SOICT

LEGAL DOCUMENT RETRIEVAL

Competition System Design

Team Members

Lê Thanh Tùng	23521740
Chu Dương Huy Phước	23521229
Mai Dương	22520302
Nguyễn Hoàng Duy	22520329

Table of Contents

Overview

Cuộc thi Legal Document Retrieval tập trung vào giải quyết bài toán "truy vấn trên dữ liệu luật pháp Tiếng Việt.

Legal Document Retrieval

Overview

Dữ liệu được cung cấp bởi ban tổ chức gồm 3 tập như sau:

- **Training data: là tập dữ liệu thật có gán nhãn, dùng để huấn luyện mô hình. Tập này gồm 119456 cặp truy vấn kèm văn bản liên quan.**
- **Public test: Là tập dữ liệu công khai phục vụ đánh giá mô hình. Tập này gồm 10000 truy vấn.**
- **Private test: Là tập dữ liệu công khai phục vụ đánh giá mô hình. Tập này gồm 50000 truy vấn.**

Overview

- **Input:**

Một truy vấn bằng ngôn ngữ tự nhiên, biểu thị một câu hỏi hoặc tình huống pháp lý.

Một kho dữ liệu pháp lý, là cơ sở dữ liệu được cấu trúc của các văn bản pháp luật.

- **Output:**

Một danh sách xếp hạng các tài liệu pháp lý liên quan đến truy vấn.

Hệ thống cần tập trung trả về 10 tài liệu phù hợp nhất.

Overview

- **Constraints:**

Hệ thống phải xử lý hiệu quả với các kho dữ liệu pháp lý quy mô lớn.
Độ liên quan cần được đánh giá dựa trên ngữ cảnh pháp lý và khả năng hiểu ngữ nghĩa của truy vấn.

- **Evaluation:**

Hiệu suất được đo lường bằng MRR@10 (Mean Reciprocal Rank @10), đánh giá chất lượng của 10 kết quả xếp hạng đầu tiên bằng cách ưu tiên thứ hạng của tài liệu phù hợp đầu tiên.

$$\mathrm{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\mathrm{rank}_i}$$

Preprocessing Data

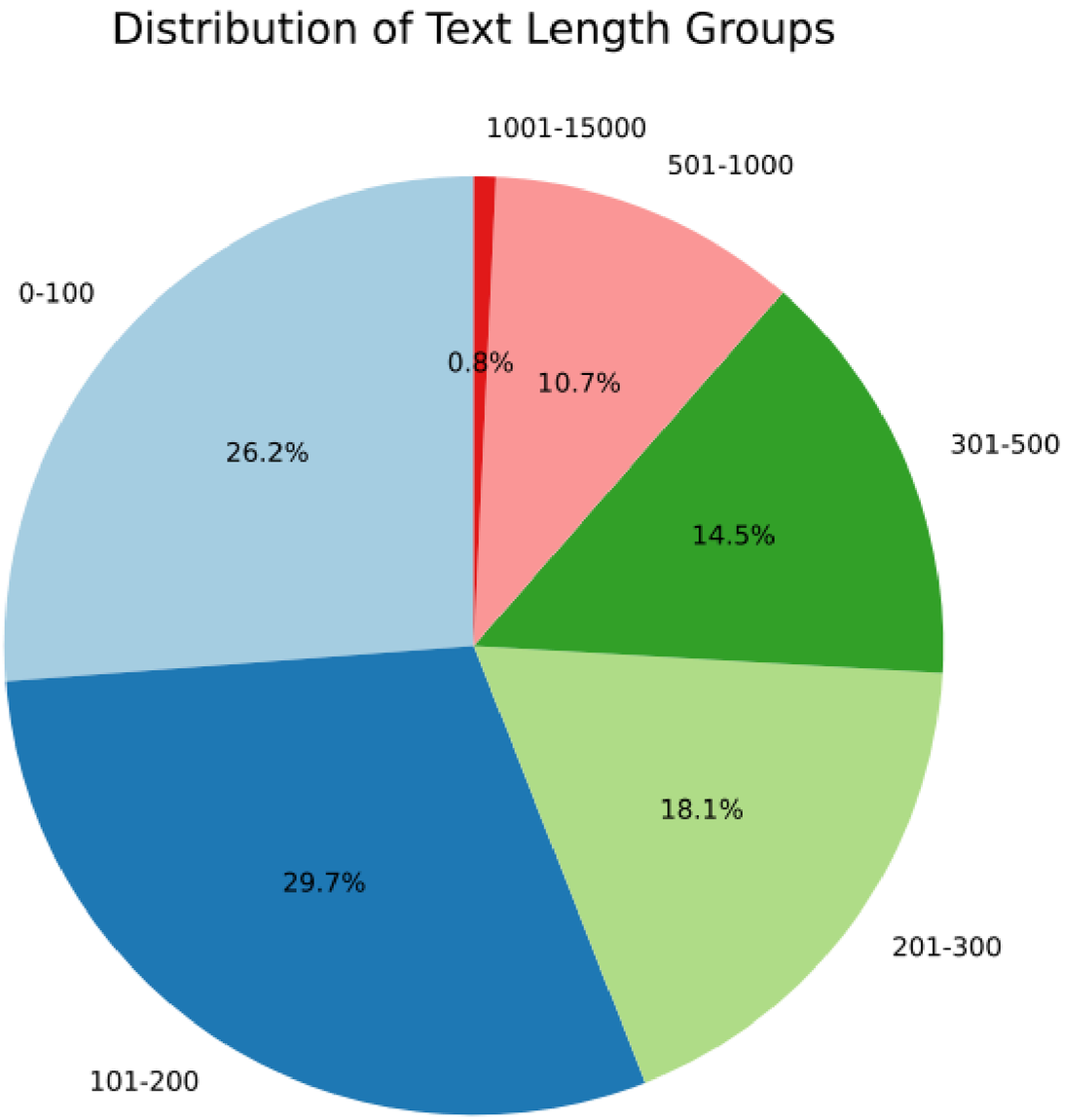
Chia tập data của BTC thành 3 tập train, val, test theo tỷ lệ 70:20:10.

