Information Retrieval Assignment 5

Jimmy Gore

December 2024

Introduction

I took a rather basic approach to this assignment. We explored many different retrieval models such as binary, tf-idf, BM25, bi-encoders, and cross-encoders.

I decided to use a bi-encoder (sentence transformer) because they had the highest precisions across all models we have explored.

Search Strategy

I first started by embedding all the answers after removing HTML tags then I saved these embeddings into .npy files. This was done so that if I wanted to change my querying strategy, I could get the results faster.

For each topic I concatenated the title and body (HTML removed), computed its embedding and performed a cosine similarity search on the answer embeddings.

Computing the answer embeddings takes quite a long time but performing the search is relatively fast.

Bi-Encoder Selection

The sentences transformers library provides many bi-encoders but I would need to determine which model I would choose for my final submission. I did this by evaluating P@10 on topics_1.json for every bi-encoder model provided by the package. I expected the results for this experiment to mirror the leaderboard in the Sentence Transformers documentation.

Bi-Encoder Model Name	P@10
all-mpnet-base-v2	0.4357
multi-qa-mpnet-base-cos-v1	0.4018
sentence-t5-xxl	0.3951
multi-qa-mpnet-base-dot-v1	0.3920
gtr-t5-xxl	0.3905
all-roberta-large-v1	0.3876
all-mpnet-base-v1	0.3783
sentence-t5-xl	0.3739
gtr-t5-xl	0.3669
all-MiniLM-L12-v2	0.3595
sentence-t5-large	0.3563
gtr-t5-large	0.3534
all-MiniLM-L6-v2	0.3532
multi-qa-distilbert-cos-v1	0.3524
all-distilroberta-v1	0.3449
all-MiniLM-L12-v1	0.3411
multi-qa-MiniLM-L6-cos-v1	0.3221
sentence-t5-base	0.3108
gtr-t5-base	0.3063
paraphrase-mpnet-base-v2	0.3050
multi-qa-MiniLM-L6-dot-v1	0.3030
multi-qa-distilbert-dot-v1	0.2999
all-MiniLM-L6-v1	0.2926
paraphrase-distilroberta-base-v2	0.2637
paraphrase-MiniLM-L12-v2	0.2507
msmarco-bert-base-dot-v5	0.2494
msmarco-distilbert-dot-v5	0.2402
paraphrase-multilingual-mpnet-base-v2	0.2363
paraphrase-TinyBERT-L6-v2	0.2296
distiluse-base-multilingual-cased-v1	0.2241
msmarco-distilbert-base-tas-b	0.2208
distiluse-base-multilingual-cased-v2	0.2196
paraphrase-multilingual-MiniLM-L12-v2	0.2106
paraphrase-MiniLM-L6-v2	0.2046
paraphrase-albert-small-v2	0.1971
paraphrase-MiniLM-L3-v2	0.1839
average_word_embeddings_glove.6B.300d	0.1007
average_word_embeddings_komninos	0.0947

Table 1: Bi-Encoder P@10 Comparison - topics_1.json

From this table we can see that all-mpnet-base-v2 scored the highest, this result was expected, as this model also scores highest out of the original models for semantic search tasks. I then used this model for my submission

Cross-Encoder Re-Ranking Attempt

When using a bi-encoder it is common to re-rank the retrieved results with a cross-encoder. This is because cross-encoders are much more computationally intensive but more accurate.

I used the ms-marco-MiniLM-L-12-v2 cross-encoder because it had the highest MRR@10 on the MS Marco dev set. I then re-ranked the results from the all-mpnet-base-v2 bi-encoder but the re-ranking actually decreased the P@10 scores.