	9.39%	0.18%	0.38%	0.89%	2.10%	1.78%
	@10	0.46%	0.12%	2.14%	14.23%	0.63%	1.32%	0.36%	1.99%	0.80%	7.08%	2.52%	12.74%	0.35%	0.58%	1.24%	3.73%	2.35%
	@3	0.29%	0.10%	1.72%	13.22%	0.55%	1.07%	0.32%	1.75%	0.72%	5.37%	2.24%	11.69%	0.31%	0.48%	1.01%	2.97%	2.11%
	@5	0.43%	0.10%	1.95%	13.86%	0.60%	1.20%	0.34%	1.81%	0.72%	6.30%	2.43%	12.35%	0.35%	0.52%	1.14%	3.36%	2.24%
NDCG	@1	0.25%	0.00%	1.39%	9.33%	0.45%	0.63%	0.21%	1.37%	0.37%	3.41%	1.72%	9.18%	0.19%	0.39%	0.89%	2.10%	1.78%
	@10	0.45%	0.10%	1.58%	15.73%	0.68%	1.47%	0.37%	2.10%	0.89%	2.90%	2.08%	14.57%	0.40%	0.56%	0.96%	1.91%	2.11%
	@3	0.24%	0.07%	1.17%	13.59%	0.51%	1.06%	0.29%	1.67%	0.77%	3.05%	1.95%	12.41%	0.37%	0.44%	0.74%	1.64%	1.84%
	@5	0.42%	0.06%	1.32%	14.44%	0.57%	1.22%	0.34%	1.74%	0.76%	3.13%	2.05%	13.57%	0.40%	0.48%	0.82%	1.50%	1.92%
Precision	@1	0.25%	0.00%	1.39%	9.33%	0.45%	0.63%	0.21%	1.37%	0.37%	3.72%	1.72%	9.18%	0.19%	0.39%	0.89%	2.10%	1.78%
	@10	0.12%	0.05%	0.53%	2.53%	0.12%	0.32%	0.08%	0.41%	0.18%	2.48%	0.46%	2.06%	0.07%	0.14%	0.25%	1.03%	0.55%
	@3	0.12%	0.10%	0.87%	6.11%	0.23%	0.54%	0.16%	0.76%	0.40%	3.10%	1.00%	4.91%	0.19%	0.23%	0.38%	1.50%	0.92%
	@5	0.20%	0.06%	0.71%	4.13%	0.18%	0.44%	0.12%	0.50%	0.24%	3.10%	0.77%	3.50%	0.15%	0.17%	0.34%	1.32%	0.71%
Recall	@1	0.19%	0.00%	0.50%	8.31%	0.40%	0.53%	0.15%	1.26%	0.37%	0.23%	1.26%	9.18%	0.19%	0.25%	0.55%	0.44%	1.30%
	@10	0.71%	0.22%	2.09%	22.48%	1.05%	2.46%	0.60%	3.05%	1.39%	1.27%	2.49%	20.63%	0.60%	0.91%	1.33%	2.10%	2.50%
	@3	0.25%	0.06%	1.09%	16.69%	0.57%	1.31%	0.35%	1.89%	1.01%	0.49%	2.06%	14.72%	0.56%	0.54%	0.61%	0.93%	1.79%
	@5	0.63%	0.06%	1.41%	18.81%	0.71%	1.71%	0.49%	2.05%	1.01%	0.95%	2.34%	17.50%	0.60%	0.65%	0.81%	1.35%	2.01%

Main - NDCG@10		General QA	Medical	Scientific	Programming	Finance
		1.23%	11.13%	0.96%	1.16%	1.58%
MAP	@1	0.54%	6.53%	0.39%	0.65%	0.50%
	@10	0.90%	9.10%	0.68%	0.92%	1.05%
	@3	0.74%	8.30%	0.58%	0.84%	0.86%
	@5	0.81%	8.79%	0.63%	0.87%	0.94%
MRR	@1	0.75%	7.71%	0.69%	0.77%	1.39%
	@10	1.28%	11.07%	1.17%	1.17%	2.14%
	@3	1.04%	9.82%	0.96%	1.04%	1.72%
	@5	1.17%	10.56%	1.08%	1.10%	1.95%
NDCG	@1	0.75%	7.47%	0.69%	0.77%	1.39%
	@10	1.23%	11.13%	0.96%	1.16%	1.58%
	@3	0.91%	9.65%	0.80%	0.99%	1.17%
	@5	1.03%	10.49%	0.84%	1.03%	1.32%
Precision	@1	0.75%	7.56%	0.69%	0.77%	1.39%
	@10	0.29%	2.19%	0.27%	0.24%	0.53%
	@3	0.47%	4.37%	0.49%	0.49%	0.87%
	@5	0.39%	3.38%	0.39%	0.35%	0.71%
Recall	@1	0.54%	6.53%	0.39%	0.65%	0.50%
	@10	1.93%	14.91%	1.36%	1.62%	2.09%
	@3	0.98%	10.52%	0.82%	1.14%	1.09%
	@5	1.29%	12.61%	1.04%	1.25%	1.41%