Sau đó chia corpus thành file template:


```
    {
        "Field_id": "law",
        "infor": [
            {
                "infor_id": "0",
                "text": "Thông tư này hướng dẫn tuần tra, canh gác bảo vệ đê Điều trong mùa lũ đối với các tu
            },
            {
                "infor_id": "1",
                "text": "1. Hàng năm trước mùa mưa, lũ, Ủy ban nhân dân cấp xã nơi có đê phải tổ chức lực lượ
            },
            {
                "infor_id": "2",
                "text": "Tiêu chuẩn của các thành viên thuộc lực lượng tuần tra, canh gác đê\n1. Là người kho
            },
            {
                "infor_id": "3",
                "text": "Nhiệm vụ của lực lượng tuần tra, canh gác đê\n1. Chấp hành sự phân công của Ban chỉ
```

Chunk data

Trong corpus thì tỉ lệ một cid lớn hơn max token của một model bert bình thường (512 token) là khá đáng kể nên cần phải chunk data



Chunk data

Chunk theo Sentence Split:

- Max_word: 400
- Cố gắng giữ theo câu


```
_id": "law",
": [

 "stt": 0,
 "infor_id": "0",
 "chunk_id": "0_0",
 "text": "Thông tư này hướng dẫn tuần tra, canh gác bảo vệ đê Điều trong mùa lũ đối


 "stt": 1,
 "infor_id": "1",
 "chunk_id": "1_0",
 "text": "1. Hàng năm trước mùa mưa, lũ, Ủy ban nhân dân cấp xã nơi có đê phải tổ c


 "stt": 2,
 "infor_id": "2",
 "chunk_id": "2_0",
 "text": "Tiêu chuẩn của các thành viên thuộc lực lượng tuần tra, canh gác đê\n1. L


 "stt": 3,
 "infor_id": "3",
 "chunk_id": "3_0",
 "text": "Nhiệm vụ của lực lượng tuần tra, canh gác đê\n1. Chấp hành sự phân công c


 "stt": 4,
 "infor_id": "4",
 "chunk_id": "4_0",
 "text": "Phù hiệu của lực lượng tuần tra, canh gác đê\nPhù hiệu của lực lượng tuần
```

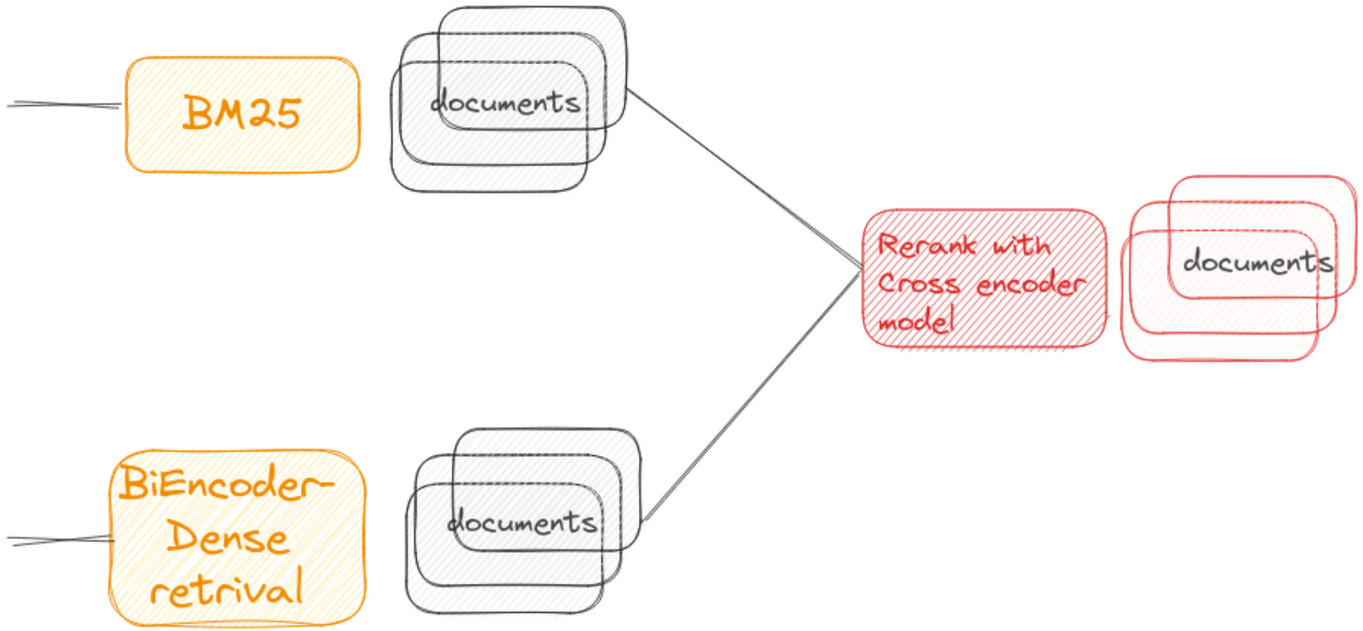
Lựa chọn giữa Sentence Chunking và Agentic Chunking?

