CS630 Representing and Accessing Digital Information
Fall 2004
Assignment 1

Due at the beginning of class, on Wednesday, September 15

Overall Goal

Implement a retrieval system that performs full-text indexing. Use the following design constraints:

Indexing: Use words as atomary indexing units. Define words as strings that are separated by whitespace. Punctuation characters are considered whitespace as well.

Query Language: Queries are conjunctions of words: w_1 AND w_2 AND ... AND w_k.

Retrieval Model: The system should return the set of documents that satisfy the query. In this context, a document satisfies a query, if all query words occur in the document.

Datastructures: The system must use an inverted index. To construct and access the inverted index, you can use any type of datastructure that allows faster than linear time access (e.g. hash table, trie).

To keep the programming simple, you can make the following simplifications:

In-Memory: You can keep all datastructures in memory and there is no need to write them to a file.

User Interface: You do not need to implement a user interface. You can write your test queries directly into your source code.

Existing Implementations: You can use existing implementations of datastructures. This means that you do not need to implement hash-table or tries from scratch. Make sure you clearly identify the places where you made use of other people's work and reference your source appropriately.

Programming Language: You can use your preferred programming language.

A corpus of 29890 documents is available from the course homepage. The documents are abstracts of physics articles. Each documents has a number as its unique identifier. Use this corpus for your retrieval system. The format is very simple.

Task 1

Implement the basic retrieval system. Print the source code and hand it in as part of your homework.

Task 2

Name and discuss the design decisions that you made in your implementation. This should include your choice of datastructures and indexing.

Task 3

How many different words occurs in the collection at least once?

Task 4

Run your retrieval system to answer the queries

- "galactic"
- "center"
- "galactic AND center"
- "center AND galactic"

For each query, report the number of documents that match the query.

Task 5

From the inverted index, it is easy to generate a histogram of how many documents a particular word occurs in. Make a log-log plot of this histogram with rank on the x-axis and number of documents on the y-axis. Argue whether Zipf's law holds for this data.

Task 6

Extend your basic retrieval system by one piece of added functionality. This can, for example, be an improve indexing scheme, a more expressive query language, a different retrieval model, etc. Demonstrate the extension using examples. Again, include the source code with your solution.