CHROMA TECHNICAL REPORT

May 29, 2024

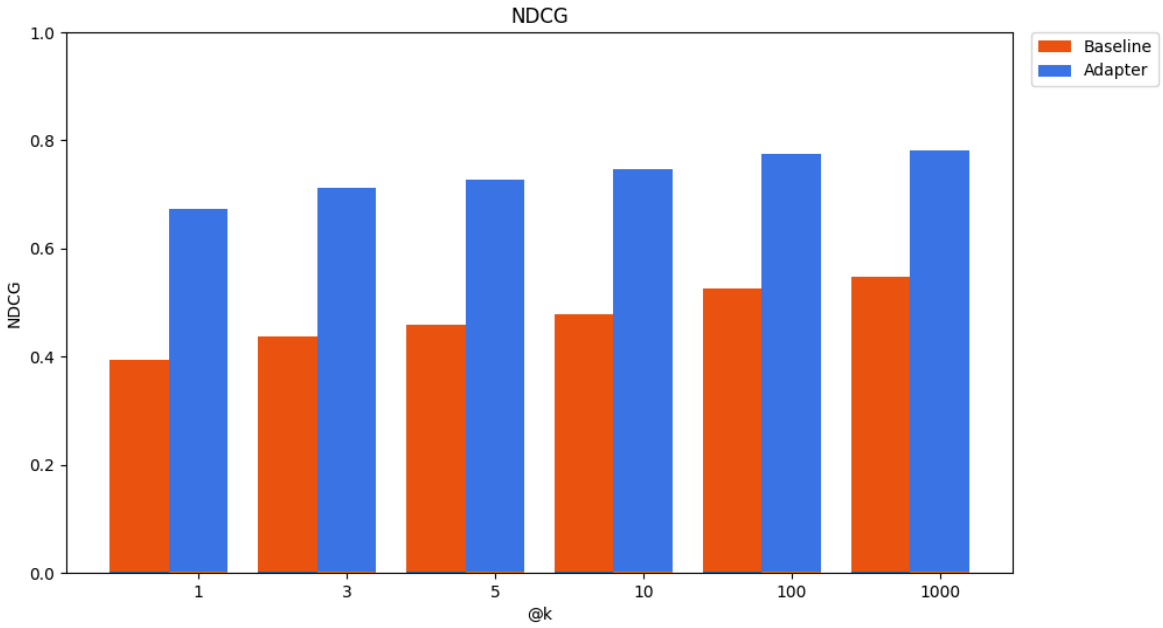
Embedding Adapters

Suvansh Sanjeev Researcher in Residence - Chroma

Anton Troynikov Cofounder - Chroma

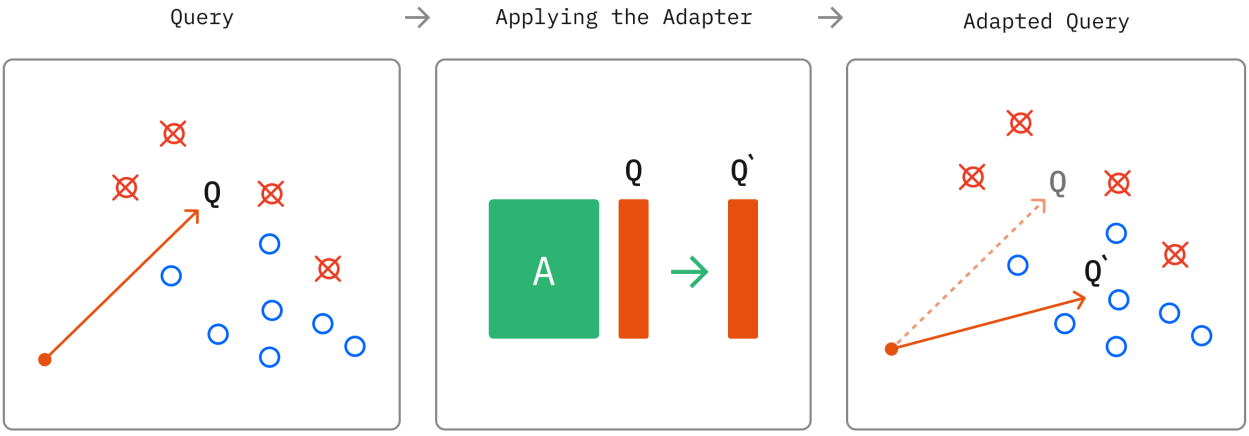
Retrieval accuracy is an important determinant of AI application performance. However, many approaches to improving retrieval accuracy require large labeled corpora, which are often not available to application developers. Additionally, many of these approaches require re-computing the entire set of embeddings.

In this work we demonstrate that applying a linear transform, trained from relatively few labeled datapoints, to just the query embedding, produces a



Normalized discounted cumulative gain (NDCG), CQADupstackEnglishRetrieval. A simple query-only linear adapter produces an up to 70% improvement.

The idea of applying a linear transform to embeddings to improve retrieval performance is not new, and has been explored in various forms. The purpose of this technical report is to more rigorously evaluate the effectiveness of this approach in practical retrieval applications.



The basic concept is fairly straightforward. Given a set of query embeddings, and a set of relevant and irrelevant document embeddings for each query, we can learn a transform which squeezes and rotates the vector space, mapping them to a new

transform as an 'adapter', since we apply it after the output of the embedding model itself.

The simplest such adapter is a linear transform, which can be implemented efficiently as a matrix multiplication. In this work we find that training an adapter applied to **just the query embedding**, from relatively few labeled query-document pairs (as few as 1,500), produces an improvement in retrieval accuracy over the pre-trained embedding model alone of **up to 70%**. These labels can be collected from human feedback, for example from application usage. Further, we show that negative examples generated as random samples from documents not labeled as relevant to a particular query, is sufficient to achieve this improvement, considerably reducing labeling cost.

While we are confident that this simple approach is likely to improve retrieval accuracy in many practical settings, particularly in retreival for AI applications involving relatively small (~10,000) collections of embeddings, our analysis is not exhaustive. We show that the magnitude of gains varies significantly, depending on the training data. Additionally, we only attempt the simplest approach of training each parameter of the linear adapter independently, rather than constraining the resulting transform to a particular rank or group. This is left to future work.

We provide all code to replicate our results as a GitHub repo, along with useful utilities for similar experiments.

Our in-depth technical report continues below. If you find our work useful, please consider citing us:

plaintext Copy Code

```
@techreport{sanjeev2024embedding,
  title = {Embedding Adapters},
  author = {Sanjeev, Suvansh and Troynikov, Anton},
  year = {2024},
  month = {May},
  institution = {Chroma},
  url = {https://research.trychroma.com/embedding-adapters},
  note = {Accessed on July 8, 2024}
}
```

Introduction

Learning is compression. By detecting patterns in the world, we can represent complex information in a more compact and efficient manner. This idea is at the core of representational learning, which focuses on learning representations of data that capture its underlying structure and semantics. In particular, embeddings seek to compress high-dimensional data, such as words or images, into dense, low-dimensional vectors. These embedded representations are learned through neural networks, optimized to capture the essential features and relationships within the data.

Once embeddings have been learned, they can be adapted and fine-tuned for various downstream tasks, allowing for more specialized and efficient performance. This adaptability is one of the key strengths of embeddings, as they provide a foundation for transfer learning, where knowledge gained from one task can be applied to another. By reducing the dimensionality of the data and warm-starting with useful representations, embeddings not only make downstream learning processes more computationally tractable, but also enable the discovery of hidden connections and similarities that might be obscured in the original representation. By leveraging pre-trained embeddings and fine-tuning them with task-specific data, we can create models that are more accurate and require less training time compared to learning from scratch. This approach has been particularly successful in natural language processing tasks, such as sentiment analysis, named entity recognition, and machine translation, where pre-trained word embeddings have significantly improved performance.