Hybrid Search

Kết hợp hai kiểu search:

+ Sparse Search (BM25)

+ Dense Search (bi-encoder + cross-encoder)



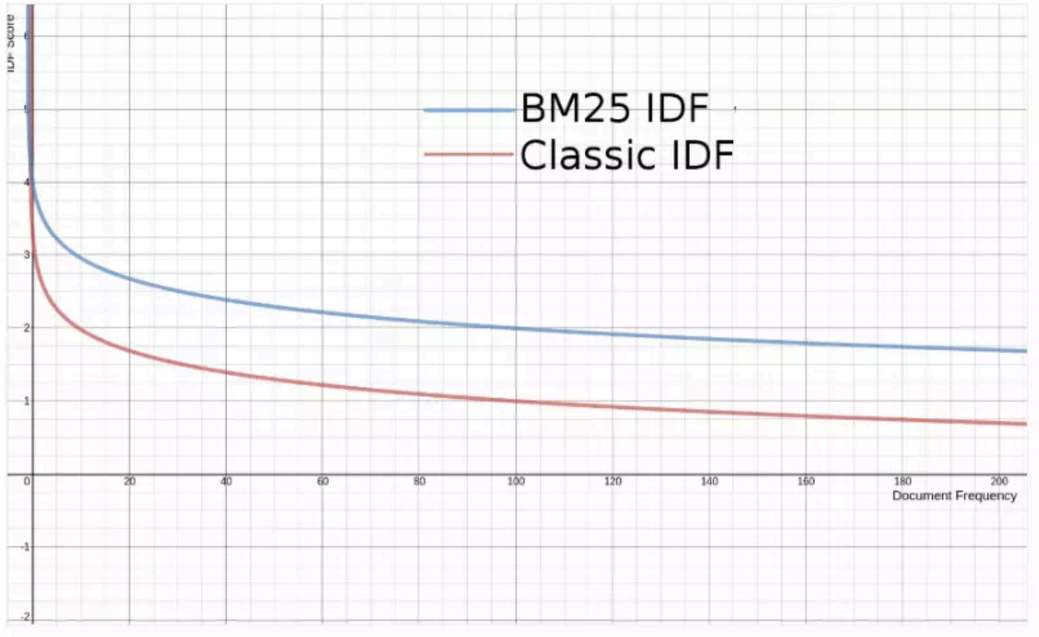
BM25 - Ranking Algorithm

BM25 (Best Match 25) là một thuật toán dùng để xếp hạng văn bản dựa trên truy vấn, thuộc nhóm các phương pháp truy hồi thông tin (Information Retrieval). Nó được phát triển từ mô hình Okapi BM25 vào những năm 1990, dựa trên mô hình không gian vector và lý thuyết xác suất.

$$\text{score}(D, Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)},$$

Why using BM25?

BM25 là một trong những phương pháp cơ bản và mạnh mẽ nhất để xếp hạng tài liệu, được sử dụng rộng rãi trong các hệ thống như: Công cụ tìm kiếm (Google, Bing), Truy hồi văn bản (tìm văn bản trong cơ sở dữ liệu lớn), Các bài toán NLP như hỏi đáp, phân loại văn bản.



Nhược điểm:

- **Văn bản ngắn: BM25 không hiệu quả với dữ liệu như câu thoại, tiêu đề.**
- **Ngữ cảnh phức tạp: BM25 không hiểu được ý nghĩa ngữ cảnh.**

