R·I·T

Rochester Institute of Technology
Golisano College of Computing and Information Sciences
School of Information (iSchool)

Lab 1
Inverted Index Construction and Query Processing

Overview

This lab consists of three major tasks:

- Processing a number of text documents to generate indexed terms.
- Building an inverted index for the terms and documents.
- Formulate and process queries using the constructed inverted index.

Resources

- You should have read Chapters 1 and 2 of Introduction to Information Retrieval.
- Carefully read the lecture examples of Weeks 1-3 and understand the technical details.
- Go over the lecture notes for Weeks 1-3.

Task 1: Inverted Index Construction (35 points)

In this task, you will read in the five text documents located in the folder in Lab1_data.zip. You need to create an inverted index using this document collection by performing the following steps:

1. Assign a unique id to each text document, i.e., 1-5.

2. Read in the text in each document and perform tokenization. Treat punctuation (e.g., ". & % $ # ! /"), symbols (e.g., "+-*/"), and spaces as delimiters.

3. Adopt a proper data structure to store the stop word list found in stopwords.txt and use it to efficiently remove all the stop words in the documents.

4. Call Porter's stemmer to perform stemming.

5. All the remaining tokens will be treated as terms in the dictionary. Documents that contain the term should appear in the postings list for the term.

Task 2: Query processing (65 points)

In this task, you need to implement search algorithms that use the constructed inverted index to perform the following queries.

1. **(5 points)** Implement a search algorithm that can handle a query with a single keyword. *Design two test cases and show the document name(s) as the search result*.

2. **(10 points)** Implement a search algorithm that can handle a query with two keywords. Assume that query terms are connected using the AND operator. As an example, a query "information technology" means "information AND technology". *Design two test cases and show the document names as the search result*.

3. **(20 points)** Implement a search algorithm that can handle a query with two keywords. Assume that query terms are connected using the OR operator. As an example, a query "information technology" means "information OR technology". *Design two test cases and show the document names as the search result*.

4. **(30 points)** Implement a search algorithm that can handle a query with *three or more keywords*. Assume that query terms are connected using the AND operator. As an example, a query "Rochester Institute Technology" means "Rochester AND Institute AND Technology". *Design two test cases and show the document names as the search result*. The query should be optimized so that shorter postings lists will be processed first. *It is also necessary to show the order in which these keywords are combined*. Using the same example, if your algorithm processes the keywords in the order of "Rochester", "Technology", and then "Institute", it should be shown as
    1. Rochester
    2. Technology
    3. Institute

Lab Submission Instructions – Create a PDF document showing all of your test cases for Task 2 and the resultant output. Your output should also show the postings for each term involved in a given query so it's obvious that your output is correct. Create a zip file called Lab1.zip that includes the PDF document with your results, the Lab1_Data folder with all documents contained therein, and your code and submit it to the Lab 1 dropbox prior to the due date. **Please do NOT use file paths that are specific to your machine. Your code should include a hard-coded path to the Lab1_Data directory, which will be in the current directory where the code resides. Failure to do this will result in significant point loss.**