Figure 7: Decoder-Only Penultimate Layer [31] Performance

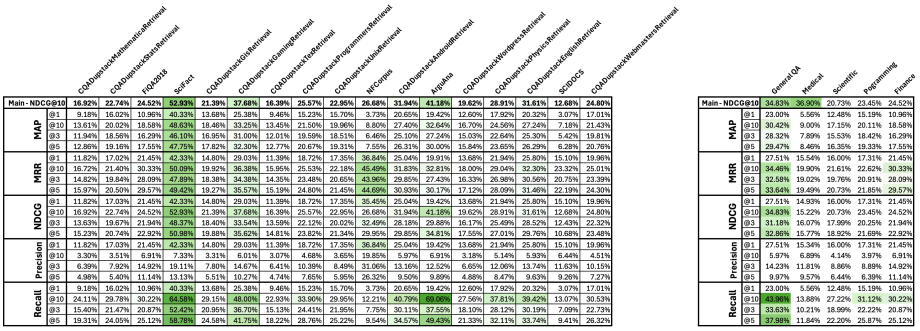


Main - NDCG@10		0.19%	0.00%	1.03%	10.64%	0.20%	0.78%	0.09%	0.84%	0.26%	2.27%	0.89%	11.40%	0.19%	0.15%	0.59%	0.92%	1.30%
MAP	@1	0.06%	0.00%	0.18%	5.14%	0.00%	0.41%	0.03%	0.29%	0.09%	0.19%	0.50%	6.40%	0.19%	0.11%	0.31%	0.21%	0.38%
	@10	0.10%	0.00%	0.56%	8.76%	0.12%	0.61%	0.05%	0.59%	0.18%	0.33%	0.67%	9.54%	0.19%	0.12%	0.44%	0.43%	0.78%
	@3	0.07%	0.00%	0.41%	7.81%	0.07%	0.54%	0.04%	0.48%	0.09%	0.23%	0.58%	8.45%	0.19%	0.11%	0.39%	0.35%	0.65%
	@5	0.07%	0.00%	0.47%	8.33%	0.12%	0.58%	0.04%	0.53%	0.16%	0.28%	0.62%	9.07%	0.19%	0.11%	0.42%	0.37%	0.70%
MRR	@1	0.12%	0.00%	0.62%	6.00%	0.00%	0.56%	0.03%	0.34%	0.09%	3.72%	0.72%	6.61%	0.18%	0.19%	0.51%	1.00%	1.19%
	@10	0.25%	0.00%	1.45%	9.47%	0.20%	0.81%	0.07%	0.73%	0.21%	6.11%	1.08%	9.61%	0.18%	0.23%	0.72%	1.87%	1.94%
	@3	0.19%	0.00%	1.18%	8.61%	0.15%	0.69%	0.05%	0.55%	0.09%	4.54%	0.86%	8.52%	0.18%	0.19%	0.64%	1.53%	1.75%
	@5	0.19%	0.00%	1.28%	9.11%	0.20%	0.76%	0.05%	0.63%	0.18%	5.30%	0.96%	9.15%	0.18%	0.19%	0.68%	1.65%	1.80%
NDCG	@1	0.12%	0.00%	0.62%	6.00%	0.00%	0.56%	0.03%	0.34%	0.09%	3.10%	0.72%	6.40%	0.19%	0.19%	0.51%	1.00%	1.19%
	@10	0.19%	0.00%	1.03%	10.64%	0.20%	0.78%	0.09%	0.84%	0.26%	2.27%	0.89%	11.40%	0.19%	0.15%	0.59%	0.92%	1.30%
	@3	0.11%	0.00%	0.74%	8.78%	0.11%	0.62%	0.04%	0.55%	0.09%	2.27%	0.70%	9.16%	0.19%	0.14%	0.48%	0.86%	1.23%
	@5	0.10%	0.00%	0.82%	9.69%	0.20%	0.71%	0.04%	0.66%	0.22%	2.38%	0.75%	10.30%	0.19%	0.13%	0.53%	0.71%	1.18%
Precision	@1	0.12%	0.00%	0.62%	6.00%	0.00%	0.56%	0.03%	0.34%	0.09%	3.41%	0.72%	6.40%	0.19%	0.19%	0.51%	1.00%	1.19%
	@10	0.08%	0.00%	0.36%	1.83%	0.06%	0.15%	0.03%	0.19%	0.07%	2.11%	0.23%	1.74%	0.02%	0.05%	0.13%	0.50%	0.38%
	@3	0.08%	0.00%	0.62%	4.00%	0.11%	0.27%	0.02%	0.27%	0.03%	2.17%	0.33%	3.75%	0.06%	0.06%	0.26%	0.83%	0.86%
	@5	0.05%	0.00%	0.46%	2.93%	0.11%	0.23%	0.01%	0.23%	0.09%	2.42%	0.29%	2.80%	0.04%	0.04%	0.19%	0.60%	0.55%
Recall	@1	0.06%	0.00%	0.18%	5.14%	0.00%	0.41%	0.03%	0.29%	0.09%	0.19%	0.50%	6.40%	0.19%	0.11%	0.31%	0.21%	0.38%
	@10	0.32%	0.00%	1.60%	15.76%	0.37%	1.15%	0.17%	1.48%	0.50%	0.74%	1.18%	17.35%	0.19%	0.18%	0.79%	1.02%	1.77%
	@3	0.07%	0.00%	0.69%	10.75%	0.14%	0.66%	0.04%	0.69%	0.09%	0.27%	0.67%	11.24%	0.19%	0.11%	0.46%	0.51%	0.94%
	@5	0.07%	0.00%	1.00%	13.03%	0.37%	0.88%	0.04%	0.94%	0.40%	0.49%	0.80%	14.01%	0.19%	0.11%	0.59%	0.61%	1.14%