One promising application of embeddings is in retrieval-augmented generation (RAG), a framework that combines the strengths of retrieval-based and generative models. In this approach, embeddings are used to retrieve relevant information from a large knowledge base, which is then used to augment the input to an LLM. By providing the generative model with additional context and information, retrieval-augmented

embeddings used to represent the query and the knowledge base documents, as the effectiveness of the retrieval directly impacts the performance of the generative model.

In recent years, there has been a growing interest in using human feedback to align and improve machine learning systems with user intent. One promising but under-explored application of this approach is in RAG. By incorporating human feedback into the retrieval process, we can learn to prioritize and select the most relevant information based on user preferences and needs. This personalization of embeddings has the potential to greatly improve the quality and usefulness of generated content, making it more tailored to individual users. In this work, we will explore cost-effective and readily deployable techniques for using annotated positives to personalize retrieval-augmented generation systems by creating more effective embeddings.

Contributions

We present the following:

- Query-only linear adapters. In leaving the document embeddings unchanged, these adapters lower the cost of retrieval adaptation by obviating the need to re-embed all of the documents by applying the adapter to all of the documents in the RAG system's corpus. Instead, the adaptation occurs on the query alone, which is a negligible overhead at query time.
- Experiments comparing various types of linear adapters to finetuned embeddings. We find that the query-only linear adapters are competitive with the more computationally onerous linear adapters as well as finetuning.
- The codebase for our adapters, including scripts used to run retrieval adaptation experiments and hyperparameter sweeps on various datasets, and LaunchKit, the lightweight library we use to launch them. LaunchKit is stripped out from RLKit from the Robotics and AI Lab at UC Berkeley.

Related Work

Linear adapters on embeddings have been explored before. The **Customizing Embeddings** Jupyter Notebook by Openai demonstrates the process of training what we term here the joint linear adapter, where both items being compared via the embeddings are mapped using the same adapter. They also explore random negative sampling for synthetic negatives, which we employ and ablate in this work.

In their paper **"Improving Text Embeddings with Large Language Models"**, Wang et al. demonstrate the use of LLM-generated synthetic documents to finetune embeddings for retrieval tasks and evaluate on the Massive Text Embedding Benchmark (MTEB) retrieval benchmarks.

The **SentenceTransformers** package provides pre-trained embedding models and a framework to finetune them for downstream tasks. They explore various related approaches, notably including Augmented SBERT, which uses cross-encoders (reranking models) to augment and annotate data for use in training bi-encoders (embedding models for sentence pair comparison tasks). Rerankers are a high-performance, high-cost approach to domain-specific adaptation.

Method

Over the course of this project, we investigated the following approaches to adapting vector embeddings to annotated query-document pairs to improve retrieval performance:

- Linear adapters on the embeddings
  - Joint: both queries and documents are mapped with the same linear adapter.

    - Query-first: we learn a joint linear adapter, warm-starting with the query-only adapter weights
    - Separate: queries and documents are mapped with separate linear adapters.
- Fine-tuned embedding models

We focus much of our attention on lightweight and readily deployable approaches. We support the use of both synthetic data and reranker models in our released code, but run only preliminary experiments with them and leave further exploration and tuning to future work.

Experimental Setup

We use the pre-trained all-MiniLM-L6-v2 embedding model from Sentence-Transformers as the baseline model in our experiments, atop which we train adapters and perform fine-tuning. The Massive Text Embedding Benchmark (MTEB) consists of benchmarks for several embedding tasks, including retrieval. We leverage the existing MTEB infrastructure for benchmark evaluation in our experiments, tracking the following retrieval performance metrics:

- Mean average precision (MAP)
- Mean reciprocal rank (MRR)
- Normalized discounted cumulative gain (NDCG)
- Precision
- Recall

Our hyperparameter sweeping and experiment-launching infrastructure is adapted from **RLKit** and released as **LaunchKit**, a separate lightweight repository. Usage is documented in the repository README. Launchkit saves experiment logs in a manner compatible with **VisKit**, a lightweight visualization script with extensive features, which was used to inspect and visualize results of hyperparameter sweeps.

We investigate the use of random negative sampling to augment our data with both negative and positive documents for each query, and running data ablations to gauge the effect of varying the quantity of training data on retrieval performance. Finally, we

Hyperparameters

Below is a guideline of the hyperparameters and values we sweep over. Note that some iterative refinement was performed, so we did not necessarily run every combination in the cartesian product of these sets of hyperparameters.

Hyperparameter	Values	Notes
Learning rate	1e-5, 3e-5, 1e-4, 3e-4, …, 1e-2	
Negative Sampling	True, False	
Loss Function	Mean Squared Error (MSE), Binary Cross-Entropy (BCE), Triplet	
Triplet Loss Margin	0.3, 1, 3	Only for triplet loss
Augmentation Threshold	5	How many negative examples to fill up to with negative sampling
Adapter Output Dimension	384	Equal to embedding dimension

