UNIVERSITY OF SOUTHERN MAINE

Introduction to Information Retrieval, Fall 2024, Assignment 3

Instructor: Behrooz Mansouri

Due: October 31, 2024

In this assignment, you will explore more advanced information retrieval models; Bi-encoder and Cross-encoder in Python. The goal is to find relevant answers to given questions about traveling. The data files remain similar to the first and second assignments.

Designing Retrieval Models

For each model, you will implement two variants: a basic model with no fine-tuning, and a fine-tuned model. The bi-encoder model should search over the full collection (all answers). To use cross-encoders (due to their efficiency), you will use them to re-rank the top-100 results returned by the bi-encoder models. Each cross-encoder variant should re-rank its corresponding bi-encoder results (e.g., with the fine-tuned cross-encoder, you should re-rank the bi-encoder re-ranked).

For your implementation, you can use any models from the sentence_transformers python library. However, part of your grade goes to developing an effective system for topics in the second topic file. Therefore, carefully read the model you have chosen and how it was trained.

Fine-tuning. To fine-tune the models, you need to split the data (topics from the first file) into train, validation, and test sets. You can try either 80-10-10, or 90-5-5- splits. Your code should separate the fine-tuning and testing stages. Your results and analysis should be reported only on your test set.

Evaluation Measures

As you can each have your own test sets, you need to create qrel file for that set. Then evaluate your models reporting the P@1, P@5, nDCG@5, MRR, MAP. For metrics other than nDCG, the relevant documents are those with scores 1 and 2 (in the qrel file). You can use any tool that you want for evaluation. However, this tool should be standard, and the results should be reproducible (Trec_eval and Ranx are recommended). For each topic, you should exactly return 100 results.

For your report in the PDF file, you need to include a table as follows for the test topics:

Model	P@1	P@5	nDCG@5	MRR	MAP
Bi-encoder					
Bi-encoder-finetuned					
Cross-encoder					
Cross-encoder-finetuned					

Report and Analysis

For each of the four systems, you will generate two result files. One result file is for your test set, and the other is for topics in the topic_2 file. Make sure the result files can be passed to Trec_eval and Ranx for evaluation using the correct format. You do not need to provide an evaluation script. For your test test, complete the table above (averaged over all the topics). Then provide a ski-jump plot based on P@5. Using this plot, discuss and reason about the success and failure of your system. You should provide at least one pair of successful and failed (topic, answer) in your discussion for each model. You should provide topics, and retrieved instances to support your claims.

Notes for submission:

1. Python file/files (only .py is acceptable, not .ipynb) with codes for all retrieval. Codes should be well-structured with comments to run. You should have a README file that provides clear guidance on how to run your code and get the result files. Explicit file paths should not be used in the code, and topic files should be passed as arguments to your code.
2. You will get 8 retrieval result files by the end of this assignment. You should submit them all following the naming convention below:
   - result_bi_1.tsv: bi-encoder with no fine-tuning on your test set
   - result_bi_2.tsv: bi-encoder with no fine-tuning on topic_2 file
   - result_bi_ft_1.tsv: bi-encoder with fine-tuning on your test set
   - result_bi_ft_2.tsv: bi-encoder with fine-tuning on topic_2 file
   - result_ce_1.tsv: cross-encoder with no fine-tuning on your test set
   - result_ce_2.tsv: cross-encoder with no fine-tuning on topic_2 file
   - result_ce_ft_1.tsv: cross-encoder with fine-tuning on your test set
   - result_ce_ft_2.tsv: cross-encoder with fine-tuning on topic_2 file
3. A .pdf file with your results table, ski-jump plot, and analysis. In this file, you will also explain what model you have chosen and justify your choice.
4. Any assumptions made by students should be explicitly mentioned in the submitted

Note: Any submission not in the format explained above will be dropped, resulting in 0, without the possibility of regrading