Text Based IR Search Engine
README

Description:

Implemented a search engine on the wikipedia dump of size 75 GB. In order to retrieve results faster and relevant, indexing and ranking is implemented. Relevance ranking algorithm is implemented using TF-IDF score to rank documents. Creating an index takes a long time on a given wikipedia dump. Result is retrieved in less than 1 second.

Prerequisites

python3 For preprocessing and Stemming, nltk (Natural Language toolkit) library. To install nltk pip3 install nltk Install etree to parse wikipedia dump xml file To install etree pip3 install elementpath stop_words.txt file must be present in the same directory to remove stop words. I have used an autocorrect library to correct the spelling.

To run the project:

Change wikipedia dump file path in create_index.py file
Change index file path to store index in create_index.py file
Command to run create_index file
python3 create_index.py
Change the file path of an index file to load index in the memory in the search file.
Command to run search file
python3 search.py
IR_Final_Project folder contains html css and main for UI interface.
Run main.py file and switch to a given local IP address and search Query.

Format of the query:

It supports two types of query.

Normal query e.g. new york , gandhi , 1981 world cup

Field query e.g. title:gandhi body:arjun infobox:gandhi category:gandhi ref:gandhi

Top 10 results will be printed.

What feature search engine support:

1. Search query in less time.
2. Spelling correction
3. Multi word query search supported.
4. Query based on particular fields such as title, body,infobox,category etc supported.
5. Result top 10 document

Datasets used:

The dataset used for our Project can be downloaded from here.

The dataset used for comparing our project with the baseline can be downloaded from here.

Output files from indexing here