Main - NDCG@10		0.69%	8.71%	0.48%	0.48%	1.03%
MAP	@1	0.36%	4.57%	0.20%	0.20%	0.18%
	@10	0.53%	6.82%	0.34%	0.34%	0.56%
	@3	0.46%	6.02%	0.29%	0.27%	0.41%
	@5	0.50%	6.48%	0.31%	0.31%	0.47%
MRR	@1	0.54%	5.76%	0.35%	0.30%	0.62%
	@10	0.77%	8.58%	0.59%	0.52%	1.45%
	@3	0.66%	7.35%	0.50%	0.42%	1.18%
	@5	0.72%	8.01%	0.53%	0.48%	1.28%
NDCG	@1	0.54%	5.43%	0.35%	0.30%	0.62%
	@10	0.69%	8.71%	0.48%	0.48%	1.03%
	@3	0.56%	7.13%	0.40%	0.35%	0.74%
	@5	0.62%	7.96%	0.40%	0.42%	0.82%
Precision	@1	0.54%	5.52%	0.35%	0.30%	0.62%
	@10	0.14%	1.84%	0.14%	0.12%	0.36%
	@3	0.26%	3.28%	0.24%	0.19%	0.62%
	@5	0.21%	2.69%	0.17%	0.16%	0.46%
Recall	@1	0.36%	4.57%	0.20%	0.20%	0.18%
	@10	0.96%	12.45%	0.67%	0.74%	1.60%
	@3	0.56%	8.00%	0.39%	0.35%	0.69%
	@5	0.74%	10.02%	0.47%	0.53%	1.00%

Figure 8: Decoder-Only Final Layer Norm [32] Performance

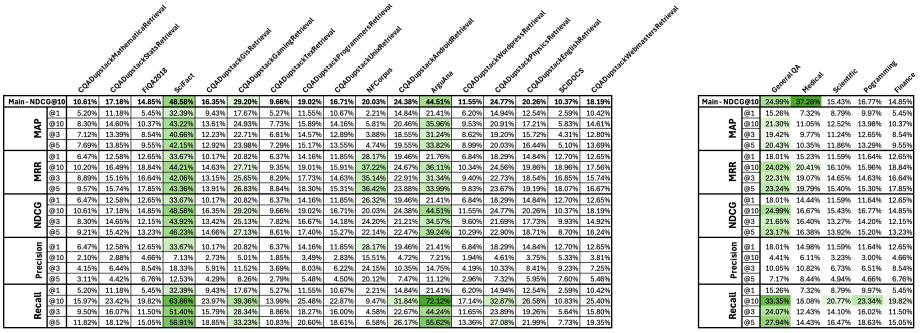
A.3.2 Encoder-Only Embedding Performance