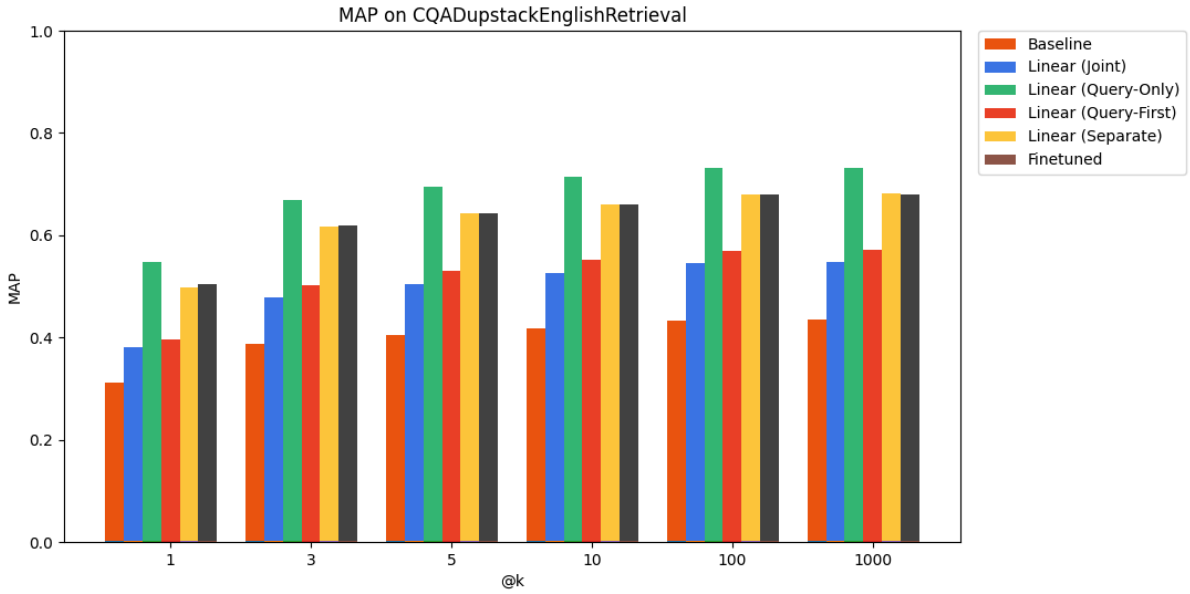
Results

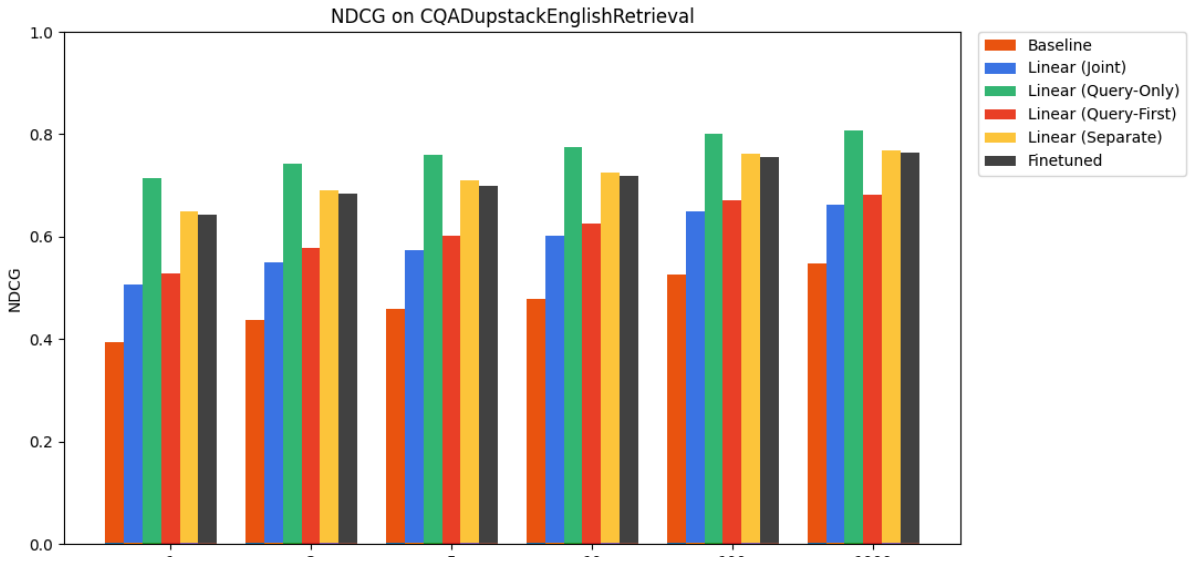
In this section, we review the main results and a few ablations that allow us to better

(NDCG) were selected as the summary results owing to their broader coverage and rank-weighting, with graphs for the remaining metrics available in the repository.

Main Results - Linear Adapters and Fine-tuning

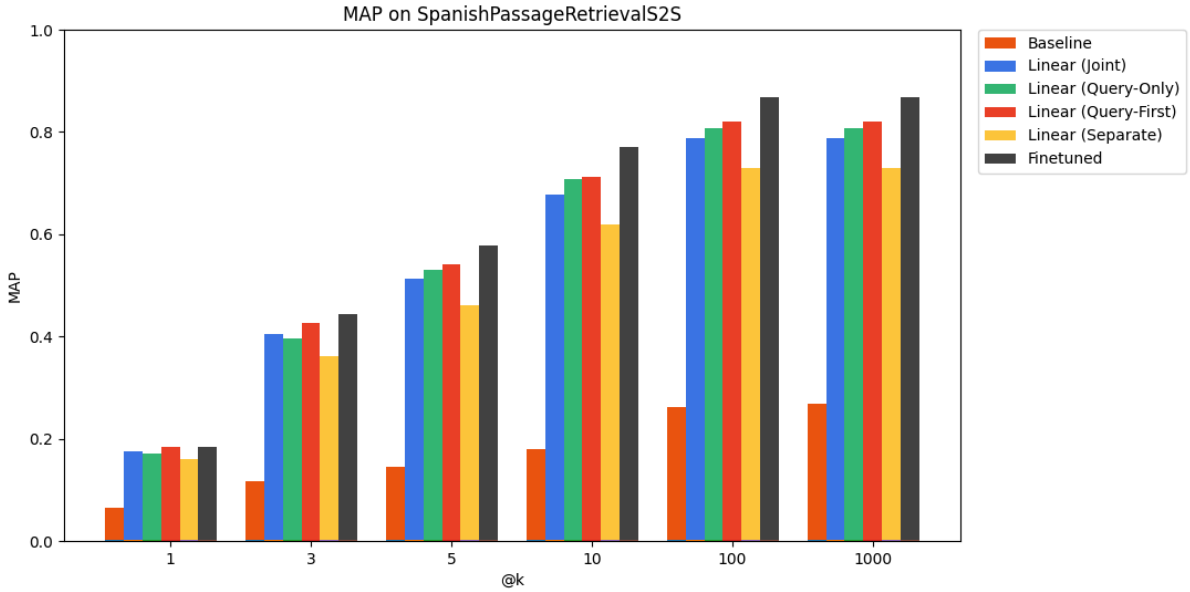
Our results for the linear adapters and finetuning are shown below. All tested approaches outperform the baseline of the pre-trained embedding model.

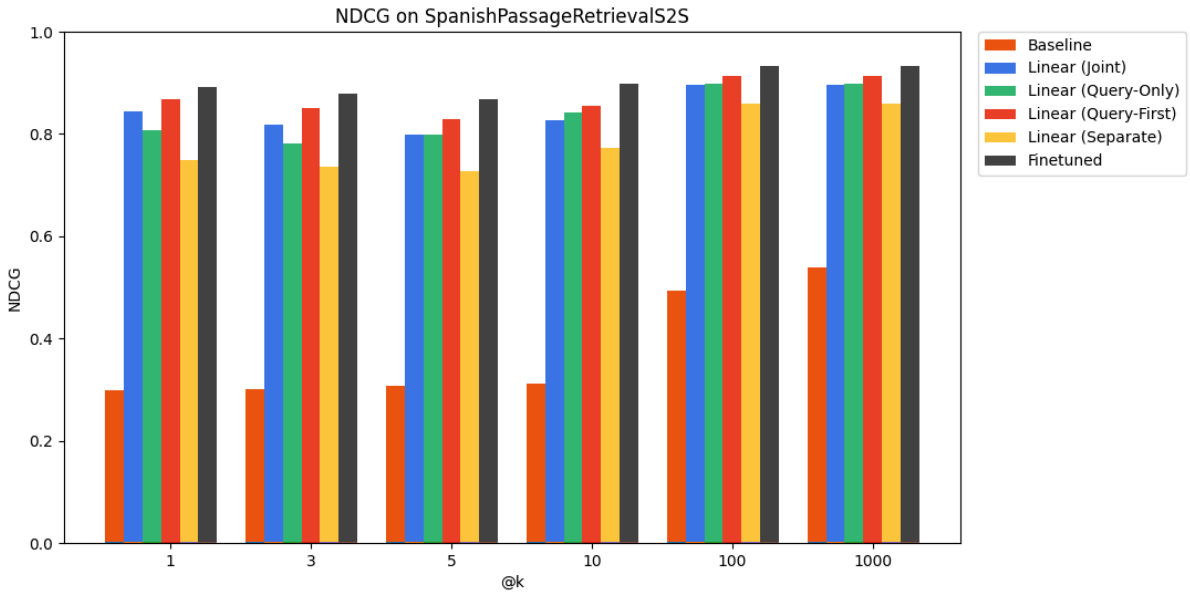




On CQADupstackEnglishRetrieval, we find that the query-only linear adapter performs the best, even surpassing the performance of finetuning.

The embedding model we use, all-MiniLM-L6-v2, is trained on English text and consequently performs relatively poorly on SpanishPassageRetrievalS2S, a Spanish language benchmark.