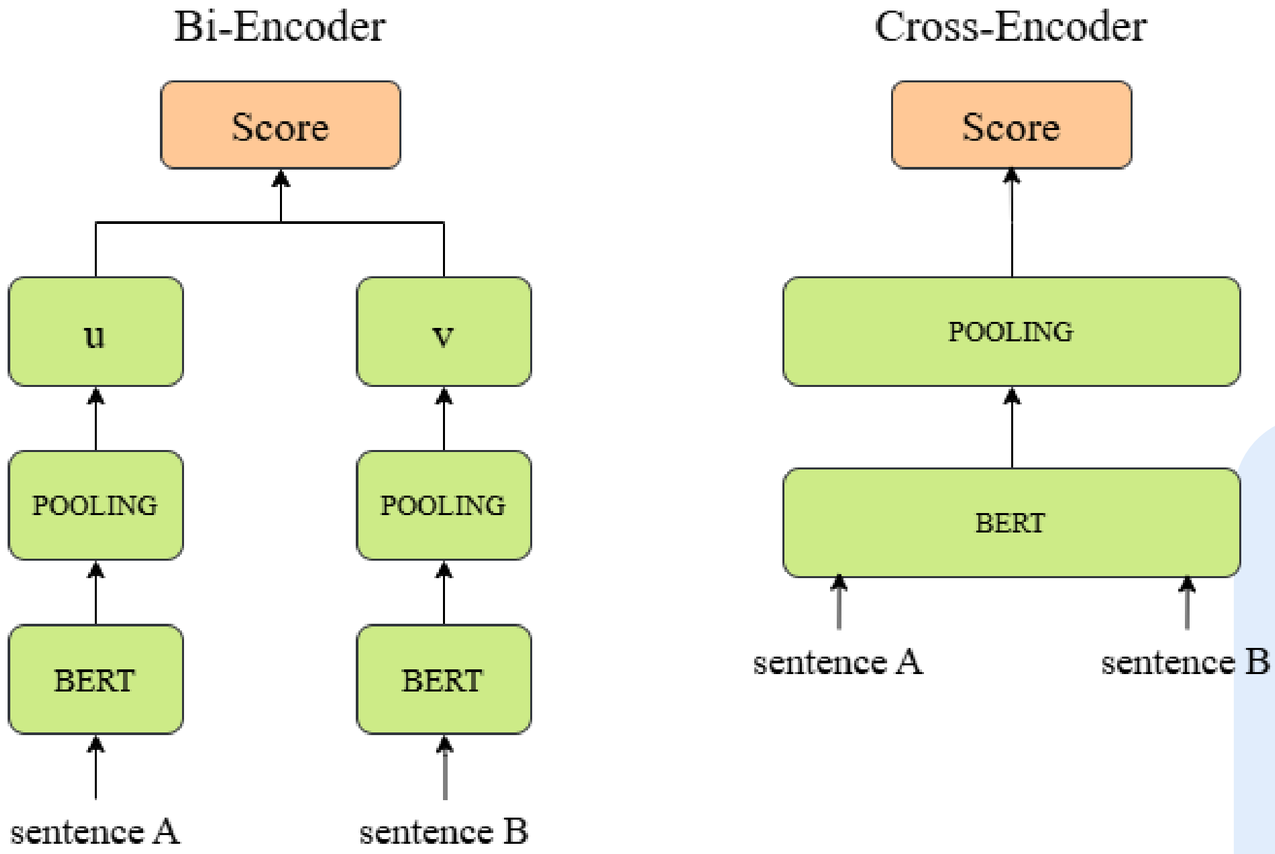
Dense Search

Bi-encoder model:

- bge-m3
- multilingual-e5-large

Cross-encoder model:

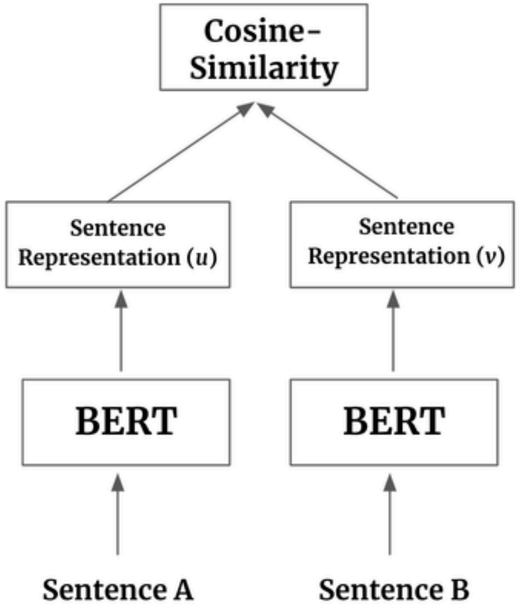
- bge-reranker-v2-m3



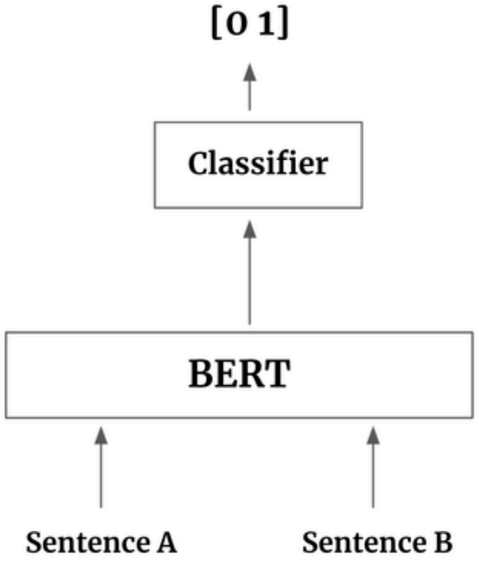
Bi-Encoder

- **Kiến trúc:** Trong mô hình bi-encoder, có hai encoder riêng biệt — một bộ dùng để mã hóa truy vấn đầu vào và một bộ dùng để mã hóa các tài liệu candidate. Các bộ encoder này hoạt động độc lập, tạo ra các embeddings cho truy vấn và từng tài liệu.
- Dùng Cosine Similarity để so sánh các embeddings

-> Sử dụng Bi-encoder khi bộ data lớn, sự tương quan giữa querry và document không được xem trọng



Bi-Encoder



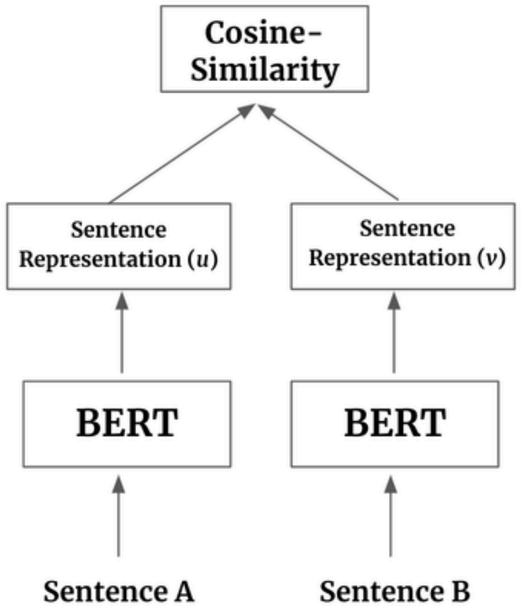
Cross-Encoder

Cross-Encoder

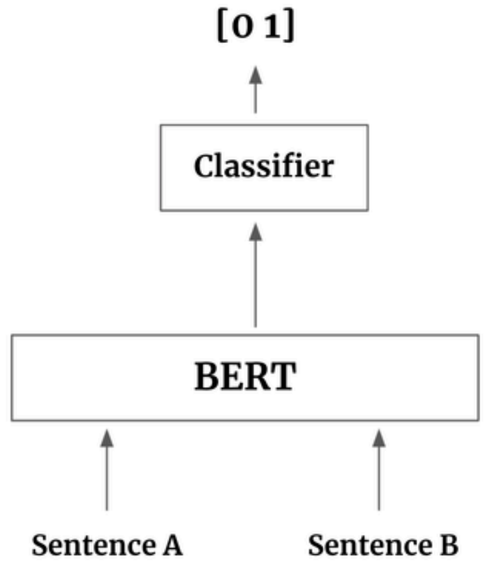
- **Kiến trúc:** Trong mô hình cross-encoder, query và document được xử lý cùng nhau trong một bộ encoder duy nhất. Điều này có nghĩa là mô hình nhận cả query và document làm đầu vào và tạo ra một biểu diễn chung.