Main - NDCG@10		16.92%	22.74%	24.52%	52.93%	21.39%	37.68%	16.39%	25.57%	22.95%	26.68%	31.94%	41.18%	19.62%	28.91%	31.61%	12.68%	24.80%
MAP	@1	9.18%	16.02%	10.96%	40.33%	13.68%	25.38%	9.46%	15.23%	15.70%	3.73%	20.65%	19.42%	12.60%	17.92%	20.32%	3.07%	17.01%
	@10	13.61%	20.02%	18.58%	48.63%	18.46%	33.25%	13.45%	21.50%	19.96%	8.80%	27.40%	32.64%	16.70%	24.56%	27.24%	7.18%	21.43%
	@3	11.94%	18.56%	16.29%	46.10%	16.95%	31.00%	12.01%	19.59%	18.51%	6.46%	25.10%	27.24%	15.03%	22.64%	25.30%	5.42%	19.81%
	@5	12.86%	19.16%	17.55%	47.75%	17.82%	32.30%	12.77%	20.67%	19.31%	7.55%	26.31%	30.00%	15.84%	23.65%	26.29%	6.28%	20.76%
MRR	@1	11.82%	17.02%	21.45%	42.33%	14.80%	29.03%	11.39%	18.72%	17.35%	36.84%	25.04%	19.91%	13.68%	21.94%	25.80%	15.10%	19.96%
	@10	16.72%	21.40%	30.33%	50.09%	19.92%	36.38%	15.95%	25.53%	22.18%	45.49%	31.83%	32.81%	18.00%	29.04%	32.30%	23.32%	25.01%
	@3	14.82%	19.84%	28.09%	47.89%	18.38%	34.38%	14.35%	23.48%	20.65%	43.96%	29.85%	27.43%	16.33%	26.98%	30.56%	20.75%	23.39%
	@5	15.97%	20.50%	29.57%	49.42%	19.27%	35.57%	15.19%	24.80%	21.45%	44.69%	30.93%	30.17%	17.12%	28.09%	31.46%	22.19%	24.30%
NDCG	@1	11.82%	17.03%	21.45%	42.33%	14.80%	29.03%	11.39%	18.72%	17.35%	36.45%	25.04%	19.42%	13.68%	21.94%	25.80%	15.10%	19.96%
	@10	16.92%	22.74%	24.52%	52.93%	21.39%	37.68%	16.39%	25.57%	22.95%	26.68%	31.94%	41.18%	19.62%	28.91%	31.61%	12.68%	24.80%
	@3	13.63%	19.67%	21.94%	48.37%	18.40%	33.54%	13.59%	22.12%	20.02%	32.49%	28.18%	29.88%	16.17%	25.49%	28.52%	12.43%	22.32%
	@5	15.23%	20.74%	22.92%	50.98%	19.88%	35.62%	14.81%	23.82%	21.34%	29.95%	29.85%	34.81%	17.55%	27.01%	29.76%	10.68%	23.48%
Precision	@1	11.82%	17.03%	21.45%	42.33%	14.80%	29.03%	11.39%	18.72%	17.35%	36.84%	25.04%	19.42%	13.68%	21.94%	25.80%	15.10%	19.96%
	@10	3.30%	3.51%	6.91%	7.33%	3.31%	6.01%	3.07%	4.68%	3.65%	19.85%	5.97%	6.91%	3.18%	5.14%	5.93%	6.44%	4.51%
	@3	6.39%	7.92%	14.92%	19.11%	7.80%	14.67%	6.41%	10.39%	8.49%	31.06%	13.16%	12.52%	6.65%	12.06%	13.74%	11.63%	10.15%
	@5	4.98%	5.40%	11.14%	13.13%	5.51%	10.27%	4.74%	7.65%	5.95%	26.32%	9.50%	9.89%	4.88%	8.47%	9.63%	9.26%	7.27%
Recall	@1	9.18%	16.02%	10.96%	40.33%	13.68%	25.38%	9.46%	15.23%	15.70%	3.73%	20.65%	19.42%	12.60%	17.92%	20.32%	3.07%	17.01%
	@10	24.11%	29.78%	30.22%	64.58%	29.15%	48.00%	22.93%	33.90%	29.95%	12.21%	40.79%	69.06%	27.56%	37.81%	39.42%	13.07%	30.53%
	@3	15.40%	21.47%	20.67%	52.42%	20.95%	36.70%	15.13%	24.41%	21.95%	7.75%	30.11%	37.55%	18.10%	28.12%	30.19%	7.09%	22.73%
	@5	19.31%	24.05%	25.12%	58.78%	24.58%	41.75%	18.22%	28.76%	25.22%	9.54%	34.57%	49.43%	21.33%	32.11%	33.74%	9.41%	26.32%

Main - NDCG@10		34.83%	36.90%	20.73%	23.45%	24.52%
MAP	@1	23.00%	5.56%	12.48%	15.19%	10.96%
	@10	30.42%	9.00%	17.15%	20.11%	18.58%
	@3	28.32%	7.89%	15.53%	18.42%	16.29%
	@5	29.47%	8.46%	16.35%	19.33%	17.55%
MRR	@1	27.51%	15.54%	16.00%	17.31%	21.45%
	@10	34.46%	19.90%	21.61%	22.62%	30.33%
	@3	32.58%	19.02%	19.76%	20.91%	28.09%
	@5	33.64%	19.49%	20.73%	21.85%	29.57%
NDCG	@1	27.51%	14.93%	16.00%	17.31%	21.45%
	@10	34.83%	15.22%	20.73%	23.45%	24.52%
	@3	31.18%	16.07%	17.99%	20.25%	21.94%
	@5	32.86%	15.77%	18.92%	21.69%	22.92%
Precision	@1	27.51%	15.34%	16.00%	17.31%	21.45%
	@10	5.97%	6.89%	4.14%	3.97%	6.91%
	@3	14.23%	11.81%	8.86%	8.89%	14.92%
	@5	9.97%	9.57%	6.44%	6.39%	11.14%
Recall	@1	23.00%	5.56%	12.48%	15.19%	10.96%
	@10	43.96%	13.88%	27.22%	31.12%	30.22%
	@3	33.63%	10.21%	18.99%	22.22%	20.87%
	@5	37.98%	11.84%	22.20%	25.87%	25.12%