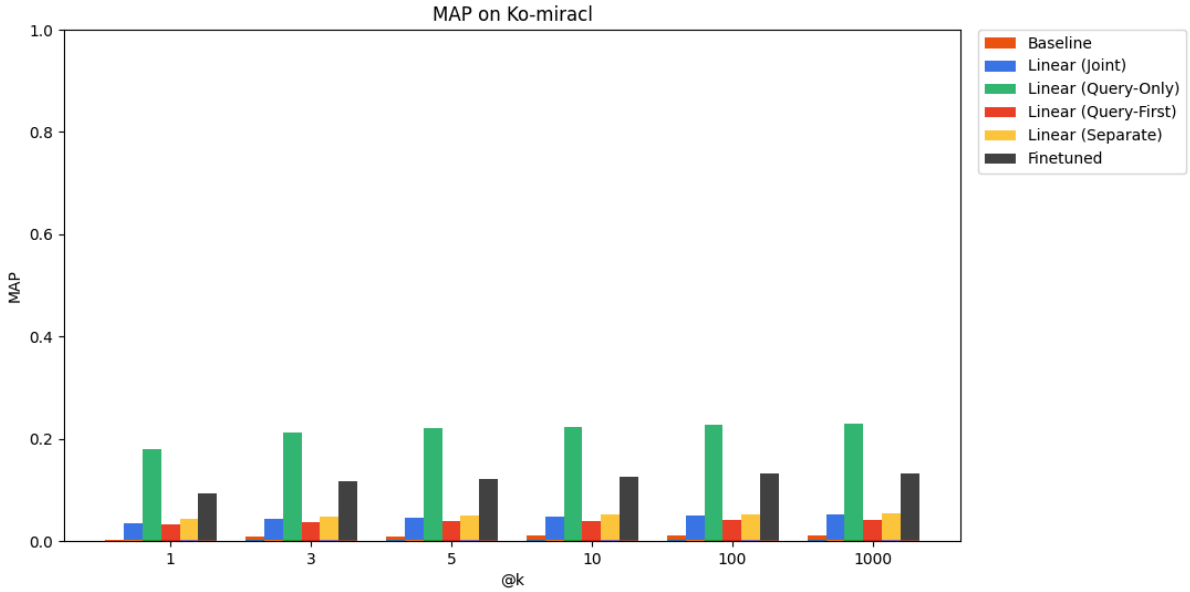


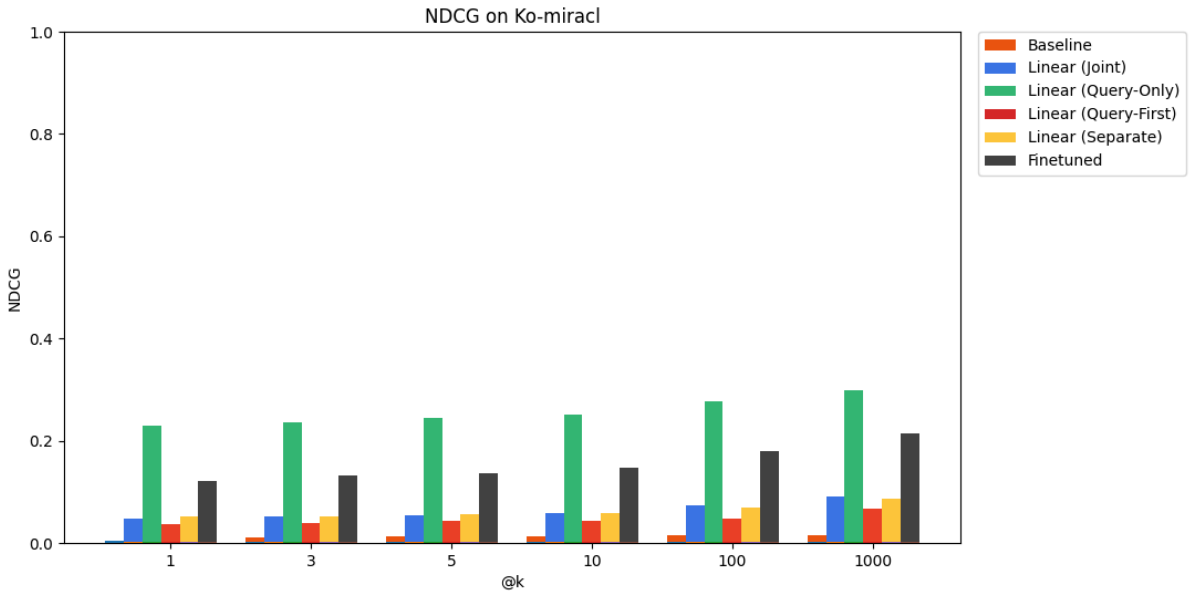


However, we see that a linear adapter is sufficient to yield respectable

competitive with the other approaches. We suspect that due to similarities in the structure of the two languages, their learned representations have a similar structure, up to a linear transformation, which the linear adapter can learn.

Finally, on Ko-miracl, a Korean language benchmark, the language gap proves too difficult to bridge through our adaptation approaches.



