Search Engines:
11-442 / 11-642

HW2: Ranked Retrieval and Structured Queries
Due Oct 3, 11:59pm

The purpose of this assignment is to gain experience with two popular ranked-retrieval algorithms.

This assignment consists of three major parts:

1. Add the BM25 and Indri ranking algorithms and new query operators to your software from Homework 1 (50%);
2. Use your software to conduct retrieval experiments (25%); and
3. Write a report that describes your software, the experiments, and your conclusions (25%).

The report is an important part of your grade. Leave enough time to do a good job on it.

Software Development

This homework extends the search engine that you developed for Homework 1. Reuse the query processing code developed for Homework 1 (i.e., discarding stopwords, using the same prefix query language, etc).

The search engine for Homework 1 had two retrieval methods (unranked Boolean, and ranked Boolean using term frequency (tf) scores). For this homework you must add two new retrieval methods: i) BM25, and ii) the query likelihood model with two-level smoothing (the method used in Indri).

Query Operators

The BM25 retrieval method must support BM25's implied SUM query operator, as well as the SYN and NEAR/n query operators. The SUM query operator is the default query operator for unstructured (bag of word) queries in BM25.

The Indri retrieval method must support the Indri AND, WAND, WSUM, and WINDOW query operator, as well as the SYN and NEAR/n query operators. The AND query operator is Indri's default query operator for unstructured (bag of words) queries, but note that the score calculation will not be the same as your Ranked Boolean AND.

The SYN and NEAR/n query operators used for Homework 1 can be reused for this homework. The SYN and NEAR/n query operator implementations are identical for all retrieval models (ranked Boolean, BM25, and Indri).

Corpus Statistics

BM25 and Indri require access to corpus statistics, for example, the number of documents and the average length of a particular type of field. Use the `Idx` class to fetch this information from the index. (See the documentation for details.)

```
// Number of documents in the corpus
System.out.println ("numdocs=" + Idx.getNumDocs());

// Information about the url field
System.out.println ("url:\t" +
                    "numdocs=" +
                    Idx.getDocCount ("url") + "\t" +
                    "sumTotalTF=" +
                    Idx.getSumOfFieldLengths("url") + "\t" +
                    "avglen="+
                    Idx.getSumOfFieldLengths("url") /
                    (float) Idx.getDocCount ("url"));
```

The first example fetches the number of documents in the corpus.

The second example fetches the number of documents that have "url" fields, the total number of term occurrences in all url fields, and the average length of a url field.

BM25 and Indri also use the length of a field to normalize term frequency (tf). Use the `Idx` class to fetch this information from the index.

```
System.out.println (Idx.getFieldLength("body", 21));
```

This example shows how to find the length of the "body" field in document 21. Note that the document id is the interal doc id.

Indri requires you to compute $P(q_i|C)$. Implement this value as $P(q_i|C) = ctf(q_i) / length_{terms}(C_{field})$, where C_{field} is the length of the 'field' part of the corpus (e.g., the length of the 'title' part of the corpus). Idx.getSumOfFieldLengths(field) provides that length.

Source Code

Your software must satisfy the same constraints that were described for Homework 1 (e.g., it must be written in Java, and **run properly under Java version 1.8**). If you are extending the software that you developed for Homework 1, **your software probably already satisfies these constraints.**

Input

Your software must accept only one parameter which is a path to a parameter file. Your software must support the same parameters that were described for Homework 1. **In addition**, it must also support several new parameters, as described below.

- **retrievalAlgorithm=** Values are "UnrankedBoolean", "RankedBoolean", "BM25", or "Indri"..
- **BM25:k_1=** Values are real numbers >= 0.0.
- **BM25:b=** Values are real numbers between 0.0 and 1.0.
- **BM25:k_3=** Values are real numbers >= 0.0.

- **Indri:mu=** Values are integers >= 0.
- **Indri:lambda=** Values are real numbers between 0.0 and 1.0.

Output

Your software must write search results in the same way that it did for Homework 1. The only difference in this homework is that your scores will be a little more interesting.

Testing Your Software

Use the HW2 Testing Page to access the trec_eval and homework testing services. You used similar services for Homework 1. **Note:** The link for these services is **not the same** as it was for Homework 1. The requirements for this homework are a little different, so the testing apparatus must also be a little different.

Experiments

You will conduct five experiments that compare different types of retrieval discussed in class. For each experiment, report the information described below.

