# CSCI 6370 IR and Web Search Project

## Part 4: Query Reformulation

### Due date: Monday, June 29 @ 11:59 PM.

## Problem Description:

I congratulate you that you have completed the first three parts of your search engine! You have already had a basic search engine running. What you have completed is a solid chassis for you to add more and more additional components to make your engine running to your (your clients’) satisfaction. There are so many you can do, but we have only two weeks left. For the sake of good experimenting and completing a working engine, I ask you to work on two separate tasks: querying reformulation, and content recommendation. Ideally, we should explore the link structures. However, if you are careful enough, you might have realized that the given web page corpus does not have links that are profound enough to be exploited for good benefits.

For Part 4 of our project, we’ll work on query reformulation. For Part 5, we will work on content recommendation.

As we have learned that query reformulation is a part of relevance feedback for query operations. In the ideal situation, users shall provide accuracy feedback to the search engine to help it reformate the search query so as to find more accurate results. In reality, we cannot expect users to do so. Therefore, we have to work on query reformulation without receiving relevant feedback from users. But how can we do this? Again, there are many ways. For the sake of simplicity and feasibility, we will work on index term correlation analysis.

Below is our approach toward query reformulation without direct relevant feedback from users.

## Algorithm:

1. For any query q, use our basis search engine to find a set of S relevant pages. Do not display pages in S.
2. Select a list of top ranked pages A from S. (Say, A has 5 pages.)
3. Extract keywords from A, save into a list K.
4. Find correlation values between keywords in q and keywords in K. Choose a short list of keywords form K that are most correlated to keywords in q. (Say, 3 to 5 keywords.) Reformulate q to include those selected keywords.
5. Use to the reformulated q to search for a new set of pages S’.
6. Correlated pages in S and S’ and finally display the pages to the user.

Note 1: For sake of evaluation, please display pages in S and S’ with different colors.

Note 2: Please do not worry about whether pages from S’ are “better” than pages in S. Our purpose now is to carry out the experiment. When sufficient time is available, we can find ways to improve the results.

## Your Task

Implement the above query reformulation algorithm for your search engine.

## Some Revision on Inverted Indices

Like what is shown below, we now require that the inverted index for each keyword is sorted by the document id’s.



The sorted order will help us to computer correlation value of any two keywords quickly and easily.

Let W be the Document by Term matrix. Mathematically, to compute the correlations between keywords, we need to compute

This matrix multiplication is practically infeasible. My solution is as follows:

For any two keywords, for example, “cat” and “hat”, to compute corr(“cat”,”hat”), we just need to walk down indices to “cat” and “hat” in parallel to sum up of products of tf-idf values for same documents.

Still not clear? I’ll explain later.

## Some Revision on Document List

For the new document list, for each document, I would like to add a list of keywords (term and tf-idf values, preferably normalized) appearing in the document. See the design example below.



You can create this keyword list by walk through the inverted index once.

I shall explain more, if this is not clear.

## Add a New Document Correlation List

Note: this is for Part 5, the final part of your project. But, if you’d like to work ahead, you have it now.



We can use two nested loops for the document list to compute all document correlation values.

We will use a cutting off threshold to filtering out document in every correlation list, if the correlation values are below the threshold.

Please note that we have kept trying to avoid computing (for keyword correlations) or (for document correlations).

## How to submit your work?

Submit your python codes to Blackboard before the deadline.

One group only needs to submit one copy. Please make sure to include group members’ names and ID’s in your submission.

## Grading

The TA would do test runs of your program to check whether it works.