  - Không dùng Cosine Similarity, tạo ra score bằng cách truy vấn cả query và documents

-> Sử dụng Cross-encoder cho các nhiệm vụ yêu cầu hiểu ngữ cảnh và mối quan hệ phức tạp giữa query và document

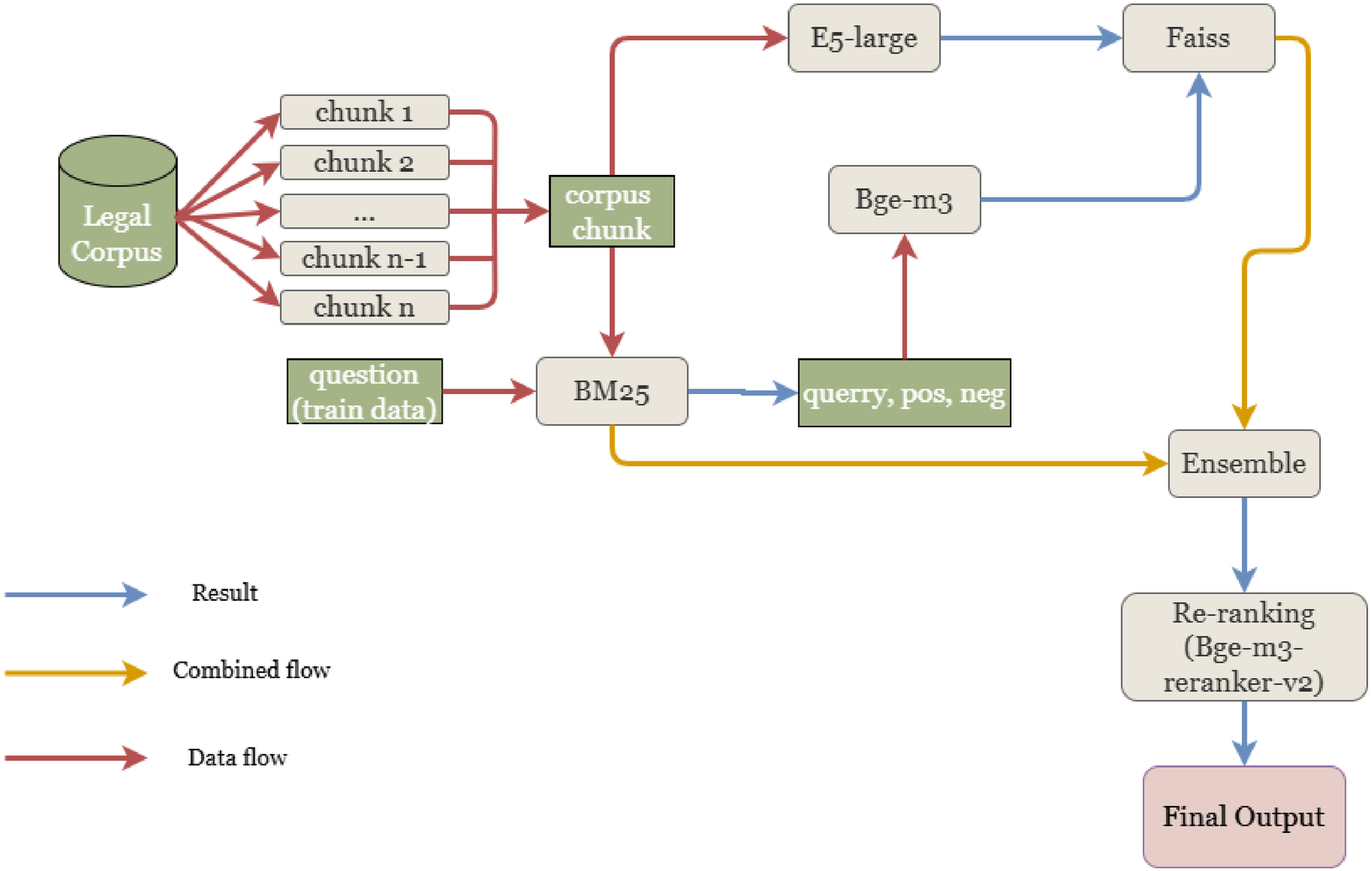


Bi-Encoder



Cross-Encoder

System Design



TẠI SAO DÙNG FAISS?

Ensemble bg3-me, e5 and BM25

- Sử dụng hai mô hình Bi-Encoder (bge-m3, multilingual-e5-large) để truy vấn tập dữ liệu. Truy xuất top k kết quả cho mỗi truy vấn từ từng mô hình.
- Loại bỏ Trùng Lặp theo ID: Kết hợp tất cả các kết quả từ hai mô hình. Lọc bỏ các kết quả trùng lặp dựa trên ID để đảm bảo tính duy nhất.
- **Kết hợp với top 20 kết quả search từ BM25**

