Dublin City University
School of Computing
CA4009: Search Technologies
Laboratory Session 3 and 4 - step wise details of procedures

December 2015

Module Coordinator: Gareth Jones

Laboratory Tutors: Debasis Ganguly, Piyush Arora

These notes are a supplement to current instructions for Laboratory Sessions 3 and 4. They do not ask you to do any additional work, but intended to give more detail to the procedures to be followed in the existing notes.

1 PART-1

Input: Given a query, example "bone", expand the query by adding terms found within a set of pseudo (assumed) relevant documents. As described in lectures. the purpose of query expansion is to make the query a better representation of the user's information need. Essentially every term appearing in the assumed relevant documents is a candidate to be added to the query to make it a better representation of the user information need. The role of the query expansion algorithm is to determine which ones should be added.

Output: A list of non-query terms ranked based on $ow(i)$ scores as explained below. The top ranked terms from this list are added to the original query as the expansion terms.

Mathematical Formula: To perform query expansion using Robertson's method, you first need to compute the Robertson/Sparck Jones relevance weight $rw(i)$ using the following equation.

$$rw(i) = \log \frac{(r(i) + 0.5)(N - n(i) - R + r(i) + 0.5)}{(n(i) - r(i) + 0.5)(R - r(i) + 0.5)}$$

where: $n(i)$ and N are defined as previously, R and $r(i)$ are new variables.

$n(i) =$ the number of documents term $t(i)$ occurs in, calculated as $N/IDF(i)$
$N =$ the total number of documents in the collection archive, which are 500,000 for our project.
$r(i)$ = the number of **known relevant** documents term $t(i)$ occurs in, $r(i) <= R$. If we assume 10 documents are relevant, then $r(i)$ value let's say for a word "calcium" will be <=10, i.e number of documents out of those 10 assumed relevant which have term "calcium" in them.
$R =$ the total number of **known relevant** documents in the collection archive, for eg. 10

The $rw(i)$ values are then used to compute the Robertson offer weight $ow(i)$ for all terms in the relevant document as follows.

$$ow(i) = r(i) \times rw(i)$$

A series of steps needs to be performed to obtain the output for Part-1.

- Step1: Go to `http://136.206.115.117:8080/IRModelGenerator/SearchServlet?query=bone&simf=BM25&k=1.2&b=0.75&numwanted=10`, you can try different values of $query$, k, b and $numwanted$.
  This assumes that 10 documents are relevant. To assume that a different number of documents are relevant simply set `$numwanted$` tp a different value.

- Step2: Download and store the results of step1. It has 10 documents ($numwanted$), for each document the downloaded file has terms with its term frequency and IDF values, we would use the IDF values to calcualte $rw(i)$.

- Step3: This step can be done in two ways, as implemented by some of the groups in the lab, it uses the file downloaded in step-2.

  - Method-1:
    Goal: Find top 10 non-query unique terms from the whole set of 10 documents.
    Approach:

    - Combine all the terms from 10 documents.
    - Get a unique list of terms as some words like "calcium" might be occurring let's say in 5 documents out of the assumed 10 relevant documents.
    - Store each of these unique terms in a data structure (array list, hash map etc.) for each of these unique terms calculate $ow(i)$ scores.
    - Sort the list based on their $ow(i)$ scores (in descending order i.e terms with highest $ow(i)$ scores comes first).
    - Top 10 non-query terms are your output.

  - Method-2:
    Goal: Find top 3 terms from each of the 10 documents. Merge the terms based on $ow(i)$ scores to get a list of 10 non-query unique terms, from whole set of 10 documents.
    Approach:

    - For each document out of those 10 documents, go through all the terms in a document.
    - For each of these terms calculate $ow(i)$ scores.
    - Sort the terms based on their $ow(i)$ scores (in descending order).
    - Get top 3 terms from each document.
    - Merge top 3 terms from each documents to get a list of 30 terms, remove duplicates.
    - Sort the list based on their $ow(i)$ scores(in descending order).
    - Top 10 non-query terms are your output.

2 PART-2

Input: Given a query, example “bone”, and expanded set of 10 terms “calcium, disease.....” obtained as the output of PART-1.

Output: Evaluate and measure the changes in retrieved results output (P5, P10 and others) when the set of expansion terms are added to the initial query. In this part we have to run trec_eval used for Laboratory Session1.

A series of steps to be performed for getting the output of Part-2.

- Step1: Go to `http://136.206.115.117:8080/IRModelGenerator/res.6.BM25.1.2.0.75`
- Step2: Save top k (eg 10, depending on how many documents you want to evaluate your system) lines in two files eg. ”queriesoutput.txt” and ”expandedqueriesoutput.txt” where each line is in a Trec format:
  301 Q0 **FBIS440260** 1 14.395763 BM25.1.2.0.75
- Step3: Go to `http://136.206.115.117:8080/IRModelGenerator/SearchServlet?query=bone&simf=BM25&k=1.2&b=0.75&numwanted=10` for the case when our query is ”bone” and other parameters (k,b and $numwanted$) as specified in the url.
- Step4: Replace the document ids (such as FBIS440260 and others) in file **”queriesoutput.txt”** by the ranked document ids returned in step 3. This represents the trec_format file with the initial retrieved set of documents for a query.
- Step5: Add the set of expanded terms (obtained in part-1) to the query, e.g. initial query: ”bone”, expanded query: ”bone calcium”
- Step6: Perform search using a modified query.
  Go to: `http://136.206.115.117:8080/IRModelGenerator/SearchServlet?query=bone+calcium&simf=BM25&k=1.2&b=0.75&numwanted=10`
- Step7: Replace the document ids (such as FBIS440260 and others) in the file **”expandedqueriesoutput.txt”** by the ranked documents ids returned in step 6. This represents the retrieved set of documents for a query after performing expansion.
- Step8: Run trec_eval as done for Lab1 using the command ”$./trec\_eval\ qrels\ result\_file$” over queriesoutput.txt and modifiedquriesoutput.txt.
- Step9: Perform analysis of results before and after performing query expansion. Compare standard measure such as P@5, P@10 etc before and after performing query expansion.
- Step10: Report in terms of your detailed analysis with respect to different queries and changing number of relevant documents and no of terms added during expansion.

Caution

For simplicity the above lab sheet has been using the example of query as ”bone”. For the assignment you should use at least 3 queries from the standard TREC collection, topics file:topic.301-450.xml introduced in the first Lab.

Make sure while creating files ”queriesoutput.txt” and ”expandedqueriesoutput.txt” where each line is in a TREC format:

301 Q0 FBIS440260 1 14.395763 BM25.1.2.0.75

you modify same with the correct query-id based on the query you select from topic.301-450.xml for your analysis.

2.1 Lab Report3

This should include the code you write for Part-1. And the results you get for part-2 in Step-9. The details regarding submission can be found in Laboratory session-3 sheet.

2.2 Lab Report4

This should include detailed analysis as done for part-2 in Step-10. The details regarding submission can be found in Laboratory session-4 sheet.