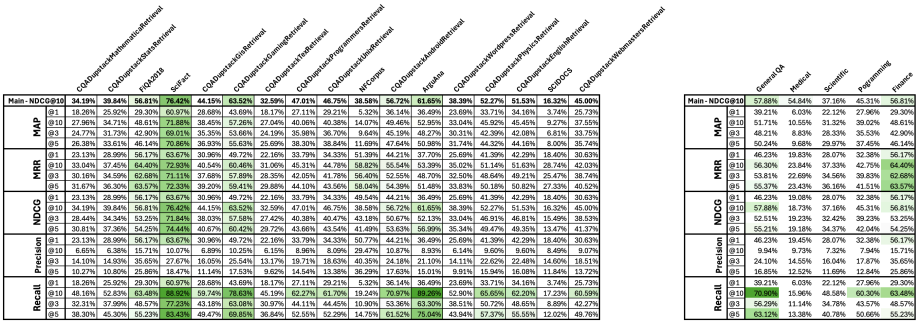
Figure 9: Base DistilBERT Performance



Main - NDCG@10		10.61%	17.18%	14.85%	48.58%	16.35%	29.20%	9.66%	19.02%	16.71%	20.03%	24.38%	44.51%	11.55%	24.77%	20.26%	10.37%	18.19%
MAP	@1	5.20%	11.18%	5.45%	32.39%	9.43%	17.67%	5.27%	11.55%	10.67%	2.21%	14.84%	21.41%	6.20%	14.94%	12.54%	2.59%	10.42%
	@10	8.30%	14.60%	10.37%	43.22%	13.61%	24.83%	7.73%	15.89%	14.16%	5.81%	20.46%	35.96%	9.53%	20.91%	17.21%	5.83%	14.61%
	@3	7.12%	13.39%	8.54%	40.66%	12.23%	22.71%	6.81%	14.57%	12.89%	3.88%	18.55%	31.24%	8.62%	19.20%	15.72%	4.31%	12.80%
	@5	7.69%	13.85%	9.55%	42.15%	12.92%	23.98%	7.29%	15.17%	13.55%	4.74%	19.55%	33.82%	8.99%	20.03%	16.44%	5.10%	13.69%
MRR	@1	6.47%	12.58%	12.65%	33.67%	10.17%	20.82%	6.37%	14.16%	11.85%	28.17%	19.46%	21.76%	6.84%	18.29%	14.84%	12.70%	12.65%
	@10	10.20%	16.49%	18.84%	44.21%	14.63%	27.71%	9.35%	19.01%	15.91%	37.22%	24.67%	36.11%	10.34%	24.56%	19.86%	18.96%	17.56%
	@3	8.89%	15.16%	16.84%	42.06%	13.15%	25.65%	8.29%	17.73%	14.63%	35.14%	22.91%	31.34%	9.40%	22.73%	18.54%	16.85%	15.74%
	@5	9.57%	15.74%	17.85%	43.36%	13.91%	26.83%	8.84%	18.30%	15.31%	36.42%	23.88%	33.99%	9.83%	23.67%	19.19%	18.07%	16.67%
NDCG	@1	6.47%	12.58%	12.65%	33.67%	10.17%	20.82%	6.37%	14.16%	11.85%	26.32%	19.46%	21.41%	6.84%	18.29%	14.84%	12.70%	12.65%
	@10	10.61%	17.18%	14.85%	48.58%	16.35%	29.20%	9.66%	19.02%	16.71%	20.03%	24.38%	44.51%	11.55%	24.77%	20.26%	10.37%	18.19%
	@3	8.30%	14.65%	12.15%	43.92%	13.42%	25.13%	7.82%	16.67%	14.18%	24.20%	21.21%	34.57%	9.60%	21.69%	17.73%	9.93%	14.92%
	@5	9.21%	15.42%	13.23%	46.23%	14.66%	27.13%	8.61%	17.40%	15.27%	22.14%	22.47%	39.24%	10.29%	22.90%	18.71%	8.70%	16.24%
Precision	@1	6.47%	12.58%	12.65%	33.67%	10.17%	20.82%	6.37%	14.16%	11.85%	28.17%	19.46%	21.41%	6.84%	18.29%	14.84%	12.70%	12.65%
	@10	2.10%	2.88%	4.66%	7.13%	2.73%	5.01%	1.85%	3.49%	2.83%	15.51%	4.72%	7.21%	1.94%	4.61%	3.75%	5.33%	3.81%
	@3	4.15%	6.44%	8.54%	18.33%	5.91%	11.52%	3.69%	8.03%	6.22%	24.15%	10.35%	14.75%	4.19%	10.33%	8.41%	9.23%	7.25%
	@5	3.11%	4.42%	6.76%	12.53%	4.29%	8.26%	2.79%	5.48%	4.50%	20.12%	7.47%	11.12%	2.96%	7.32%	5.95%	7.60%	5.48%
Recall	@1	5.20%	11.18%	5.45%	32.39%	9.43%	17.67%	5.27%	11.55%	10.67%	2.21%	14.84%	21.41%	6.20%	14.94%	12.54%	2.59%	10.42%
	@10	15.97%	23.42%	19.82%	63.86%	23.97%	39.36%	13.99%	25.48%	22.87%	9.47%	31.84%	72.12%	17.14%	32.87%	26.58%	10.83%	25.40%
	@3	9.50%	16.07%	11.50%	51.40%	15.79%	28.34%	8.86%	18.27%	16.00%	4.58%	22.67%	44.24%	11.65%	23.89%	19.26%	5.64%	15.80%
	@5	11.82%	18.12%	15.05%	56.91%	18.85%	33.23%	10.83%	20.60%	18.61%	6.58%	26.17%	55.62%	13.36%	27.08%	21.99%	7.73%	19.35%