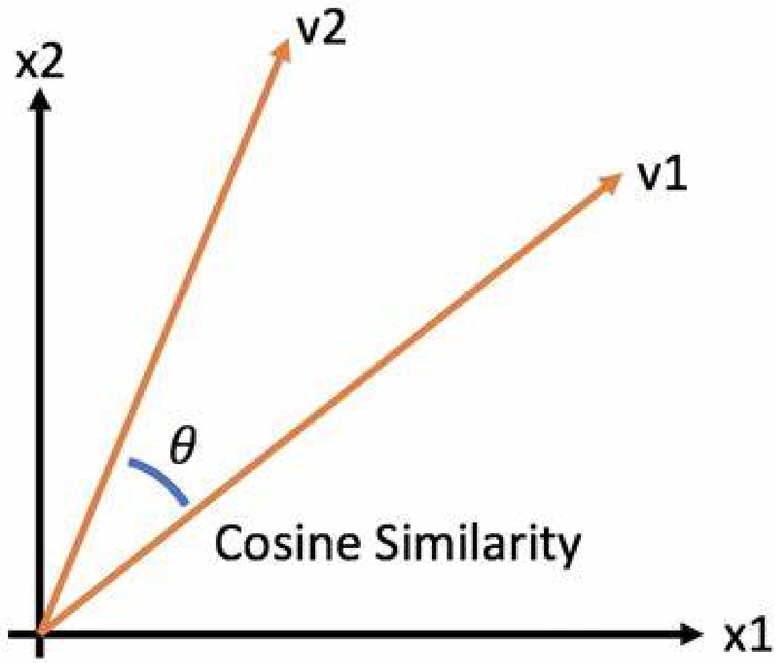
$$\text{ensemble} = \left(w_{\text{bge-m3}} \cdot \text{S}_{\text{bge-m3}} + w_{\text{e5}} \cdot \text{S}_{\text{e5}}\right) \cdot \text{S}_{\text{BM25}}.$$

Vấn đề với thang đo giữa BM25 và Bi-encoder

Bi-encoder trả về kết quả cosine similarity -> giá trị vector thuộc [0,1]

BM25 không bị giới hạn về giá trị tối đa mà thay đổi tùy thuộc vào kích thước tập dữ liệu và cách các từ trong truy vấn liên quan đến tài liệu

=> Mất cân bằng giữa các trọng số



Fine-tune bge-m3

Dùng hard negative từ BM25 để finetune sửa sai cho chính BM25

=>Phải tối ưu kết quả từ BM25

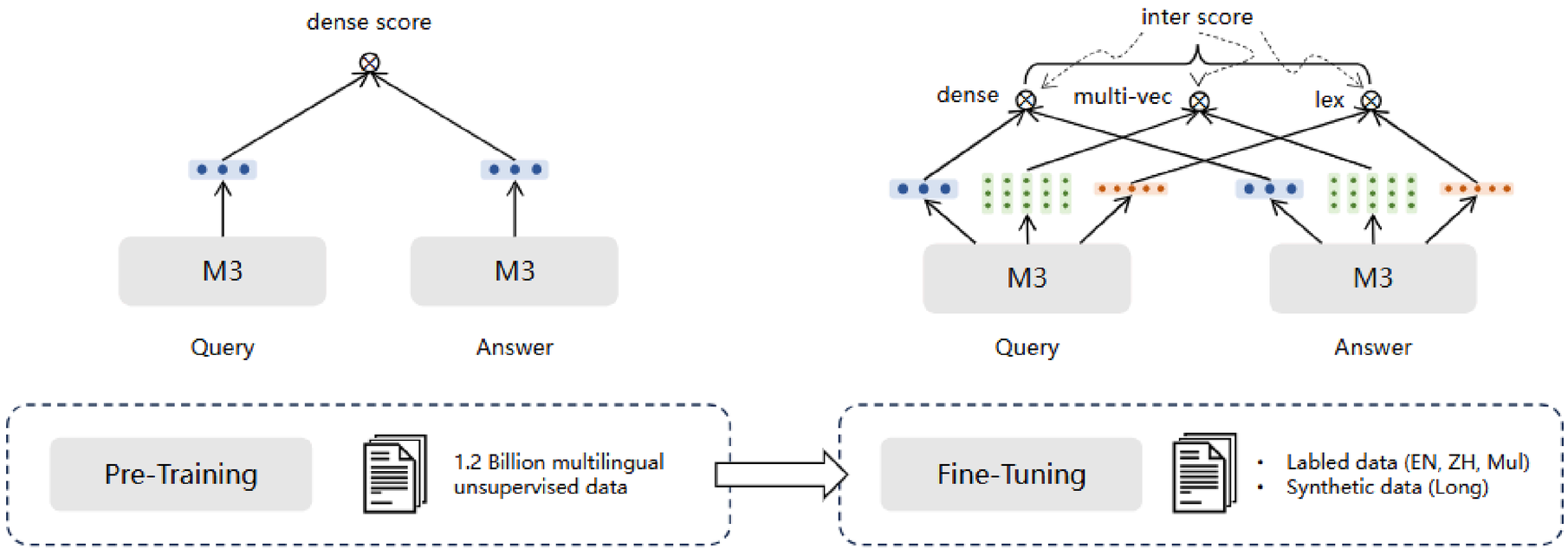
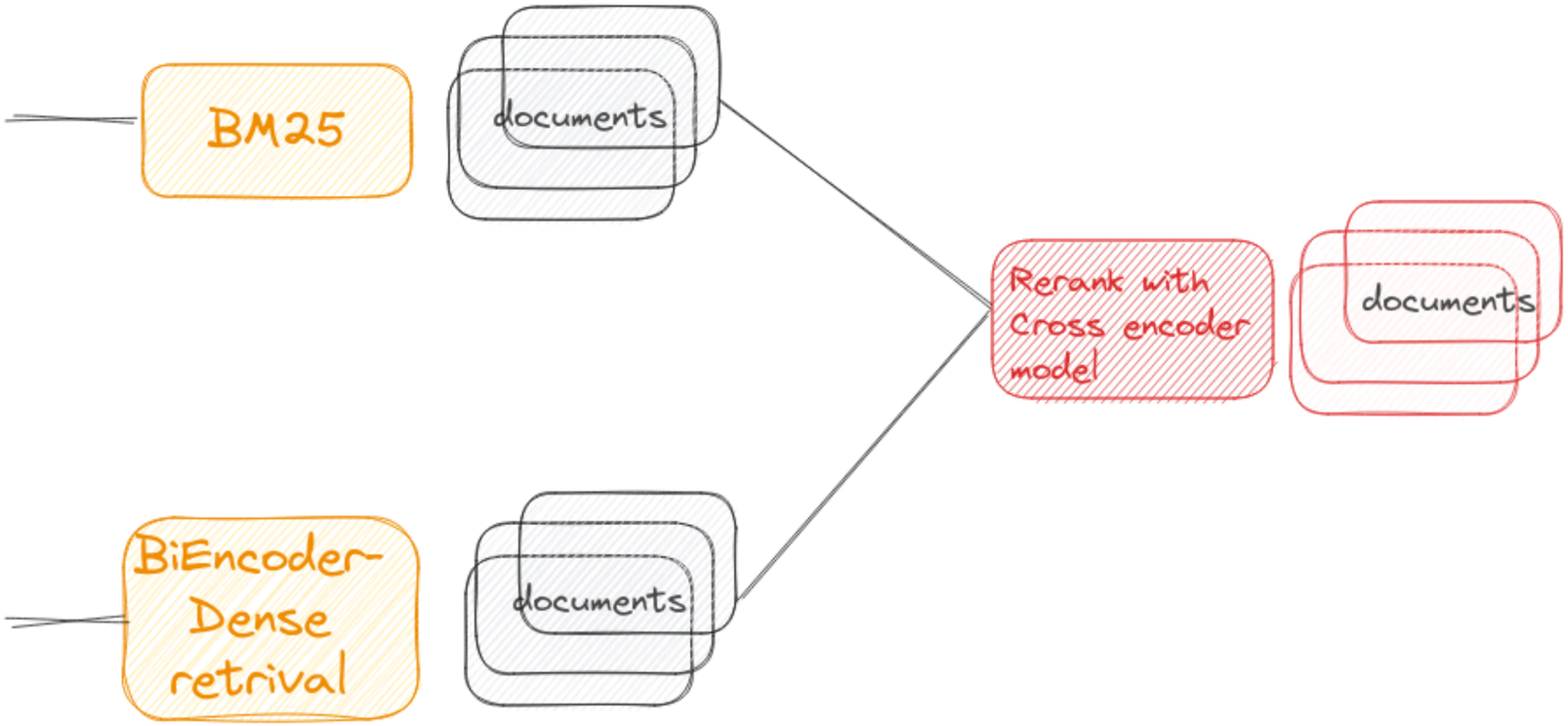


