Mixture-of-LoRA-Experts for Continual Learning in Generative Retrieval

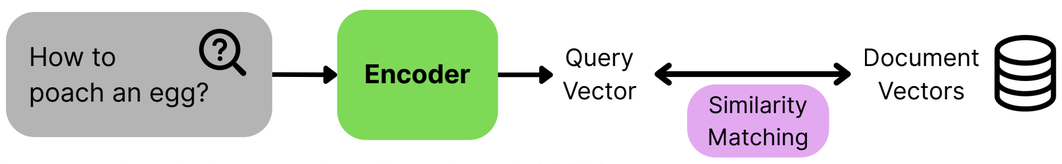
CS6101 - 02: Benjamin Chek[1], Choong Kai Zhe[1], Zak Tng[1]

[1]National University of Singapore

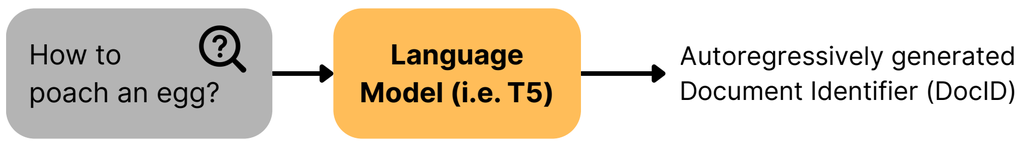
Motivation

Retrieval Augmented Generation (RAG) combines **LLMs** with **information retrieval methods** to enhance factual grounding, and adapt models to new knowledge with less computational expense. [1]

Traditional Retrieval relies on matching to find relevant documents. In sparse retrieval (i.e. BM25), queries are matched to documents by keywords. Dense retrieval methods encode queries and documents into an embedding space to match based on semantic meaning.



Generative Information Retrieval (GIR) is a paradigm that uses a model to generate relevant document identifiers based on the user query.



- A document is "indexed" by training the model to associate documents with a DocID. Document information is now stored in the model weights. A document is retrieved by autoregressively generating the DocID it thinks is the answer.

There are **two fundamental challenges of GIR**: [6]

1. **Catastrophic forgetting** - finetuning on documents B will cause it to forget documents A.
2. **Updating the model** with new documents requires **full retraining**.

MixLoRA-DSI [5] **aims to solve this.** LoRAs prevent the need for full retraining, and adding LoRAs help with adding new documents.

Introduction

MixLoRA-DSI's core idea:

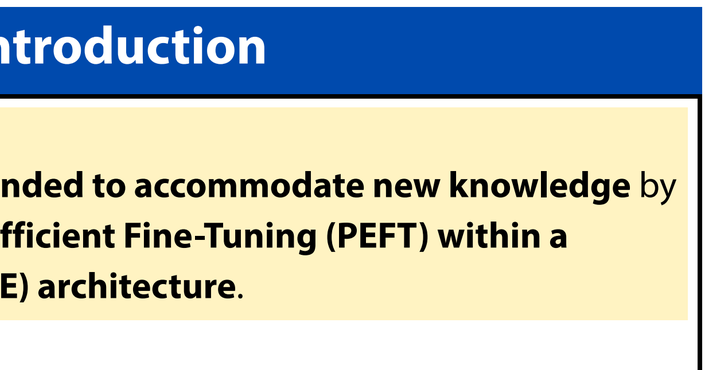
- A DSI model can be **expanded to accommodate new knowledge** by **leveraging Parameter-Efficient Fine-Tuning (PEFT) within a Mixture-of-Experts (MoE) architecture**.

Definitions:

- **DSI (Differentiable Search Index)** [7] refers to the language model, whose weights contain the "index" of documents.
- **Mixture-of-Experts (MoE)** [4] is a pre-training architecture that improves efficiency by routing each input through the top-k specialised sections of the feed-forward network ("experts"), to reduce computational cost.
- **Mixture-of LoRA-Experts (MixLoRA)** [2] draws upon MoE for continual learning by using LoRA modules as lightweight "experts". Each FFN layer remains frozen, while a router activates the top-k LoRA adapters, learned during parameter-efficient fine-tuning, to capture new information.

Core Architecture:

T5 model pretrained using RIPOR instructions



Using a **energy-based Out of Distribution Detection mechanism**, a new LoRA is added when new document tokens are not represented well by the existing experts.

[1] Akari Asai, Zexuan Zhong, Danqi Chen, Pang Wei Koh, Luke Zettlemoyer, Hannaneh Hajishirzi, and tau Wen-Yih. 2024. Reliable, Adaptable, and Attributable Language Models with Retrieval. Retrieved from https://arxiv.org/abs/2403.03187
[2] Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. 2024. MixLoRA: Enhancing Large Language Models Fine-Tuning with LoRA-based Mixture of Experts. Retrieved from https://arxiv.org/abs/2404.15159
[3] Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. 2023. Scalable and Effective Generative Information Retrieval. Retrieved from https://arxiv.org/abs/2311.09134
[4] Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. 2024. Boosting continual learning of vision-language models via mixture-of-experts adapters. Retrieved from https://doi.org/10.1109/CVPR52733.2024.02191
[5] Tuan-Luc Huynh, Thuy-Trang Vu, Weiqing Wang, Trung Le, Dragan Gašević, Yuan-Fang Li, and Thanh-Toan Do. 2025. MixLoRA-DSI: Dynamically Expandable Mixture-of-LoRA Experts for Rehearsal-Free Generative Retrieval over Dynamic Corpora. Retrieved from https://arxiv.org/abs/2507.09924
[6] Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. 2025. From Matching to Generation: A Survey on Generative Information Retrieval. Retrieved from https://arxiv.org/abs/2404.14851
[7] Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer Memory as a Differentiable Search Index. Retrieved from https://arxiv.org/abs/2202.06991