Main - NDCG@10		24.99%	37.28%	15.43%	16.77%	14.85%
MAP	@1	15.26%	7.32%	8.79%	9.97%	5.45%
	@10	21.30%	11.05%	12.52%	13.98%	10.37%
	@3	19.42%	9.77%	11.24%	12.65%	8.54%
	@5	20.43%	10.35%	11.86%	13.29%	9.55%
MRR	@1	18.01%	15.23%	11.59%	11.64%	12.65%
	@10	24.02%	20.41%	16.10%	15.96%	18.84%
	@3	22.31%	19.07%	14.65%	14.63%	16.64%
	@5	23.24%	19.79%	15.40%	15.30%	17.85%
NDCG	@1	18.01%	14.44%	11.59%	11.64%	12.65%
	@10	24.99%	16.67%	15.43%	16.77%	14.85%
	@3	21.65%	16.40%	13.27%	14.20%	12.15%
	@5	23.17%	16.38%	13.92%	15.20%	13.23%
Precision	@1	18.01%	14.98%	11.59%	11.64%	12.65%
	@10	4.41%	6.11%	3.23%	3.00%	4.66%
	@3	10.05%	10.82%	6.73%	6.51%	8.54%
	@5	7.17%	8.46%	4.94%	4.66%	6.76%
Recall	@1	15.26%	7.32%	8.79%	9.97%	5.45%
	@10	33.35%	18.08%	20.77%	23.34%	19.82%
	@3	24.07%	12.43%	14.10%	16.02%	11.50%
	@5	27.94%	14.43%	16.47%	18.63%	15.05%

Figure 10: Finetuned DistilBERT Performance



Main - NDCG@10		CQADupstackMathematicaRetrieval	CQADupstackStatsRetrieval	FiQA2018	SciFact	CQADupstackGisRetrieval	CQADupstackGamingRetrieval	CQADupstackTexRetrieval	CQADupstackProgrammersRetrieval	CQADupstackUnixRetrieval	NFCorpus	CQADupstackAndroidRetrieval	ArguAna	CQADupstackWordpressRetrieval	CQADupstackPhysicsRetrieval	CQADupstackEnglishRetrieval	SCIDOCS	CQADupstackWebmastersRetrieval
	@10	**34.19%**	**39.84%**	**56.81%**	**76.42%**	**44.15%**	**63.52%**	**32.59%**	**47.01%**	**46.75%**	**38.58%**	**56.72%**	**61.65%**	**38.39%**	**52.27%**	**51.53%**	**16.32%**	**45.00%**
MAP	@1	18.26%	25.92%	29.30%	60.97%	28.68%	43.69%	18.17%	27.11%	29.21%	5.32%	36.14%	36.49%	23.69%	33.71%	34.16%	3.74%	25.73%
	@10	27.96%	34.71%	48.61%	71.88%	38.45%	57.26%	27.04%	40.06%	40.38%	14.07%	49.46%	52.95%	33.04%	45.92%	45.45%	9.27%	37.55%
	@3	24.77%	31.73%	42.90%	69.01%	35.35%	53.66%	24.19%	35.98%	36.70%	9.64%	45.19%	48.27%	30.31%	42.39%	42.08%	6.81%	33.75%
	@5	26.38%	33.61%	46.14%	70.86%	36.93%	55.63%	25.69%	38.30%	38.84%	11.69%	47.64%	50.88%	31.74%	44.32%	44.16%	8.00%	35.74%
MRR	@1	23.13%	28.99%	56.17%	63.67%	30.96%	49.72%	22.16%	33.79%	34.33%	51.39%	44.21%	37.70%	25.69%	41.39%	42.29%	18.40%	30.63%
	@10	33.04%	37.45%	64.40%	72.93%	40.54%	60.46%	31.06%	45.31%	44.78%	58.82%	55.54%	53.39%	35.02%	51.14%	51.63%	28.74%	42.03%
	@3	30.16%	34.59%	62.68%	71.11%	37.68%	57.89%	28.35%	42.05%	41.78%	56.40%	52.55%	48.70%	32.50%	48.64%	49.21%	25.47%	38.74%
	@5	31.67%	36.30%	63.57%	72.33%	39.20%	59.41%	29.86%	44.10%	43.56%	58.04%	54.39%	51.48%	33.83%	50.18%	50.82%	27.33%	40.52%
NDCG	@1	23.13%	28.99%	56.17%	63.67%	30.96%	49.72%	22.16%	33.79%	34.33%	49.54%	44.21%	36.49%	25.69%	41.39%	42.29%	18.40%	30.63%
	@10	34.19%	39.84%	56.81%	76.42%	44.15%	63.52%	32.59%	47.01%	46.75%	38.58%	56.72%	61.65%	38.39%	52.27%	51.53%	16.32%	45.00%
	@3	28.44%	34.34%	53.25%	71.84%	38.03%	57.58%	27.42%	40.38%	40.47%	43.18%	50.67%	52.13%	33.04%	46.91%	46.81%	15.49%	38.53%
	@5	30.81%	37.36%	54.25%	74.44%	40.67%	60.42%	29.72%	43.68%	43.54%	41.49%	53.63%	56.99%	35.34%	49.47%	49.35%	13.47%	41.37%
Precision	@1	23.13%	28.99%	56.17%	63.67%	30.96%	49.72%	22.16%	33.79%	34.33%	50.77%	44.21%	36.49%	25.69%	41.39%	42.29%	18.40%	30.63%
	@10	6.65%	6.38%	15.71%	10.07%	6.89%	10.25%	6.15%	8.96%	8.09%	29.47%	10.87%	8.93%	6.14%	9.60%	9.60%	8.49%	9.07%
	@3	14.10%	14.93%	35.65%	27.67%	16.05%	25.54%	13.17%	19.71%	18.63%	40.35%	24.18%	21.10%	14.11%	22.62%	22.48%	14.60%	18.51%
	@5	10.27%	10.80%	25.86%	18.47%	11.14%	17.53%	9.62%	14.54%	13.38%	36.29%	17.63%	15.01%	9.91%	15.94%	16.08%	11.64%	13.72%
Recall	@1	18.26%	25.92%	29.30%	60.97%	28.68%	43.69%	18.17%	27.11%	29.21%	5.32%	36.14%	36.49%	23.69%	33.71%	34.16%	3.74%	25.73%
	@10	48.16%	52.83%	63.48%	88.92%	59.74%	78.83%	45.19%	62.27%	61.70%	19.24%	70.97%	89.26%	52.90%	65.65%	62.20%	17.23%	60.59%
	@3	32.31%	37.99%	48.57%	77.23%	43.18%	63.08%	30.97%	44.11%	44.45%	10.90%	53.36%	63.30%	38.51%	50.72%	48.65%	8.89%	42.27%
	@5	38.30%	45.30%	55.23%	83.43%	49.47%	69.85%	36.84%	52.55%	52.29%	14.75%	61.52%	75.04%	43.94%	57.37%	55.55%	12.02%	49.76%