Figure 2: **Multi-stage training process of M3-Embedding with self-knowledge distillation.**

SỬ DỤNG TOP 20 KẾT QUẢ CỦA BM25



SỬ DỤNG TOP 20 KẾT QUẢ CỦA BM25

Table 1: Precision values on validation set for different top-k BM25 retrievals.

Top-k Documents	Precision (%)
Top-5	51.82
Top-10	61.06
Top-20	69.23
Top-30	70.5

Fine-tune bge-m3

Cơ chế in batch negatives:

Ví dụ một batch gồm

$$[[q, p, n_1, n_2], [q', p', n'_1, n'_2]],$$

trong đó các negative cho passage tích cực p bao gồm n1, n2, p′, n′ 1, n′ 2, và tương tự, các negative cho passage tích cực p′ bao gồm n′ 1, n′ 2, p, n1, n2.

```
!torchrun --nproc_per_node 2 \
    -m FlagEmbedding.finetune.embedder.encoder_only.m3 \
    --model_name_or_path BAAI/bge-m3  \
    --cache_dir ./cache/model \
    --train_data /kaggle/working/data_round1.jsonl \
    --cache_path ./cache/data \
    --train_group_size 2 \
    --query_max_len 128 \
    --passage_max_len 400 \
    --pad_to_multiple_of 2 \
    --same_dataset_within_batch True \
    --small_threshold 0 \
    --drop_threshold 0 \
    --output_dir /kaggle/working \
    --overwrite_output_dir \
    --learning_rate 2e-5 \
    --fp16 \
    --num_train_epochs 18 \
    --per_device_train_batch_size 4 \
    --dataloader_drop_last True \
    --warmup_ratio 0.1 \
    --gradient_checkpointing \
    --deepspeed /kaggle/working/FlagEmbedding/examples/finetune/ds_stage0.json \
    --logging_steps 1\
    --save_steps 5200    \
    --weight_decay 0.01 \
    --negatives_cross_device \
    --temperature 0.02 \
    --normalize_embeddings True \
    --unified_finetuning True \
    --use_self_distill True \
    --fix_encoder False \
    --self_distill_start_step 0 \
    --gradient_accumulation_steps 8
```

Fine-tune bge-m3

Cơ chế in batch negatives:

1 positive 1 negative
1 positive 1 negative
1 positive 1 negative
1 positive 1 negative

- Luôn có 1 positive và “random”để chọn các negative còn lại

```
!torchrun --nproc_per_node 2 \
    -m FlagEmbedding.finetune.embedder.encoder_only.m3 \
    --model_name_or_path BAAI/bge-m3  \
    --cache_dir ./cache/model \
    --train_data /kaggle/working/data_round1.jsonl \
    --cache_path ./cache/data \
    --train_group_size 2 \
    --query_max_len 128 \
    --passage_max_len 400 \
    --pad_to_multiple_of 2 \
    --same_dataset_within_batch True \
    --small_threshold 0 \
    --drop_threshold 0 \
    --output_dir /kaggle/working \
    --overwrite_output_dir \
    --learning_rate 2e-5 \
    --fp16 \
    --num_train_epochs 18 \
    --per_device_train_batch_size 4 \
    --dataloader_drop_last True \
    --warmup_ratio 0.1 \
    --gradient_checkpointing \
    --deepspeed /kaggle/working/FlagEmbedding/examples/finetune/ds_stage0.json \
    --logging_steps 1\
    --save_steps 5200    \
    --weight_decay 0.01 \
    --negatives_cross_device \
    --temperature 0.02 \
    --normalize_embeddings True \
    --unified_finetuning True \
    --use_self_distill True \
    --fix_encoder False \
    --self_distill_start_step 0 \
    --gradient_accumulation_steps 8
```