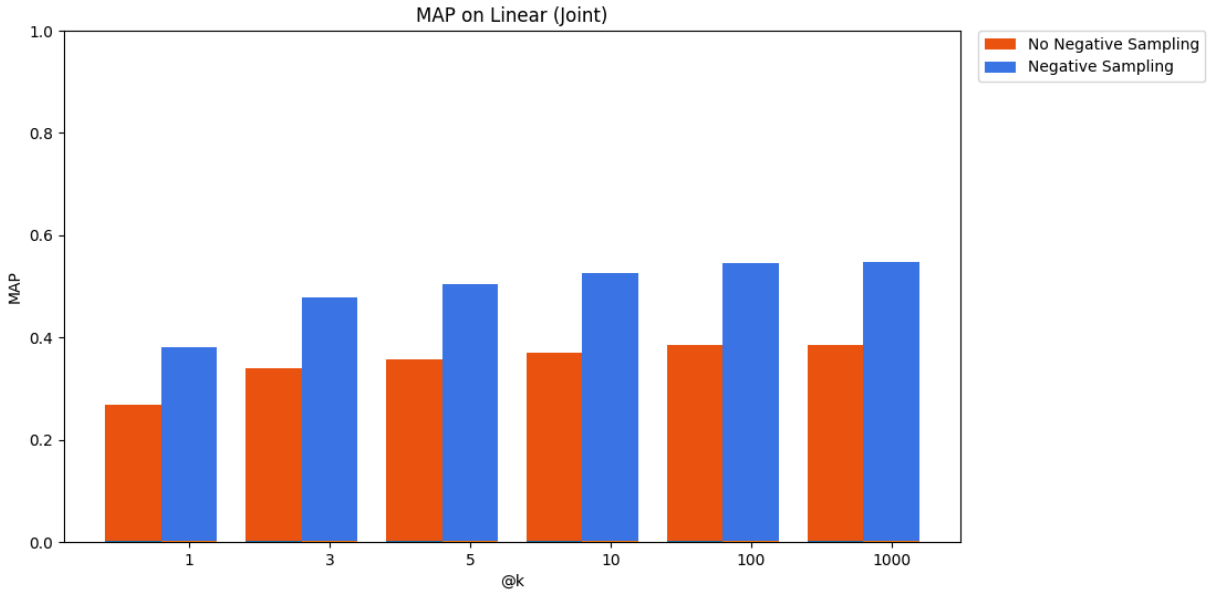


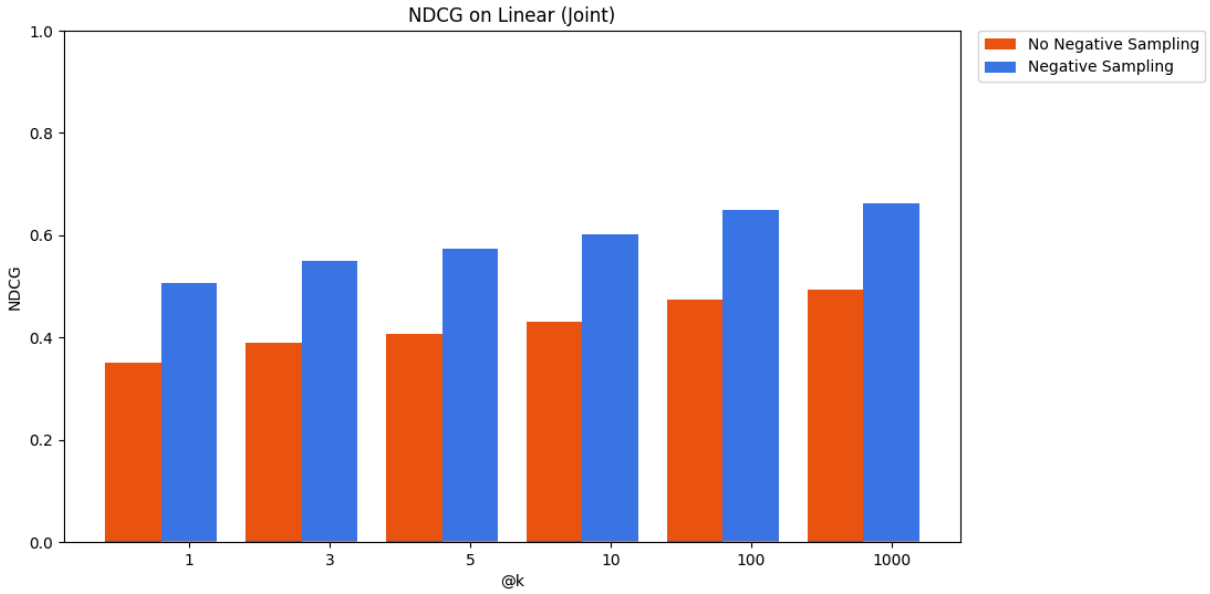
However, the improvement relative to baseline is still considerable, with the query-

These experiments serve to demonstrate the potential of linear adapters, as they seem to be **competitive with finetuning, while being much more lightweight and computationally efficient**, for many applications. We focus our remaining experiments on further investigating linear adapters under various ablations.

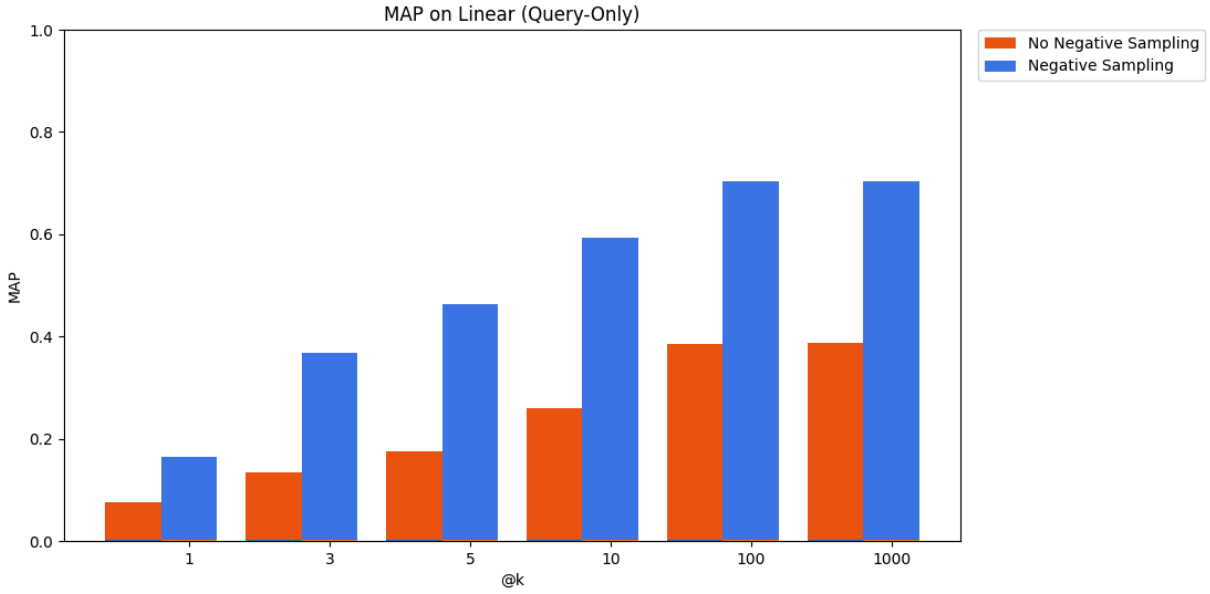
Negative Sampling

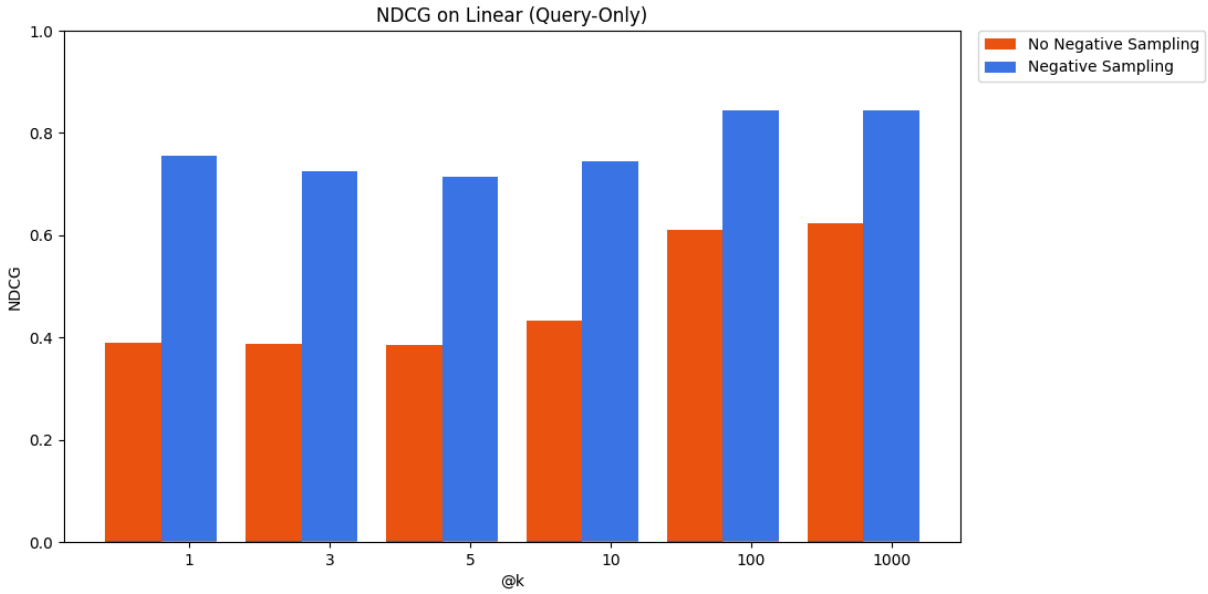
We present two types of linear adapters on two different benchmarks to demonstrate the effect of negative sampling. The remaining graphs can be found in the repository.