Main - NDCG@10		General QA	Medical	Scientific	Programming	Finance
	@10	57.88%	54.84%	37.16%	45.31%	56.81%
MAP	@1	39.21%	6.03%	22.12%	27.96%	29.30%
	@10	51.71%	10.55%	31.32%	39.02%	48.61%
	@3	46.21%	8.83%	28.33%	35.53%	42.90%
	@5	50.24%	9.68%	29.97%	37.45%	46.14%
MRR	@1	46.23%	19.83%	28.07%	32.38%	56.17%
	@10	56.30%	23.84%	37.33%	42.75%	64.40%
	@3	53.81%	22.69%	34.56%	39.83%	62.68%
	@5	55.37%	23.43%	36.16%	41.51%	63.57%
NDCG	@1	46.23%	19.08%	28.07%	32.38%	56.17%
	@10	57.88%	18.73%	37.16%	45.31%	56.81%
	@3	52.61%	19.23%	32.42%	39.23%	53.25%
	@5	55.21%	19.18%	34.37%	42.04%	54.25%
Precision	@1	46.23%	19.45%	28.07%	32.38%	56.17%
	@10	9.94%	9.73%	7.32%	7.94%	15.71%
	@3	24.10%	14.55%	16.04%	17.87%	35.65%
	@5	16.85%	12.52%	11.69%	12.84%	25.86%
Recall	@1	39.21%	6.03%	22.12%	27.96%	29.30%
	@10	76.90%	15.96%	48.58%	60.30%	63.48%
	@3	56.29%	11.14%	34.78%	43.57%	48.57%
	@5	63.12%	13.38%	40.78%	50.66%	55.23%

