COL764/COL7364: Information Retrieval & Web Search
Assignment 3: Text ranking with transformer-based models

October 8, 2025

Deadline

Submission of the complete implementation with report on the algorithms is due on October 21, 2025, 11:59 PM.

Instructions

Follow all the instructions. Not following these instructions will result in penalty.

1. **The assignment is to be done individually or in pairs.** Do **not** collaborate by sharing code, algorithms, or any other details. Discussion for clarification is allowed, but it must happen on Piazza.
2. All programs must be written in **Python (3.12)**. We are restricting this assignment to Python since the APIs required are abundantly available in python.
3. All submissions will be evaluated on **Ubuntu Linux**. Ensure that file names, paths, and argument parsing are Linux-compatible. Each team will be given an account on the baadal machine. The code needs to be tested there as we will be running evaluation scripts on baadal only.
4. **No deadline extensions** will be given. Late submission is allowed with penalty as outlined in the Introductory class. Since the assignment requires substantial implementation effort, as well as some manual tuning for performance, so it is advised to start early. **Do not wait until the last moment.**

1 Assignment Description

In this assignment, you are expected to experiment and evaluate multiple methods. Apart from defining a framework and a few minimum things that are expected, the rest is up to you and marks will awarded based on the quality of the work done. Much of the development and experimentation will have to take place on your own machines. It will take a few days for us to get the baadal VMs ready. The tasks are straightforward.

Task 1: Implement the "retrieve-and-rerank" framework discussed in class using BERT.

Task 2: Evaluate the above against 2 different baselines.

Task 3: Implement at least 1 way of improving the accuracy numbers.

NOTE: It is *crucial* that you submit a well-written report that includes details of the method used, plots which show comparisons of the various metrics on the different baselines.

2 Tasks

Task 0: Setup

You simply need to set up the infrastructure required for this assignment. This includes the following:

- Install `pyserini` using `pip install pyserini`. Pyserini comes with an ready-made index for the MS-MARCO dataset. Note that pyserini requires a Java installation. See here for more details: `https://github.com/castorini/pyserini/`
- Install `ir_datasets` using `pip install ir_datasets`. We will use this for downloading the `trec-dl-hard` queries. See here for more details: `https://ir-datasets.com/msmarco-passage.html#msmarco-passage/trec-dl-hard`
- Download the `trec_eval` package to compute the metrics.

  For any issues on baadal , the following file might help with the task_0 Task 0 Setup Report (PDF).

Task 1: Implement the "Retrieve and Rerank" framework using BERT

As discussed in class, the retrieve and rerank framework retrieves the top-k candidates using a fast index (pyserini with BM25) and then reranks using a "heavy" model. Make sure you understand that the rerank component here requires you to input pairs of ⟨query,document⟩.

Analyse the results with varying values of k. Note that, we are purposefully not specifying which values of k you should experiment with. The only constraint is that $k >= 10$. The final set of results will contain the top-10 results.

Analyse the behaviour of this retrieve and rerank framework with the following metrics: NDCG@1, NDCG@5, NDCG@10, MRR@10. Explain this behaviour with tables and charts as appropriate ("Explain" means, actually write down your observations in the report).

Note the following for your implementation (and make sure you report *all* the relevant details in your report.

- Recall how BERT is supposed to be used as a re-ranker – it has to be fine-tuned end-to-end with a relevance classifier. You are *not required* to fine-tune the model. Instead, choose a suitable fine-tuned model from HuggingFace (`https://huggingface.co/`. You need to carefully document and report exactly which model in your report. When awarding marks, we will consider the choice of model.
- Also recall that BERT has a width limitation and we studied multiple ways of chunking documents and aggregating scores. *If this is needed*, carefully document how you decided to do chunking and how you decided to aggregate the scores. This will also carry additional marks.

Task 2: Comparisons with BERT alternatives

Present an evaluation *and analysis* of the same retrieve-and-rerank framework with *at least two* alternatives to BERT. Your evaluation should report at least the same metrics as for Task 1 and

may include other metrics that you think are important or offer insight. The analysis should contain *a minimum* of comparison tables and plots. Additional analysis and/or insights you may be able to offer will be considered for marks.

Some points for you to consider *and document in your report*:

- For a good understanding of how these models work, please read the original papers carefully.
- Look for experiments that others have already done on the same benchmark (there are leader boards for TREC tasks). Do any of the methods there fit into the framework of "retrieve-and-rerank"? If so, can you reproduce their results? Or reproduce their methods?

Task 3: Improving on the best from Task 1 and Task 2

In this task, utilize at least *one method* of improving your metrics on the best methods from the previous two tasks. There are multiple options to consider. For example: query rewriting (this could be zero-shot, or using pseudo-relevance feedback), multi-stage reranking.

Explain the method in detail in your report. Report on *at least* the metrics that you computed for Tasks 1 and 2. Additional analysis, creative ways of improving the performance will be considered for marks.

Please note that the method you choose should be among those that we have already studied in class, or have a clear basis in a paper that has been previously published. What is *not* acceptable: a one line, black-box API that rewrites the query.

3 Submission

- We are allowing the use of `ir_datasets, huggingface, pyserini`. If you need any other library, *clear it with the TAs first.*
- Your report will carry a lot of weight, so make sure you spend a lot of time and effort to write it properly.
- Apart from a zip of all your code, you are required to submit a shell script that will generate every single plot that is present in your report. **More details of exactly what you need to submit here will be explained later.**