CQADupstackEnglishRetrieval - Joint Adapter



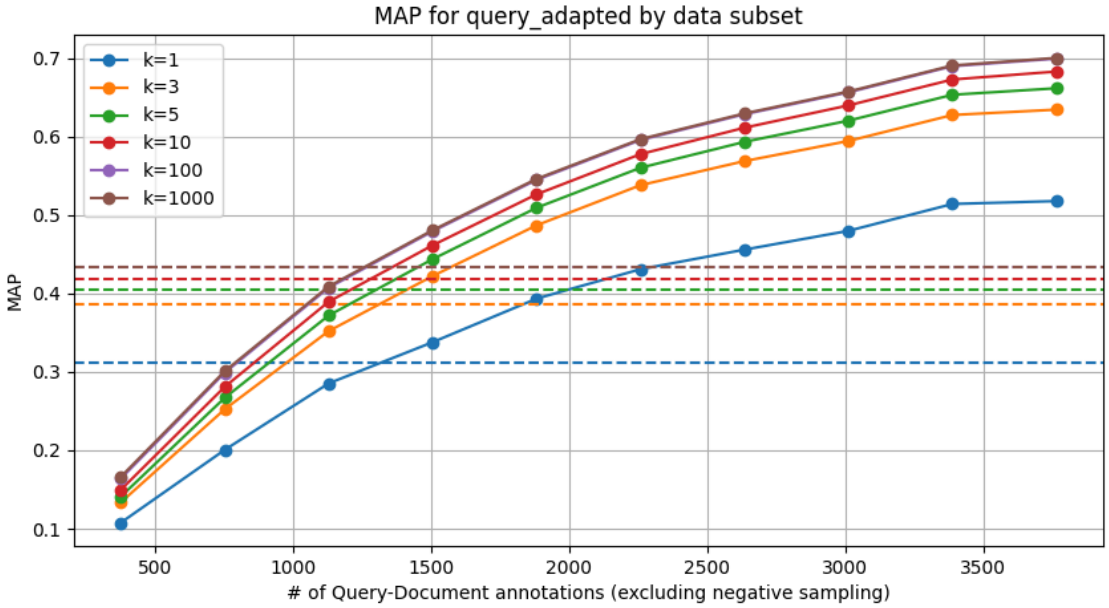


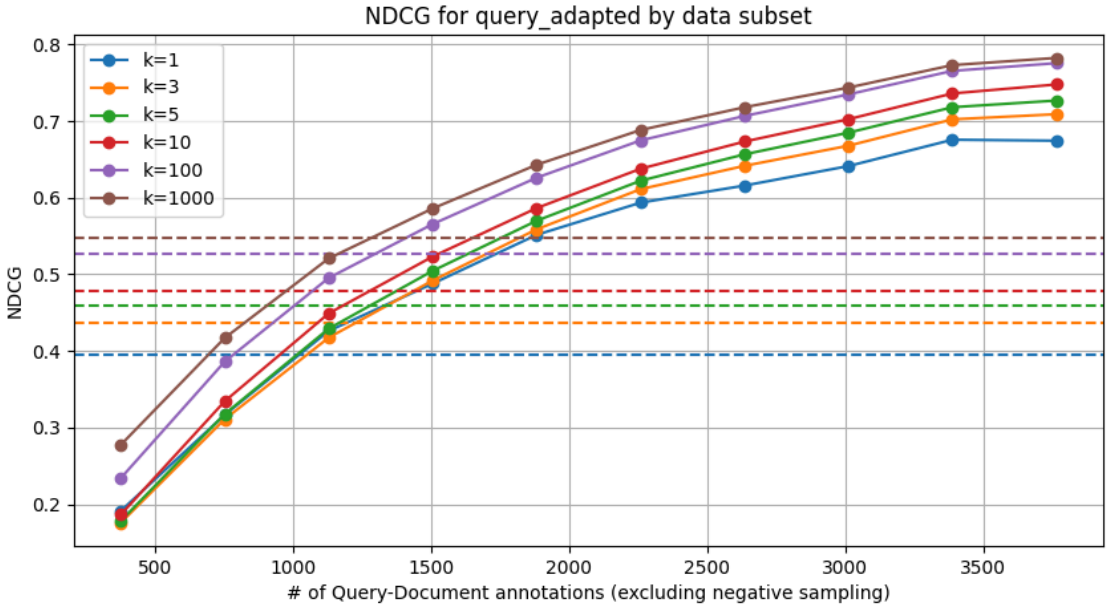
SpanishPassageRetrievalS2S - Query Only Adapter

The result is definitive and has long been established by previous work – **negative sampling, by sampling randomly from unrelated documents, is a cheap way to get a sizable benefit on retrieval performance**.

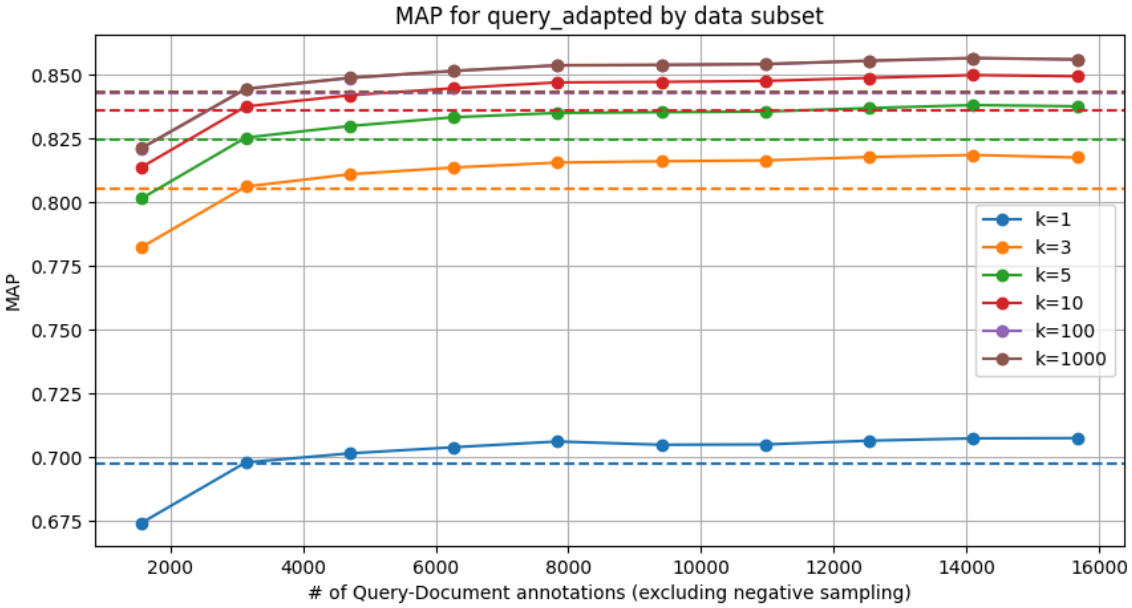
Data Ablation

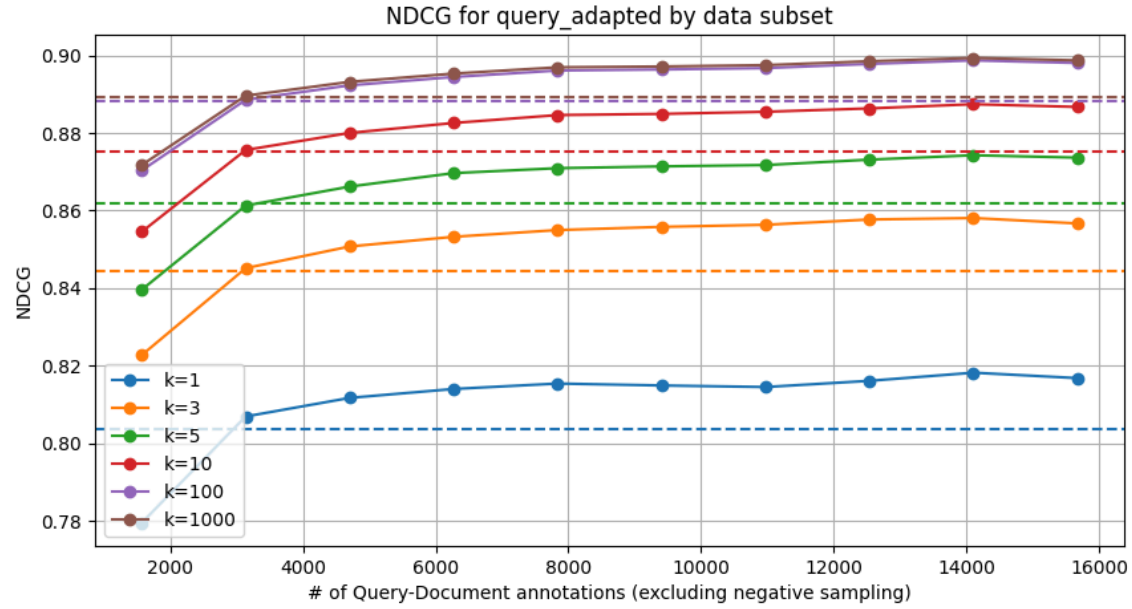
We examine the effect of varying the amount of training data on retrieval performance metrics. The dashed lines indicate the performance of the pre-trained embedding model baseline on the entire dataset.





