Project Assignment: Fine-Tune an Embeddings Model and Show Retrieval Accuracy Improvements

---

1\. Project Overview

In this project, you will fine-tune an **embeddings model** on **domain-specific data** from the **r/cisco** subreddit, which focuses on networking and telecommunications. The goal is to **improve retrieval accuracy** for domain-specific queries by adapting the model to the language and terminology used in networking discussions.

You will compare retrieval performance **before and after fine-tuning** using **key retrieval metrics** such as:

* **Normalized Reciprocal Rank (NRR)**
* **Normalized Discounted Cumulative Gain (NDCG)**
* **Mean Average Precision (MAP)**

Additionally, you will:
✅ **Create a custom dataset** of networking-related questions to evaluate retrieval improvements.
✅ **Build a RAG-based demo application** (using **Streamlit, Chainlit, or a similar tool**) to showcase the improvements in real-time retrieval.
✅ **Demonstrate before/after results** in your evaluation.

For reference, you can use **LlamaIndex's embedding fine-tuning guide**:
🔗 [LlamaIndex Fine-Tuning Embeddings](https://docs.llamaindex.ai/en/stable/examples/finetuning/embeddings/finetune_embedding/)

2\. Data Collection & Preprocessing

The training data is available in **an S3 bucket**:
📌 **s3://bigdatateaching/reddit-project/reddit/parquet/**

A. Extracting Domain-Specific Data

* **Filter for posts from the r/cisco subreddit.**
* Extract **post titles, body text, and relevant comments** as training data.
* Perform **text cleaning** (removing noise, links, formatting issues).
* Generate **training pairs** (e.g., question-answer or context-relevant response pairs).

B. Creating an Evaluation Dataset

* **Manually curate a set of networking-related queries** that resemble real-world search behavior.
* These should include:
  * General networking questions (e.g., *“What is the best firewall for small businesses?”*).
  * Cisco-specific queries (e.g., *“How to configure VLANs on a Cisco switch?”*).
  * Multi-step troubleshooting questions (e.g., *“Why is my BGP session not establishing?”*).

This dataset will be used to **evaluate retrieval accuracy before and after fine-tuning**.

3\. Technical Components & Implementation

A. Baseline Retrieval (Before Fine-Tuning)

* Use **a pre-trained embeddings model** (e.g., **Nomic, BAAI-bge, or similar**) to compute vector embeddings.
* Store embeddings in a **vector database** (FAISS, Weaviate, Pinecone, Chroma).
* Run **initial retrieval tests** using BM25 \+ embeddings-based search.
* Compute baseline retrieval metrics (**NRR, NDCG, MAP**).

B. Fine-Tuning the Embeddings Model

* Train the model using **domain-specific Reddit data** from **r/cisco**.
* Use contrastive learning or fine-tuning techniques from **LlamaIndex's fine-tuning guide**.
* Save the **fine-tuned embeddings model** for evaluation.

C. Evaluation & Comparison

* **Recompute vector embeddings** using the fine-tuned model.
* Perform **retrieval on the evaluation dataset**.
* Compare **before vs. after** performance using:
  * **NRR (Normalized Reciprocal Rank)** – How well does the top-ranked result match?
  * **NDCG (Normalized Discounted Cumulative Gain)** – How well are relevant results ranked?
  * **MAP (Mean Average Precision)** – Does retrieval improve across multiple queries?

D. Building a RAG-Based Demo Application

* **Develop a simple UI (Streamlit, Chainlit, etc.)** to interact with the system.

* Allow users to:
  * Enter a **query** and see **retrieved results** before/after fine-tuning.
  * Compare **search rankings, scores, and retrieved document snippets**.
  * View retrieval statistics (**NRR, NDCG, MAP scores**).

E. Deployment

* Deploy the solution **locally or on a cloud service** (AWS, Hugging Face Spaces, or a Flask-based API).

4\. Evaluation & Success Metrics

A. Embedding Model Performance

* Fine-tuning **successfully improves retrieval performance** compared to the baseline.
* Results show improvements in **NRR, NDCG, and MAP scores**.

**B. Evaluation Dataset & Analysis **

* A **well-structured dataset of domain-specific queries** is created.
* **Retrieval accuracy comparisons** before and after fine-tuning are presented.

**C. RAG Demo Application **

* A **working RAG-based app** (Streamlit, Chainlit, etc.) is developed.
* Users can **query the system, see retrieved documents, and compare performance**.

**D. Extra Credit **

* Additional **analysis on retrieval patterns**, fine-tuning challenges, or alternative embedding models.

E. Success Metrics

* **Higher NRR/NDCG/MAP scores** after fine-tuning.
* **More relevant documents retrieved** based on human evaluation.
* **User study or feedback** (if possible) to validate improvements.

5\. Why This Project Matters

This project teaches **critical skills in retrieval and fine-tuning** that are essential for **building domain-specific AI applications**:

✅ **Practical Fine-Tuning Experience:** Learn how to **adapt embeddings models to specific industries**.
✅ **Retrieval Accuracy Optimization:** Apply and measure retrieval performance using **real-world metrics**.
✅ **Hands-on RAG Implementation:** Build an **end-to-end RAG system** with a real dataset.
✅ **Real-World Impact:** Fine-tuning embeddings can **dramatically improve enterprise search systems**.

By completing this project, you will develop **deep expertise in retrieval models, embeddings fine-tuning, and evaluation methodologies**—skills that are highly valuable in **AI search, NLP, and enterprise AI applications**.

Tools & Resources

💡 **Vector Databases:** FAISS, Pinecone, Weaviate, ChromaDB
💡 **Embeddings Models:** Nomic, BAAI-bge, LlamaIndex
💡 **Fine-Tuning Frameworks:** LlamaIndex, Hugging Face, SentenceTransformers
💡 **Evaluation Metrics:** NRR, NDCG, MAP