Results

Pretrained Models (MMR10 Scores)		
Model	**Val (MMR10)**	**Test (MMR10)**
BM25	0.6139	0.6052
Bge	0.6660	0.6681
e5	0.6612	0.6576
Bge+e5	0.6804	0.6811
Bge+bm25	0.6534	0.6497
Bge+e5+bm25	0.6493	0.6597
After Fine-Tuned Bge-m3 (MMR10 Scores)		
Model	**Val (MMR10)**	**Test (MMR10)**
Bge	0.5317	0.5555
Bge+e5+bm25	0.7417	0.7193
Bge+bm25	0.7421	0.7348

Table 2: The result First-Retrieval (MMR10 scores)

Results

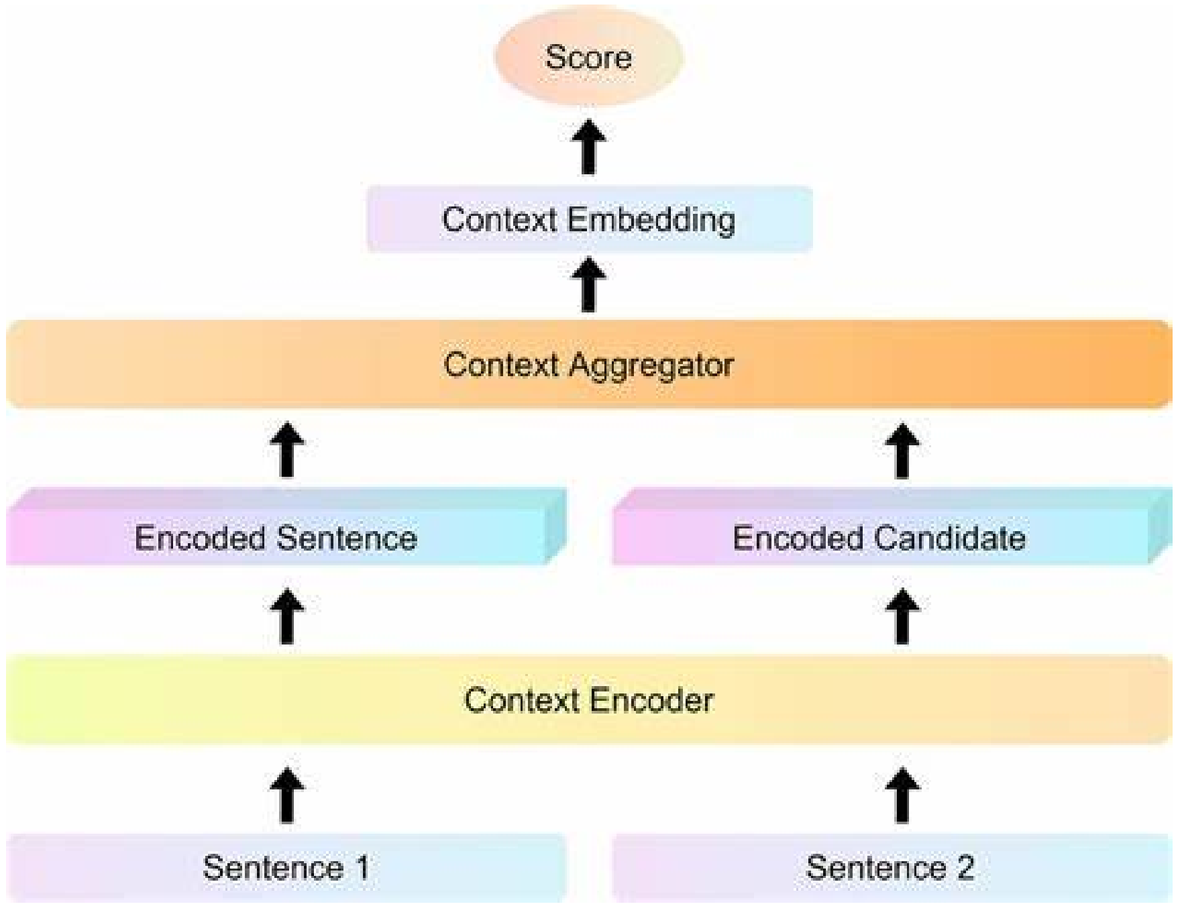
Analysis suboptimal of finetune

- Train_group_size khá nhỏ(=2) nên nó không học được hết 20 hard negative từ bm25.
- Negative của in batch lớn hơn negative từ bm25. Mà in batch theo 1 paper khác nói rằng không khác gì random
- Paper “Learning To Retrieve: How to Train a Dense Retrieval Model Effectively and Efficiently” chỉ ra rằng bm25 có thể không tốt khi finetune bi encoder.

Results

Rerank

Dùng Cross-encoders - bge-reranker-v2-m3 để rerank top 100 kết quả



Results

Reranking (MMR10 Scores)		
First-Retrieval by	**Val (MMR10)**	**Test (MMR10)**
Bge-pretrained + e5	0.7150	0.7155
Bge-tuned+bm25	0.7708	0.7746

Table 3: The result after reranking (MMR10 scores)

THANKS FOR WATCHING