CQADupstackEnglishRetrieval – Linear (Query-Only)





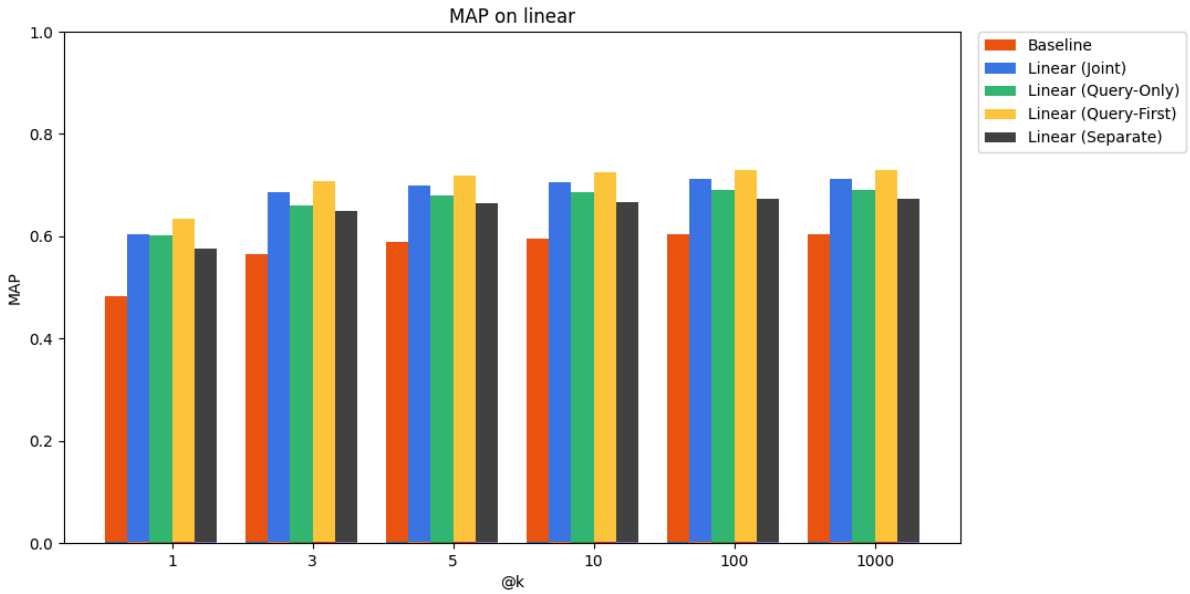
QuoraRetrieval – Linear (Query-Only)

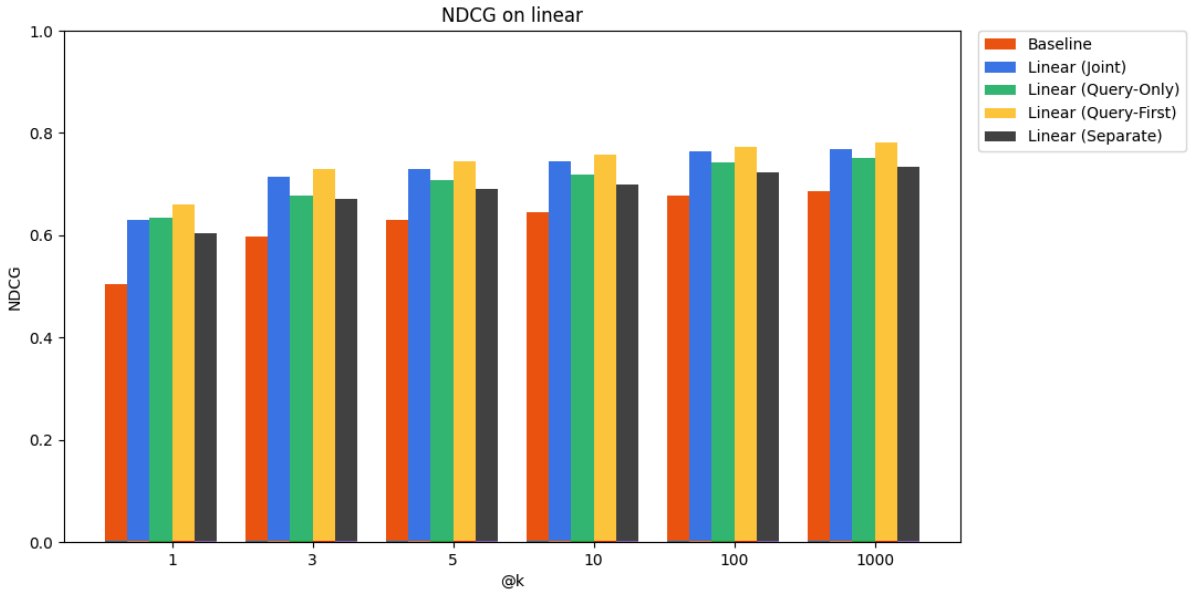
We can see that breakeven is about 20-35% of the data. The subsets we trained on were randomly sampled, so it is likely that greater data efficiency can be obtained through selective sampling. Additionally, noting the scale of the y-axes, we note that the saturation curve looks vastly different between the two benchmarks used here. For training embedding adapters, properties of the particular dataset in question are likely to greatly influence the returns on using a greater proportion of the data.

Our intuition is that datasets with greater heterogeneity yield steeper curves and merit training on a greater proportion of data, though it's not clear how this notion of heterogeneity can be assessed beforehand. Relating these curves to easily-measurable properties of the data is possibly an interesting direction for future work. **Recent results** using gzip as a proxy for measuring the complexity of training data for LLMs may be a useful starting point.

