# CSCI 6370 IR and Web Search Project

## Part 4: Content Recommendation, and Completion of the Project

### Due date: Monday, July 6 @ 11:59 PM. This is deadline is FIRM!

## Problem Description:

I congratulate you that you have completed the first four parts of your search engine!

This is the last part of the project. You are so close to the completion. So, keep up your good work!

Like Part 4, here we will work on adding another good component to your search engine: content recommendation. In the ideal situation, we’d like the users to help you out, by giving us relevance feedback or related sorts of information needs/preferences. However, in reality, those good deeds are not easily available, and we cannot expect users to help us much. Therefore, we have to work on solutions without receiving direct 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 document correlation analysis.

Below is our approach toward content recommendation 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. Search from the document correlation list for pages that are closely correlated to pages in A. Let S’ denote the set of those pages.
4. 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 content recommendation algorithm for your search engine.
* Wrap up your project. You have two options:
  + Option 1:
    - Submit your codes to BB or Oscar if needed.
    - Explain to Oscar how to run your codes if needed
    - Oscar will test to see how well your search engine works
  + Option 2:
    - Deploy your search engine to the cloud, if you comply with the following conditions in order to avoid any potential intellectual property issues
      * **Condition 1**: You must not archive nor disseminate the web pages from “www.netfunny.com” to your cloud site.
      * **Condition 2**: Your search shall direct all the search result links to “[www.netfunny.com](http://www.netfunny.com)”. To do so, for every url following your local directory path, strip off the leading “rhf/”. For example, for a link

“rhf/www.netfunny.com/rhf/jokes/00/Jan/aol.html”

change it to

“www.netfunny.com/rhf/jokes/00/Jan/aol.html”

**Acknowledgement**:

I thank Erik Enriquez for sharing his idea pertaining to Option 2 to me. I think his idea is good and valuable.

## Add a New Document Correlation List

In order to compute document correlations, I suggest you add a Document Correlation List as shown below. This correlation list will be computed as a part of the inverted indices.

We can use two nested loops for the Document List to compute all document correlation values. We will use a cutting off threshold to filter out documents 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.

If you choose Option 2, please submit the **URL** of your search engine.

## Grading

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