INFORMATION RETRIEVAL

HOMEWORK EXERCISES L05. NEURAL IR

SUZAN VERBERNE 2022

Universiteit Leiden

GENERL 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 1 - SOLUTIONS

1. Results table with DPH and DPH >> ColBERT

	name	map	ndcg	ndcg_cut.10	P.10	mrt
0	DPH	0.068056	0.165653	0.609058	0.658	49.933852
1	DPH >> ColBERT	0.074727	0.172074	0.689785	0.750	689.051942

EXERCISE 1 - SOLUTIONS

```
br = pt.BatchRetrieve(index, wmodel="BM25") % 100
pipeline = br >> pt.text.get_text(dataset, 'abstract') >> colbert
pt.Experiment(
    [br, pipeline],
    topics,
    qrels,
    names=['BM25', 'BM25 >> ColBERT'],
    eval_metrics=["map", "ndcg", 'ndcg_cut.10', 'P.10', 'mrt']
)
```

2. Results table with BM25 and B25 >> ColBERT

	name	map	ndcg	ndcg_cut.10	P.10	mrt
0	BM25	0.077892	0.177767	0.644374	0.692	49.320280
1	BM25 >> ColBERT	0.081463	0.180422	0.666929	0.738	688.697242

EXERCISE 1 - SOLUTIONS

- It depends on the use case:
  - MAP and nDCG on the full ranking are higher for BM25. Thus, a user who inspects all 100 results will get more relevant information with BM25+ColBERT than with DPH+ColBERT
  - nDCG@10 and P@10 are higher for DPH, indicating a higher effectiveness by BM25 for the top-10 retrieved documents. Thus, a user who only inspects the top-10 will get more relevant information with DPH+ColBERT than with BM25+ColBERT

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.

[3] https://pyterrier.readthedocs.io/en/latest/experiments.html

EXERCISE 2 - SOLUTIONS

1. topics.iloc[0:5]

index	qid	query
0	1	what is the origin of covid 19
1	2	how does the coronavirus respond to changes in the weather
2	3	will sars cov2 infected people develop immunity is cross protection possible
3	4	what causes death from covid 19
4	5	what drugs have been active against sars cov or sars cov 2 in animal studies

EXERCISE 2 - SOLUTIONS

2. Result per query:

```
br = pt.BatchRetrieve(index, wmodel="BM25") % 100
pipeline = br >> pt.text.get_text(dataset, 'abstract') >> colbert
result = pt.Experiment(
    [pipeline],
    topics,
    qrels,
    names=['BM25 >> ColBERT'],
    eval_metrics=['ndcg_cut.10'],
    perquery=True
)
result.sort_values('value')
```

- qid 31 has the lowest nDCG@10: 0.151771
- qids 42, 20, 40, 22, 38 have the highest nDCG@10: 1.00

EXERCISE 2 - SOLUTIONS

- 31 : how does the coronavirus differ from seasonal flu
- 42 : does vitamin d impact covid 19 prevention and treatment
- 20 : are patients taking angiotensin converting enzyme inhibitors ace at increased risk for covid 19
- 40 : what are the observed mutations in the sars cov 2 genome and how often do the mutations occur
- 22 : are cardiac complications likely in patients with covid 19
- 38 : what is the mechanism of inflammatory response and pathogenesis of covid 19 cases

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?

EXERCISE 3 - SOLUTIONS

1. Irrelevant for the query (31) with the lowest nDCG (label 0)

```
res1 = pipeline(topics.iloc[30:31])
res1.merge(dataset.get_qrels(), how='left')
```

	qid	query	docno	score	rank	label	iteration
0	31	how does the coronavirus differ from seasonal flu	luloic87	21.411621	0	0.0	4

Relevant for the query (42) with the highest nDCG (label 2)

```
res1 = pipeline(topics.iloc[41:42])
res1.merge(dataset.get_qrels(), how='left')
```

	qid	query	docno	score	rank	label	iteration
0	42	does vitamin d impact covid 19 prevention and ...	wr9hkvd3	24.390995	0	2.0	5

EXERCISE 3 - SOLUTIONS

2. Attention analysis

```
text = dataset.irds_ref().docs_store().get('luloic87').abstract[:300] + '...' # truncate text
colbert_factory.explain_text('how does the coronavirus differ from seasonal flu', text)
```

```
text = dataset.irds_ref().docs_store().get('wr9hkvd3').abstract[:300] + '...' # truncate text
colbert_factory.explain_text('does vitamin d impact covid 19 prevention and treatment', text)
```

- q 31 (left): the highest attention to the general tokens ‘corona virus’; there is no relation to flu in the abstract
- q 42 (right): the highest attention to the relevant tokens ‘vitamin d’ and ‘influences’