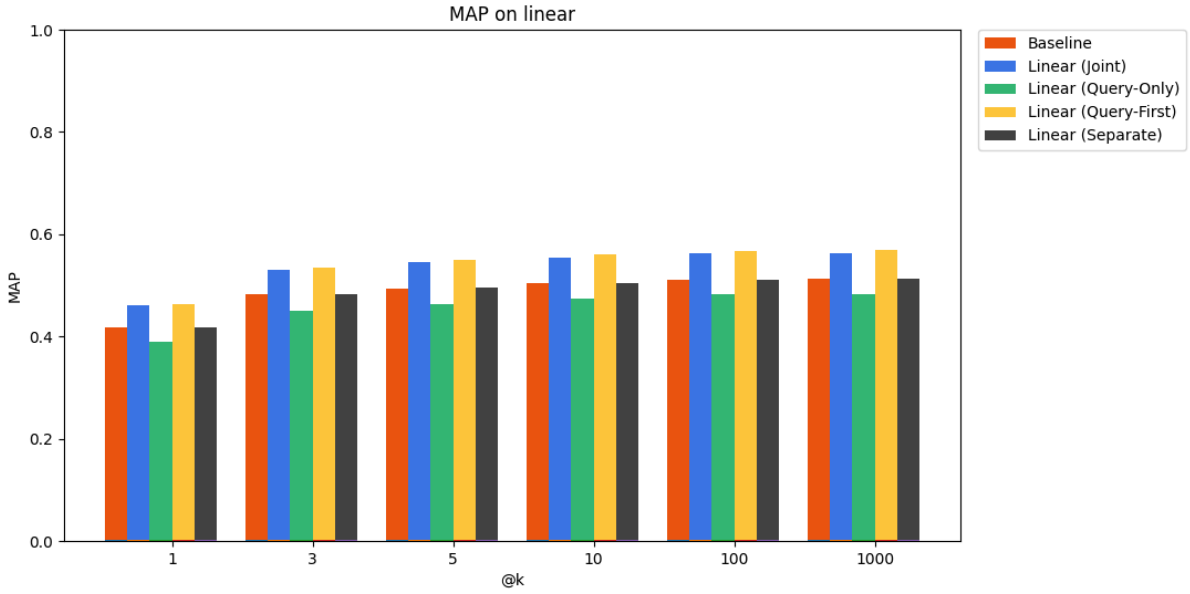
Holdout Performance

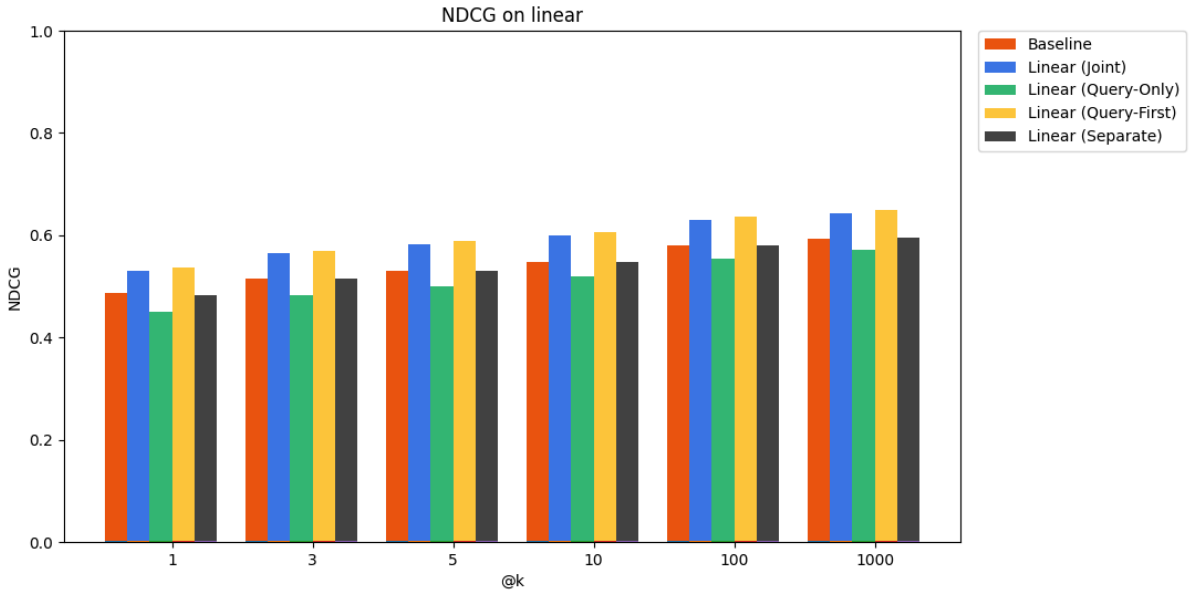
We select from benchmarks which contain multiple data splits, evaluating the retrieval performance of linear adapters on held out data after being trained only on the training split.




NDCG on linear
NDCG
@k
Baseline
Linear (Joint)
Linear (Query-Only)
Linear (Query-First)
Linear (Separate)

SciFact


MAP on linear
MAP
@k
Baseline
Linear (Joint)
Linear (Query-Only)
Linear (Query-First)
Linear (Separate)



Quora-PL

As expected, the performance gains are more modest on the held-out data. In particular, it seems the query-only adapter is at a slight disadvantage compared to the joint linear adapter here. However, it still beats the pre-trained embedding model baseline, demonstrating promise for real-world deployment.

Discussion

We evaluated several retrieval performance metrics at a range of thresholds on various benchmarks, sweeping across numerous hyperparameters to obtain the best performance possible. In doing so, we found that a couple patterns emerged: Binary cross-entropy was dominated by mean squared error, but triplet loss and mean squared error shone in different settings. For the linear adapters, 0.003 was frequently, though not always, the best choice of learning rate. The query-only linear adapter was consistently viable, which has exciting implications for personalized

The scalability of the query-only linear adapter opens up exciting possibilities for real-world applications. Given its computational efficiency, it would be feasible to deploy this approach in a live setting, where user-specific query adapters could be trained on-the-fly. We found in the data ablation section that the rate of improvement as a proportion of data can vary substantially, and we speculated that this is likely mediated by some notion of heterogeneity. Fortunately, the query-only linear adapter obviates the need to re-embed documents, making it possible to perform adaptation at a very granular level, even with a limited compute budget. By personalizing the retrieval process at the individual user level, we may be able to observe significant improvements in online metrics and user feedback. **This granular adaptation could potentially yield substantial gains, even with limited training data, due to the increased homogeneity within a single user's query distribution**. Implementing and evaluating this approach in a real-world scenario would provide valuable insights into the practical benefits of personalized retrieval-augmented generation.

In addition to the query-only linear adapter, we also explored in our research the possibility of training a reranker to improve retrieval performance. While this approach has the potential to capture more complex relationships between queries and documents, our initial experiments yielded unstable results. We hypothesize that this instability may be due to the limited quantity of data available for training the reranker. As reranking models typically require a substantial amount of labeled data to learn effective ranking functions, the scarcity of data in our experiments likely hindered the reranker's performance. Future work could investigate the use of larger datasets or more data-efficient reranking algorithms to overcome this limitation.

Another avenue we briefly explored was the use of synthetic data to augment our training set. By generating synthetic documents to serve as positive examples for queries, we hoped to augment the data with positives in addition to the negative sampling. However, our initial attempts did not yield significant improvements in retrieval performance. This could be due to the challenges in generating realistic synthetic data that remains in-distribution for the dataset. Due to time constraints, we were unable to fully explore this approach, but we believe that further research into effective methods for generating and utilizing synthetic data could potentially enhance the adaptation process, particularly with the promising results by Wang et al on the use of synthetic data for improving embeddings.

To further enhance the expressiveness of the adaptation process, exploring neural

patterns and relationships between user preferences and the retrieval process. This increased flexibility could potentially lead to better personalization and improved retrieval performance. In our experiments, we saw that the separate linear adapter for queries and documents performed relatively poorly in most settings we tested despite having the greatest representational capacity. As its representation space is a superset of that of all other linear adapters we trained, this underperformance points to a training or data issue, and we would be hesitant to write off this approach entirely based on these results.

Therefore, it is important to consider the relationship between the adapter's representational capacity and the dataset size. Intuitively, a more expressive adapter, such as a neural network, would allow for a more fine-grained representation of user preferences, but it may also require more training data to effectively learn the parameters. On the other hand, in settings with limited data, decreasing the representational capacity by learning a rank-constrained weight matrix rather than a dense one, as in our linear adapters, would reduce the risk of overfitting. Conducting experiments to evaluate the interaction between representational capacity and dataset size could provide valuable insights into the optimal configuration for different scenarios. This spectrum of representational capacity offers a range of options to balance adapter expressivity with available data and computational resources.

As we continue to explore and refine these techniques, we move closer to the goal of creating truly personalized and effective retrieval-augmented generation systems that can better serve the needs and preferences of individual users.

References

- MTEB
- Sentence-Transformers