CS6200 Information Retrieval

Homework5 : Relevance Assessments, IR Evaluation

Objective

In this assignment, you will build upon the previous homework by assessing your vertical search engine. You will collaborate with the same team you established for HW3.

Your task involves evaluating queries relevant to your topical crawl and gathering manual relevance assessments through both your vertical search engine and a web interface.

You will implement information retrieval (IR) evaluation measures, essentially creating your version of treceval. While you can refer to the provided *treceval* code for guidance, you must develop your own implementation.

We have created a discussion thread in Piazza. It is meant to be a supportive space to help each other succeed in this assignment. Whether you're encountering hurdles, have discovered something interesting, or want to share your progress, this is the place!

Assessments and IR Evaluation.

Obtaining queries

Each team will be assigned 3-4 queries specific to the topic you worked on in HW3. The queries are the topics given to you as part of your seeds on your Teams channel.

Assessment graphical interface

To evaluate the relevance of documents, you'll need to develop a web interface that presents the topic/query alongside a list of documents. Each URL in the list should be clickable, leading to the document text either directly from the raw-html field in ES or live from the original URL. A good starting point would be the web GUI used for vertical search in HW3.

The interface must include an input field for each URL/snippet, allowing the assessor to assign a 3-scale grade: "non-relevant," "relevant," or "very relevant" (or 0, 1, 2). Various input methods like checkboxes, radio buttons, dropdown lists, or text boxes can be used. Furthermore, the interface should capture the assessor's ID (by name). Consider adding a "submit" button and a counter to track the number of documents assessed.

The input assessments should be stored in a QREL file in text format as follows:

QueryID AssessorID DocID Grade

QueryID AssessorID DocID Grade

.....

where DocID is the canonical URL, AssessorID is your name (no spaces, like "Michael_Jordan"), Grade is one of {0,1,2}. You can temporarily store information in ES or a database if that is easier for you.

Students within each group can collaorate on the creation of the eval web interface!

An alternative would be to record your assessments in Excel spreadsheets and use them as the evaluator interface. Once all team members have completed their assessments, they will need to be transferred to the QREL file.
You will have to demo your assessment process (the interface and the recording of grades).

Manual assessments.

Each student must manually assess about 200 documents for each topic/query. So if your team has 3 queries, each student will assess 600 documents in total. Each document will undergo three assessments, one per query, assuming there are three team members.

The QREL file should record all the assessments and be placed in your Github repository when you are done.

Write your own trec_eval

Write a program that replicates the functionality of trec_eval. It should take as input a ranked list file and a QREL file, both in TREC format. Both files may contain results and assessments for multiple query IDs.

First, sort docIDS per queryID by score. Then for each query compute R-precision, Average Precision, nDCG, precision@k and recall@k and F1@k (k=5,10, 20, 50, 100). Average these valuess across queryIDs. If run with -q option, your program should display the measures for each queryID before showing the averages.

Run your treceval on HW1 runs with the provided qrel to confirm it gives the same values as the provided treceval.

Run your trec_eval on the HW3 vertical search engine.

Precision-Recall Curves (MS students only)

For each one of the HW4 queries, create a precision-recall plot.

Extra Credit

These extra problems are provided for students who wish to dig deeper into this project. Extra credit is meant to be significantly harder and more open-ended than the standard problems. We strongly recommend completing all of the above before attempting any of these problems.

Points will be awarded based on the difficulty of the solution you attempt and how far you get. You will receive no credit unless your solution is “at least half right,” as determined by the graders.

EC1: applies to the student in the group who creates the web interface

A nice, fast, web interface for evaluation, will get you EC points.

Also EC points will be given for highlighting query terms in the document source when the entire document is displayed.

Rubric

A detailed rubric can be found in Canvas