

INFORMATION RETRIEVAL

HOMEWORK EXERCISES L05. NEURAL IR

SUZAN VERBERNE 2022

GENERAL INSTRUCTIONS

- You are going to do a practical exercise with ColBERT reranking in PyTerrier (A Python framework for performing information retrieval experiments, building on Terrier)
- Submit only the added code snippets and requested output as PDF, don't submit the whole notebook
- If you need help, you can contact the TAs at ircourse@liacs.leidenuniv.nl
- The goal of this exercise are
 1. to learn to recognize the steps of a retrieval and ranking pipeline;
 2. to understand the data structures necessary for the pipeline;
 3. to learn to interpret model output

EXERCISE PREPARATIONS

Preparation:

1. Start a new Python notebook on Google colab
2. Make sure that the notebook uses a GPU (under Resources -> Change runtime type)
3. Follow the steps in this tutorial (3.2 only):
<https://github.com/terrier-org/ecir2021tutorial/blob/main/notebooks/notebook3.2.ipynb>
and make sure everything runs and you get output

EXERCISE 1

- Let's first collect the results
- In the Experiment function, ColBERT is compared to DPH, the model that is used as default first-stage retriever in Terrier [1].
 1. Show the results table with DPH and DPH >> ColBERT as two rows
 2. Look up in the PyTerrier documentation [2] how you can replace DPH by BM25. Adapt this in your code and show the result table for BM25 and BM25 >> ColBERT
 3. Which do you prefer, DPH+ColBERT, or BM25+ColBERT?

[1] DPH hypergeometric model: Robertson and Walker, 1994. "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval"

[2] https://pyterrier.readthedocs.io/en/latest/pipeline_examples.html

EXERCISE 2

The test collection used in the tutorial is TREC-COVID

<https://ir.nist.gov/covidSubmit/index.html>

We are going to explore the topic collection a bit.

1. Add code to your notebook to print the first 5 topics from the test collection.
2. Look up in the PyTerrier documentation [3] how you can view the results per query. Show the query (id and content) with the lowest nDCG@10 and the queries with the highest nDCG@10.

EXERCISE 3

1. Run the retrieval pipeline for the query with the lowest nDCG@10 and one of the queries with the highest nDCG@10. What is the highest ranked document? Is it relevant or not?
2. Output the attention matrix for the document retrieved in first position for the query with the lowest nDCG@10 and for the document retrieved in first position for the query with the highest nDCG@10. What does it tell you?