**ASSIGNMENT - 1**

**DESIGN AND IMPLEMENTATION OF**

**DOMAIN SPECIFIC INFORMATION RETRIEVAL SYSTEM**

**BY**

**N IMAYAVALLI**

**2015A8PS0444H**

**https://github.com/SupremeFlash**

**Problem Definition:**

* Implementation of a text based information retrieval system
* A domain specific search engine
* The searching module is built according to the Vector Space model of Information Retrieval.
* The program builds the index of all the files, then retrieves all the documents relevant to queries input by user.
* This program performs ranked retrieval using the tf-idf scoring model.

The indexer program needs to be run just once for a file system.

The runner program needs to be run every time a query needs to be processed.

Single and multiple term queries have been handled using the tf-idf indexes.

Phrase queries are yet to be implemented, even though postingslist building is done.

Postingslist, tf-idf

Pickle

The task is to build a search engine which will cater to the needs of a particular domain. You have to feed your IR model with documents containing information about the chosen domain. It will then process the data and build indexes. Once this is done, the user will give a query as an input. You are supposed to return top 10 relevant documents as the output. Your results should be explainable. The design document should clearly explain the working of the model along with detailed explanation of any formulas that you might have used.

Domain I have used : Wikipedia Articles

1. Stemming:

3

a. Martin Porter’s ‘Porter Stemmer’ can be used for this purpose. Implementation in multiple languages can be found in the above link.

2. Tokenization:

a. For this step you can use any standard tokenizer or inbuilt package. Following are a few sources:

i. Python’s NLTK package.

description of the application’s architecture along with the major data structures used in the project. Precision and Recall, if possible, should also be calculated. Running for all the preprocessing should be mentioned. Also mention the running time for search or retrieval.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tokenization and Normalization | | 5 | | |
| 2. | Efficient usage of Data Structures with justification | | 10 |
| 3. | Index Construction | | 5 |
| 4. | Accurate Data Retrieval | | 5 |
| 5. | Viva | | 10 |
| 6. | | Novelty / Out-of-the-box thinking (Anything that is not  covered in the lectures.) | | |

MY DATASET:

I have used the corpus 'Wiki Small' from:

the book "Search Engines: Information Retrieval in Practice"'s webpage:

http://www.search-engines-book.com/collections/

The corpus was created from a snapshot of the English Wikipedia downloaded from static.wikipedia.org in September 2008.

The corpus contains 6043 articles as .html files.

which I extracted as .txt files.

The .html files are categorized according to each consecutive letter of their filenames,

but in my ExtractedText folder, I have removed the categorization and just collected all the files together.

**Algorithm used: Pseudo Code**

**Enhancements:**

RUNNING TIME :

Indexer.py

Runner.py

PRECISION :

RECALL :