Dublin City University
School of Computing
CA4009: Search Technologies
Laboratory Session 4 & Laboratory Session 5

November 2016

Module Coordinator: Gareth Jones

Laboratory Tutors: Piyush Arora, Wei Li

1 Introduction

Laboratories 4 and 5 extend the investigation of the topic of ranked information retrieval (IR) studied in laboratories 1-3 to examine relevance feedback and query expansion. The laboratories focuses on *pseudo* relevance feedback as introduced in lectures using the Robertson offer weight ($ow(i)$) for the probabilistic model of information retrieval.

Laboratory 4 focuses on the implementation of relevance feedback and Laboratory 5 on an experimental investigation of the effect of applying relevance feedback in information retrieval systems.

You should make a single submission of a report describing your work in Laboratories 4 and 5 and the code you develop at the end of Laboratory 5, with a final submission deadline of the start of the Laboratory 6 as before.

2 Laboratory 4: Query Expansion using Relevance Feedback

Relevance feedback (RF) in information retrieval seeks to improve retrieval effectiveness. The first stage in the RF process is to mark documents retrieved in an initial run as relevant, and in the next stage query expansion terms are selected from these relevant documents and added to the initial query.

As described in lectures, document relevance information can come either directly from user feedback, from observing user behaviour (e.g. which documents they click on), or by assuming that the top ranked retrieved documents are relevant (the pseudo relevance feedback approach). This laboratory takes the pseudo relevance feedback approach, i.e. in your investigations you will assume that a number of top ranked documents from the previous run with the current query are relevant.

2.1 Robertson's Offer Weight $ow(i)$ method

RF using the probabilistic model of iIR was introduced in lectures. This section summarises the details of this method.

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 in the previous laboratories, R and $r(i)$ are new variables.

$n(i) =$ the number of documents term $t(i)$ occurs in,
$N =$ the total number of documents in the collection archive.
$r(i) =$ the number of **known relevant** documents term $t(i)$ occurs in,
$R =$ the total number of **known relevant** documents in the collection archive.

In pseudo relevance feedback the value of R is assumed as part of the operation of the algorithm. For example, if you assume that the top 5 ranked retrieved documents are relevant the value of R = 5. The value of $r(i)$ for each term i which occurs in one of these assumed relevant documents can then be computed based its occurrences in these documents. It should be obvious to you that $r(i)$ will always be $\leq R$.

The value of $rw(i)$ for each term i can then be used to compute the Robertson offer weight $ow(i)$ for each term i term as follows:

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

where $r(i)$ has the same meaning as above.

2.2 Expanding the Query

All the terms occurring the top R ranked documents are potential query expansion terms. The purpose of $ow(i)$ is to determine which ones are likely to be the most effective. That is, which ones when added to the query are likely to improve its effectiveness in retrieving relevant documents at a higher rank in a subsequent retrieval run for this query.

To decide which terms to add, the terms are ranked in decreasing order of $ow(i)$. The top ranked terms from this ranked list are then selected as the expansion terms to be added to the original query.

2.3 Implementing Query Expansion based on $ow(i)$

In addition to the name of the term and the term frequency in this document, the search interface used in the earlier laboratories also shows the value of $(N/n(i)$, where as defined above, N is the total number of documents in search collection and $n(i)$ being the total document frequency of a term i in the collection.

For this investigation, you are need write a stand alone program in java (or any other programming language of your choice) which will be able to make a HTTP request to

`http://136.206.115.117:8080/IRModelGenerator/SearchServlet` with appropriate parameters (query, numwanted, simf, k1 and b).

where "query" is words to be used in the query and "numwanted" is the number of ranked documents to be returned, and "simf" is the query-document matching function to be used. For the simf, k1, and b parameters, you should use the values `BM25`, `1.2` and `0.75`.

An example URL with the parameters is as follows:

```
http://136.206.115.117:8080/IRModelGenerator/SearchServlet?query="bone disease"&
simf=BM25&k=1.2&b=0.75&numwanted=50
```

This web service returns information about each of the "numwanted" retrieved documents.

In order to operate query expansion using the above method, you need to parse the necessary information from the details of the returned relevant documents to compute $uw(i)$.

After you decide how many documents you are going to assume to be relevant for pseudo relevance feedback (i.e the value of R) (a good number to start with is 10), you need to parse and extract out pertinent information from the 10 top ranked documents.

You already know the value of R, but you need to $r(i)$ based on the presence of each term i in the top R documents.

To calculate $rw(i)$, you also need to know N and $n(i)$. N is the total number of documents in the collection, which is approximately 500,000 (the actual value of N isn't exactly 500,000, you should consider why using a number of N which is $\approx N$ is OK). To calculate $n(i)$ for each term i of interest, you can use N with the returned value of $N/n(i)$.

You then need to calculate $ow(i)$ each term i using your calculated values of $rw(i)$ and $r(i)$, and to rank these terms by $ow(i)$.. You can then add a number of these top ranked terms to the original query (again a good number of start with is 10).

Prepare a report for this laboratory including a description of how you implemented each of the above stages illustrating these with examples from your code. You should submit the itself as a separate file.

3 Laboratory 5: Experimenting with Query Expansion in IR

Once you have implemented QE using the Robertson $ow(i)$ method, in laboratory 5, you should investigate how it behaves when used using the TREC test collections from the earlier laboratories.

Carry out a new run with your initial expanded query and compare with $R = 10$ and 10 expansion terms. As before, compute the IR metric values using trec_eval. Then compare the metric values for the original query and the expanded queries. Note the changes that occur in the ranking of the retrieved documents and examine the documents retrieved to try to explain why you think these changes have happened.

Include results for at least 3 TREC topic "Title" field queries with results for the original and expanded queries with 10 documents assumed relevant and 10 expansion terms in your report, including the identities of the queries and the original and expanded queries.

You should next experiment with alternative values (at least 2 different numbers of assumed relevant documents, for each of which you should try at least 4 different numbers of expansion terms) for the number of assumed relevant documents and the number of expansion terms comparing the averaged values for the TREC topic sets.

Include your results in your report, comment on what you observe in terms of the best values for the number of assumed relevant documents and the number of expansion terms used. Again try to explain the difference in behaviour that you observe comparing the original and expanded queries.