Implementation

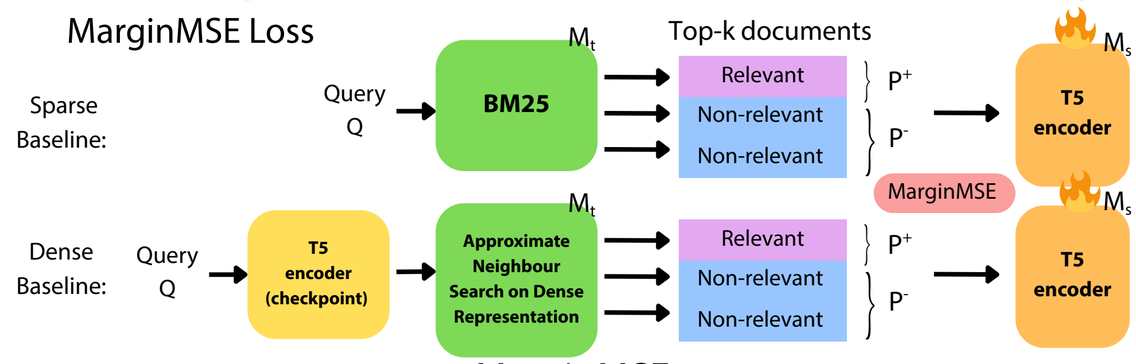
Our goal was to implement the MixLoRA-DSI paper from scratch to empirically validate its performance claims and understand practical implementation challenges.

Datasets used in paper:

- MSMARCO and NQ320k
- Data is split into D_0, D_1, D_2, D_3, D_4. Model is **trained sequentially on each one to mimic continual learning**.

Step 1: Pretraining T5 Encoder (based on RIPOR [3])

- Train T5 encoder to predict relevance score given query-document pair and negative document examples derived from baseline, using MarginMSE Loss

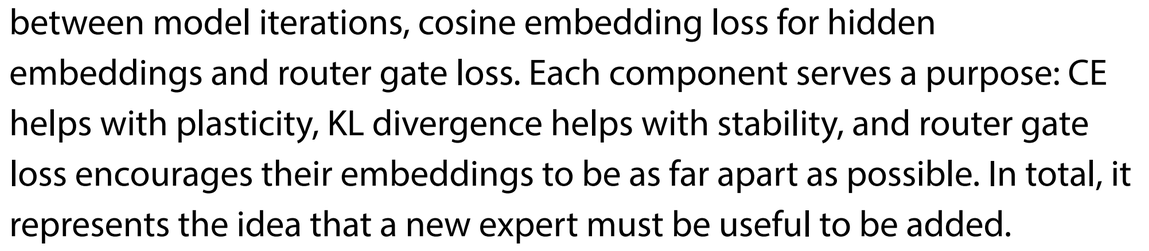


MarginMSE:

$$\mathcal{L}(Q, P^+, P^-) = \text{MSE}(M_s(Q, P^+) - M_s(Q, P^-), M_t(Q, P^+) - M_t(Q, P^-))$$

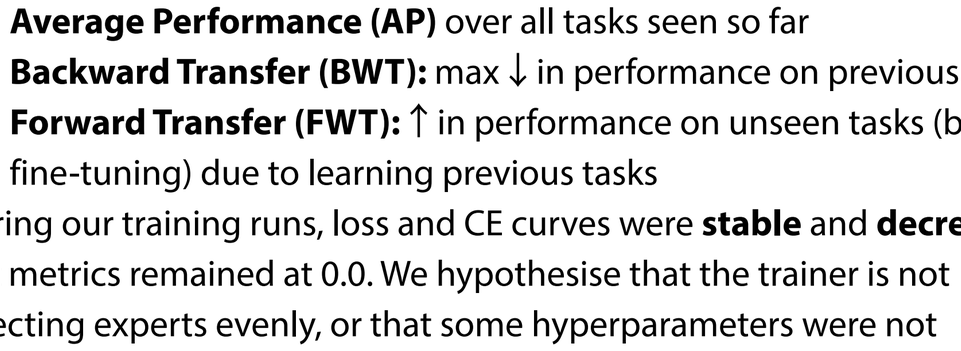
Step 2: Generate Residual Quantization (RQ)-based DocIDs

- RQ compresses DocIDs. It captures hierarchical structures between documents and is sufficiently succinct to be scalable for large datasets. Append RQ centroids to the original T5 vocabulary space. FAISS was chosen as the RQ model.



Step 3: Train T5 encoder to produce RQ-based DocIDs given a query

- *(Implementation detail)* Constrained Decoding: During the decoding process, apply a mask to allow the model to only produce tokens corresponding to codebook centroids. Without the mask, the model will produce either human language tokens or codebook centroids.



The loss function comprises 4 objectives: Cross entropy, KL divergence between model iterations, cosine embedding loss for hidden embeddings and router gate loss. Each component serves a purpose: CE helps with plasticity, KL divergence helps with stability, and router gate loss encourages their embeddings to be as far apart as possible. In total, it represents the idea that a new expert must be useful to be added.

Metrics and Results

Per-split evaluation: **Recall@10** and **Mean Reciprocal Recall@10**

Evaluation over splits:

- **Average Performance (AP)** over all tasks seen so far
- **Backward Transfer (BWT):** max ↓ in performance on previous tasks
- **Forward Transfer (FWT):** ↑ in performance on unseen tasks (before fine-tuning) due to learning previous tasks

During our training runs, loss and CE curves were **stable** and **decreasing**, but metrics remained at 0.0. We hypothesise that the trainer is not selecting experts evenly, or that some hyperparameters were not mentioned in the paper.