CS F469 Information Retrieval Assignment 2

Instructor: Dr. Prajna Upadhyay

Deadline: 24 April, 2024, 11:55 PM

This assignment has to be done in groups of 4 people. Read the statement very carefully. Total points: 50

In this assignment, you will implement different ranking models in Python (no restriction on version).

1 Dataset

1.1 NFCorpus

NFCorpus is a full-text English retrieval data set for Medical Information Retrieval. It contains a total of 3,244 natural language queries, with 169,756 automatically extracted relevance judgments for 9,964 medical documents. The corpus contains training, development and testing subsets randomly split on the query level (correspondingly, 80%, 10% and 10%).

Relevance judgments are constructed from the direct and indirect links on the `NutritionFacts.org` website. The most relevant level corresponds to a direct link from a `NutritionFacts` article (query) to a medical article (document) from the cited sources section of a page, the next level is used between a query that links internally another `NutritionFacts` article that in turn links directly a medical document. Finally, the lowest level is reserved for queries and document connected through a topic/tag system on the `NutritionFacts.org`.

The dataset contains the following kinds of files:

1. Queries, or the `*.queries` files, which contain 4 different kinds of queries (`titles`, `nontopic-titles`, `vid-titles`, `vid-desc`). They are divided into training, validation, and test sets. Format is one query per line in a tab-separated format `QUERY_ID` , `QUERY_TEXT`.

2. Documents, located at `raw/doc_dump.txt`. Format is one line per document in a tab-separated format: `ID`, `URL`, `TITLE`, `ABSTRACT`

3. Query-Document relevance values, or the `merged.qrel` file which contains the relevance scores for each query-document pair. There are a total of 4 relevance levels – `3`, `2`, `1`, and `0` (`0` is not in the files), written in the TREC format: `QUERY_ID`, `0`, `DOC_ID`, `RELEVANCE_LEVEL`.

4. `nfdump.txt`: Unfiltered dump of the NutritionFacts.org site as of July 27, 2015. One line per document in the following format: `ID`, `URL`, `TITLE`, `MAINTEXT`, `COMMENTS`, `TOPICS_TAGS`, `DESCRIPTION`, `DOCTORS_NOTE`, `ARTICLE_LINKS`, `QUESTION_LINKS`, `TOPIC_LINKS`, `VIDEO_LINKS`, `MEDARTICLE_LINKS`. Some fields may be empty depending on the content type (i.e., videos, blog posts and Q&A).

1.2 GENA Knowledge Graph

GENA, short for the Graph of Mental Health and Nutrition Association [1], is a knowledge graph designed to illustrate the connections between nutrition and mental well-being, derived from Pubmed[1] biomedical abstracts. The entities in GENA represent names of chemicals, food, and various health conditions. They are connected to each other by semantic binary relations. Triples are listed in a comma-separated ⟨`subject, relation, object`⟩ format. Some triples are:

- ⟨`vitamin K, have disadvantage of, anxiety`⟩
- ⟨`B12 insufficiency, be direct risk factor to, cognitive decline`⟩
- ⟨`Biotin-dependent carboxylases, play crucial roles in metabolism of, fatty acids`⟩
- ⟨`post-traumatic stress disorder, synonym, PTSD`⟩

2 Prerequisites

2.1 Indexing

Use *the linked reference material* to index the documents in `raw/doc_dump.txt` and `raw/nfdump.txt`. Create appropriate document structures as required for more refined querying.

2.2 Evaluation

For evaluating your results, refer to Trec eval script from NIST: `https://trec.nist.gov/trec_eval/` *Linked* guide to use the trec_eval tool.

3 Problem Statement

Complete the following experiments.

1. **Experiment 1**: Indexing the datasets (**Total 3, 1 for report**). Follow section 2.1.

[1] `https://pubmed.ncbi.nlm.nih.gov/`

2. **Experiment 2**: Vector-based Models (**Total 5, 1 for report**). Implement `nnn`, `ntn`, and `ntc` notations.

3. **Experiment 3**: Rocchio Feedback Algorithm for Query Expansion using Pseudo-Relevance Feedback **(Total 5, 1 for report)**

4. **Experiment 4**: Probabilistic Retrieval **(Total 5, 1 for report)**. Implement Language Model and BM-25 ranking models.

5. **Experiment 5**: Entity-based retrieval models **(Total 5, 1 for report)**. Knowledge Graph is GENA (Section 1.2)

6. **Experiment 6**: Query Expansion using Knowledge Graphs **(Total 5, 1 for report)**. You should propose and implement an expansion technique that also considers relations in the knowledge graph.

7. **Experiment 7**: Learning to Rank models **(Total 15, 3 for report)**. You should experiment with at least 1 model for pointwise, pairwise, and list wise approaches. You are free (but not restricted) to use the libraries referred here for your experiments.

8. **Experiment 8**: Anything out-of-the-box that improves your NDCG scores **(Total 7, 1 for report)**

4 What should you submit?

Files to be submitted.

1. A zip file with only code, no indexes or models

2. A report.pdf: Reports not written in latex will attract penalty. Report should document all findings from the experiments.

Checklist for zip file.

1. Your code files.

2. **requirements.txt**: You may include extra libraries for implementation. To be able to run your submission, all your dependencies should be written to a 'requirements.txt' file and submitted. To create your requirements.txt file, run the following command after you have finished implementation:

```
pip freeze > requirements.txt
```

5 Grading Principles

Demos will be conducted for evaluation. Scores will be awarded based on:

1. Whether or not all files are included in the submission (30% penalty is any of the files are missing or if indexes are included)
2. Whether or not your code runs into error (30% penalty)
3. If your code/report is found to be plagiarized (100% penalty)
4. If the report was generated using a tool like ChatGPT (100% penalty for report)
5. Penalty of 10% for not using latex to write report
6. Implementation correctness and viva
7. Report structure and completion of experiments (breakup of points is provided)

References

[1] Linh D. Dang, Uyen T.P. Phan, and Nhung T.H. Nguyen. GENA: A knowledge graph for nutrition and mental health. *Journal of Biomedical Informatics*, 145:104460, 2023.