- Precision at 10, 20, and 30 documents; and
- Mean average precision (MAP).

Experiment 1: Baselines

Compare your Ranked Boolean implementation from Homework 1 to the BM25 and Indri retrieval models. Use the default text representation (body inverted lists). Use bag-of-word queries for the full query set exactly as it was provided, i.e., using only the default query operator for each retrieval method (OR for Ranked Boolean, SUM for BM25, and AND for Indri).

Use the following parameters for the retrieval methods:

- **BM25:** Set the constants to: k_1=1.2, b=0.75, k_3=0.
- **Indri:** Set the parameters to: mu=2500, lambda=0.4 for this experiment..
  Compute $P(q_i|C)$ as $P(q_i|C) = ctf(q_i) / length_{terms}(C_{field})$.
  $length_{terms}(C_{field})$ is provided by Idx.getSumOfFieldLengths(field).

Experiment 2: BM25 Parameter Adjustment

Extend the baselines used in Experiment 1 by modifying the k_1 and b parameters of the BM25 retrieval method. Use your knowledge of how BM25 works, and how changing k_1 or b changes its priorities, giving more or less emphasis to term frequency or documents of a certain length, to guide your choices.

You do not need to explore the entire parameter space and you are not asked to find the optimal settings. However, you will need to explain and justify each choice, so give them some thought.

Use the unstructured queries, and only the default operator for the retrieval method.

Test at least seven settings for each parameter. When you are varying one parameter, hold other parameters constant at their default values.

Experiment 3: Indri Parameter Adjustment

Extend the baselines used in Experiment 1 by modifying the μ and λ smoothing parameters of the Indri retrieval method. Use your knowledge of how Indri works, and how changing μ and λ changes its priorities, giving more or less emphasis to document term frequency, document length, and corpus term frequency, to guide your choices.

You do not need to explore the entire parameter space and you are not asked to find the optimal settings. However, you will need to explain and justify each choice, so give them some thought.

Use the unstructured queries, and only the default operator for the retrieval method.

Test at least seven settings for each parameter. When you are varying one parameter, hold other parameters constant at their default values.

Experiment 4: Different Representations

The Indri retrieval model allows evidence from different representations of the document content (e.g., body, title, ...) to be combined to create a better estimate of how well a concept (e.g., a query term) matches the document. For example, the unstructured query "The Time Traveler's Wife" might be transformed into the following structured query.

```
#AND (
  #WSUM(0.1 time.url      0.2 time.title      0.3 time.inlink      0.4 time.body)
  #WSUM(0.1 traveler.url  0.2 traveler.title  0.3 traveler.inlink  0.4 traveler.body)
  #WSUM(0.1 wife.url      0.2 wife.title      0.3 wife.inlink      0.4 wife.body))
```

Create a version of the query set that uses the representations provided in your index (url, meta keywords, title, body, and inlink). Test five different combinations of weights.

Note: Use the same weights (i.e., the same query template structure) for each query.

Experiment 5: Sequential Dependency Model Queries

The Indri retrieval model is often used with sequential dependency model (SDM) queries. Develop SDM versions of the query set. You may use Don Metzler's perl script to generate sequential dependency model queries, or you may generate them manually (it's very easy). Test five different combinations of weights for SDM queries.

Note: Use the same weights (i.e., the same query template structure) for each query.

The Report and Source Code

You must submit one .zip, .gz, or .tar file to the software test web service that contains a report and your source code. Your latest submission before the homework deadline will be used for grading. When you submit your materials for grading, you do not need to select tests to run. We will run a complete set of tests on your software.

1. **Report:** You must describe your work and your analysis of the experimental results in a written report. A report template is provided in Microsoft Word and pdf formats. Please address all the questions in this template report within the specified section. Do not change the section headings.

   Submit your report in pdf format. Name your report *yourAndrewID*-HW2-Report.pdf. When your uploaded file is unzipped, the report should be in the same directory as your source code.

2. **Source code:** Your submission must include all of your source code, and any files necessary to make and run your source code. Your source code must run within our testing service, so please check that it does before you make your final submission. We will look at your source code, so be sure that it has reasonable documentation and can be understood by others.

FAQ

If you have questions not answered here, see the Frequently Asked Questions file. If your question is not answered there, please contact the TA or the instructor.

Updated on September 22, 2017
Jamie Callan