Figure 11: intfloat/e5-mistral-7b-instruct Performance

A.4 Dataset Metadata

	Domain	Queries	Docs	Average Document Length	Average Query Length	Avg Docs per Query	MTEB Path
FiQA2018	Finance	648	57,638	767	63	2.6	mteb/fiqa
CQADupstackEnglishRetrieval	General QA	1,570	40,221	482	48	2.4	mteb/cqadupstack-english
CQADupstackGamingRetrieval	General QA	1,595	45,301	489	49	1.4	mteb/cqadupstack-gaming
ArguAna	Medical	1,406	8,674	1,029	1,193	1.0	mteb/arguana
NFCorpus	Medical	323	3,633	1,590	22	38.2	mteb/nfcorpus
CQADupstackAndroidRetrieval	Programming	699	22,998	594	52	2.4	mteb/cqadupstack-android
CQADupstackGisRetrieval	Programming	885	37,637	1,012	52	1.3	mteb/cqadupstack-gis
CQADupstackProgrammersRetrieval	Programming	876	32,176	1,056	55	1.9	mteb/cqadupstack-programmers
CQADupstackUnixRetrieval	Programming	1,072	47,382	1,005	50	1.6	mteb/cqadupstack-unix
CQADupstackWebmastersRetrieval	Programming	506	17,405	707	52	2.8	mteb/cqadupstack-webmasters
CQADupstackWordpressRetrieval	Programming	541	48,605	1,123	49	1.4	mteb/cqadupstack-wordpress
CQADupstackMathematicaRetrieval	Scientific	804	16,705	1,153	49	1.7	mteb/cqadupstack-mathematica
CQADupstackPhysicsRetrieval	Scientific	1,039	38,316	819	53	1.9	mteb/cqadupstack-physics
CQADupstackStatsRetrieval	Scientific	652	42,269	1,055	56	1.4	mteb/cqadupstack-stats
CQADupstackTexRetrieval	Scientific	2,906	68,184	1,297	47	1.8	mteb/cqadupstack-tex
SCIDOCS	Scientific	1,000	25,657	1,203	72	4.9	mteb/scidocs
SciFact	Scientific	300	5,183	1,498	90	1.1	mteb/scifact

Figure 12: Dataset Source, Attibuted Domain, and Metadata

A.5 Finetuning Triplet Sample

Scientific Domain Sample:

Query: *Which hormone abnormalities are characteristic to Pendred syndrome?*

Positive Example:

A coherent organization of differentiation proteins is required to maintain an appropriate thyroid function in the Pendred thyroid. CONTEXT: Pendred syndrome is caused by mutations in the gene coding for pendrin, an apical Cl-/I- exchanger. OBJECTIVE: To analyze intrathyroidal compensatory mechanisms when pendrin is lacking, we investigated the thyroid of a patient with Pendred syndrome. The expression of proteins involved in thyroid hormone synthesis, markers of oxidative stress (OS), cell proliferation, apoptosis, and antioxidant enzymes were analyzed. RESULTS: Three morphological zones were identified: nearly normal follicles with iodine-rich thyroglobulin .....

Negative Example:

A meta-analysis of prospective randomized controlled trials evaluating endovascular therapies for acute ischemic stroke. INTRODUCTION: A recent randomized controlled trial (RCT), the Multicenter Randomized CLinical trial of Endovascular treatment for Acute ischemic stroke in the Netherlands (MR CLEAN), demonstrated better outcomes with endovascular treatment compared with medical therapy for acute ischemic stroke (AIS). However, previous trials have provided mixed results regarding the efficacy of endovascular treatment for AIS. A meta-analysis of all available trial data was performed to summarize the available evidence. METHODS: A literature search was performed to identify all prospective RCTs comparing endovascular therapies with medical .....

General QA Domain Sample:

Query: *Why am I always rejected by the men I am interested in?*

Positive Example:

I am a man, just not competitively and I will tell you why I reject most of the time. I think this will actually mean something because I am homeless. I reject because I watch a lot of porn. I am 40, no kids. No money. I reject because I have no money. No car. Can't take you someplace anywhere. I would like to. Because I can talk to you. I don't have ANY gold. That's not me rejecting you. I know stuff other than rejection, projection, protection. How am I going to take care of myself? .....

Negative Example:

The use of microwave ovens to cook food has been a major source of debate for decades. While some people believe that microwaving food is dangerous and can cause health problems, others believe that microwaves are a safe and convenient way to quickly prepare meals. While both sides can agree that microwaving food does not provide the same nutritional value as traditional cooking methods, there is no scientific evidence that food prepared in a microwave oven is bad for your health. To understand the potential health risks associated with microwaving food, it is important to understand how microwaves work. Microwave ovens use high frequency radio waves to heat food. These waves penetrate food and cause the water molecules in it to vibrate, generating heat and cooking the food. Microwave ovens are designed to contain these waves, and the radiation emitted from them is non-ionizing, meaning it does not have enough energy to cause chemical changes in the food or the environment. .....

A.6 Alternative Evaluation Metrics

Supplemental Metric Definitions:

- **NDCG@k**: Normalized Discounted Cumulative Gain, measuring ranking quality by relevance and position.
- **MAP@k**: Mean Average Precision at rank k, capturing precision across top-k results.
- **MRR@k**: Mean Reciprocal Rank at k, reflecting the rank of the first relevant document.

Evaluation Framework: All models were evaluated on independent hardware using evaluation scripts sourced from the MTEB leaderboard framework [7], ensuring consistency and comparability across diverse datasets and domains.

A.7 Supplementary Operational Details

Caching Mechanism: LLM embeddings were precomputed for efficiency. Custom vector caching was created to efficiently retrieve precomputed vectors for evaluation in a "compute once, evaluate many" fashion.

Compute Resources: All experiments were conducted on dual Nvidia RTX A6000 GPUs, with total runtime for evaluations estimated at 48 hours for both LLM Encoder and Decoders (Mistral and e5), and estimated at 2 and a half hours for all SLM Encoders (BERT-based models).