

# Illinois Institute of Technology

## CS429: Introduction to Information Retrieval Fall 2005

Instructor: Dr. Nazli Goharian (nazli@ir.iit.edu)

Teaching Assistant: Alana Platt (platala@iit.edu)

### Project Part 2: Query Processing

**Date Given:** Sept 22<sup>nd</sup>

**Due Date:** Oct 18<sup>th</sup> - by no later than 3:15pm on blackboard digital box & also a hard copy of all deliverables but the code.

---

**Grading:** This assignment is 10 points out of the total 35 points allocated for all assignments in the semester. It will be graded on the scale of 100.

*NOTE: You can use IRIS or use any programming language to implement this assignment.*

#### OBJECTIVE

By now you have learned how the initial phase of a search engine is done. You have learned to pre-process a text collection to create an index. In this process you learned how the tokenizer work to identify terms. You learned how you could incorporate stemming of the terms into the system.

Now, it is the time to work on the query-processing phase of a search engine. In this part of the project, you will apply different strategies and different similarity measures to perform relevance judgment. Furthermore, you will apply one of the utilities in information retrieval, namely, Relevance Feedback, to experiment and observe the effect of such utility in the accuracy of the search.

You are asked to use the standard evaluation benchmark in information retrieval, provided by TREC, “treceval”, to evaluate the accuracy of your search engine.

#### REQUIREMENTS

The assumption is that there is enough memory to store your entire inverted index. The system requirements for this phase of the project are:

**Inverted Index:** Your system should include a memory-based inverted index. Do not store Stop Words in the lexicon. Use the *big collection* provided on the class site.

**Queries:** Enable your application to read a list of queries from a file. You need to download the “query file”. These queries are tagged and need to be pre-processed same as you did for the documents to identify the query terms. You can either modify the existing query parser or create a new parser for this purpose (your choice!). For those experimentations that you are required to stem document terms, you need to make sure that you also stem query terms at the time of query processing. Naturally, the stemming rules for both documents and queries should be the same! Thus, use Porter stemmer in both cases. Take care of Stop Words by eliminating them from queries. It makes sense to use the same Stop Word list for both documents and queries! Note that the queries are identified by their unique numbers, as they are each numbered; and you should only use the *title* part of the queries for retrieving documents. Note that you need to identify Special Terms in queries in the same way that they are identified in the documents.

**Relevance Ranking:** Using different information retrieval strategies and similarity measures, perform the query processing to identify the relevant documents and obtain relevance ranking. For this, you need to execute all the queries from the “query file” with the specified strategy and similarity measures listed below (see a, b, c, d, e).

- a) Vector Space Model *dot product*
- b) Vector Space Model *Cosine*
- c) Vector Space Model *Pivoted cosine Normalization*
- d) Vector Space Model *Pivoted Unique Normalization*
- e) Probabilistic model.

**Choice of Term Weight:** Use the following term weight in your experimentations for both the document term weight and query weight (unless any of the above (a-e) have its own term weight such as in d and e:

$$w_{ij} = \frac{(\log tf_{ij} + 1.0) * idf_j}{\sum_{j=1}^t [(\log tf_{ij} + 1.0) * idf_j]^2}$$

**Utilities:** You need to apply Relevance Feedback utility on strategies (b), (d) and (e).

Use top 5 retrieved documents for each query using Relevance Feedback. Get all the terms (stems, special terms) from the top 5 retrieved documents (i.e., the 5 documents that are scored highest in the relevance ranking); sort these terms (stems, special terms) by their document frequency in the top 5 documents. Take top 2 terms (stems, special terms) and add them to the existing query and execute the query one more time (note: the top two terms are the top two besides the existing query terms in a given query-i.e, if query has 3 terms then adding 2 more terms makes the new query a 5 term query). Relevance feedback is to be done for similarity measures b, d, and e. To find the “good” terms use a sort criteria that gives you the best result This worth to try: using the product of document frequency of the term in the top docs and the term inverse document frequency).

## **RUNS & REPORTS**

You are asked to do experimentations and gather statistics and provide reports as described below. Here are some explanation on what you need to consider for your experimentations and what you are expected to report:

1. Identify the top 100 retrieved documents with their relevance ranking scores for each query using similarity measures listed above (a, b, c, d, and e). You have to output the entire top 100 retrieved documents for all queries and their scores in the format specified in “output file” This output file will be used as the input to *treceval* software to generate Average Precision (see 2).
2. Obtain *average precision* for your similarity measures using *treceval*. A description of the format and how to use *treceval* is given in the link to *treceval*.

**Report 1:** This report gives a comparison in the accuracy of the search among different strategies and similarity measures you are using for this project part. Provide the following report as specified below. The report is based on the result of all queries together. The average precision for each query for each of the similarity measures at the different points of recall is initially calculated by *treceval*. *Treceval* averages all the average precisions of different queries. The number below is the average of all average precisions for each similarity measure. Number of runs are 7: a, b, b with Relevance Feedback, c, c with Relevance Feedback, d, e. Note: No Stemming and phrasing is used for this report. Fill out the given table with your result and submit. You also need to submit 8 *treceval* summary pages; one summary page for each similarity measure. The summary pages are generated also by *treceval*. If you modify any parameter, to achieve a better accuracy, explain that.

| Similarity Measure              | Average Precision w/o Relevance Feedback | Query Processing Time (sec) | Average Precision with Relevance Feedback | Query Processing Time (sec) |
|---------------------------------|------------------------------------------|-----------------------------|-------------------------------------------|-----------------------------|
| a) Dot Product                  | X                                        | X                           | -                                         |                             |
| b) Cosine                       | X                                        | X                           | X                                         | X                           |
| c) Pivoted Cosine Normalization | X                                        | X                           | -                                         |                             |
| d) Pivoted Unique Normalization | X                                        | X                           | X                                         | X                           |
| e) Probabilistic                | X                                        | X                           | X                                         | X                           |

**Report 2:** On a new page, for the query number 304

<num> Number: 304

<title> Topic: Endangered Species (Mammals)

Output the top two relevance feedback terms. Include their document frequencies in the top 5 documents for this query when using the vector space model (dot product). The format of this report should be as follows (with your own values). Note: No Stemming and phrasing is used for this report.

| term | Frequency |
|------|-----------|
| T1   | 18        |
| T2   | 15        |

**Report 3:** This report gives a comparison in query processing timing and accuracy of the search by alternative use of stemming and/or phrasing, with or without applying relevance feedback (RF). For this report use your results based on vector space model with *Cosine Pivoted Unique Normalization*, i.e., (d).

| Utility                          | Time for Query Processing in seconds | Average Precision |
|----------------------------------|--------------------------------------|-------------------|
| no stemmer, no phrases ; w/o RF  | X                                    | X                 |
| stemmer, no phrases ; w/o RF     | X                                    | X                 |
| no stemmer, phrases ; w/o RF     | X                                    | X                 |
| stemmer, phrases ; w/o RF        | X                                    | X                 |
| no stemmer, no phrases ; with RF | X                                    | X                 |
| stemmer, no phrases ; with RF    | X                                    | X                 |
| no stemmer, phrases ; with RF    | X                                    | X                 |
| stemmer, phrases ; with RF       | X                                    | X                 |

### Deliverables

**All students must submit on the blackboard. The copy submitted on Digital Drop Box should be a single zipped file that includes summary; design document; results; readme; and the source code ready to compile and run.**

1. Summary (10%): include the following:
  - a. Status of the project, complete/incomplete. If incomplete state what is incomplete. Failing to state the exact status or false or misleading statement will result a zero for the entire assignment.
  - b. Time spent on the project. - number of hours.
  - c. Problems encountered - List at least 3 to 4 biggest problems that you encountered while you were working on the project and how you solved them.
  - d. Things you wish you had been told prior to being given the assignment.
  - e. Observations- any interesting observation that you made about the runs described above.
2. Design Document (20%): The design document should be written prior to coding. There should **not** be any code in your design document. The goal of your design document should be for any programmer to be able to implement the project in any language using only your design document as a reference.
3. Code (10%): The portion of your code that relates to this part of the project. No extra piece of code should be submitted. You should also submit a Readme file explaining how to compile and run your tests.
4. Results (60%):
  - a. Report 1 and TRECEVAL report of each run.
  - b. Report 2
  - c. Report 3