CSE 108 Project: SIMPLE SEARCH ENGINE

Introduction

Anteros Search Engine is a simple search engine to crawl various website and bring the most relevant search results. It is designed with a simple interface to make it very simple for users. It is fast, effective and shows most relevant results in response of a user query.

Functionalities

1. Very easy and simple user interface.
2. Almost accurate and most relevant search result.
3. Fast search result. Around 100ms for a 10 word sentence in a database of 50000+ wikipedia page.
4. Crawler can be stopped when needed and if I start the crawler again it will begin from the last place I stopped it.
5. To increase accuracy, diagram frequency along with monogram frequency was used. Some words are more important these are extracted from the website name and given their frequency 30 for monogram and 10000 for diagram which made the engine more accurate.

Functions That Can be Added:

1. Currently the engine is ready for only English language, but it can be extended for multilanguage with a few line of codes.
2. Currently textfile and binary file is used to keep index. But Further some database system can be used which will make the engine fast and ready for millions of websites.

3. Query autocomplete feature can be added.
4. An additional dictionary can be added so that the engine can fix user mispronunciation
5. If the user enters ‘capital of dhaka’/’capital:of,..dhaka’ the search engine will give accurate result but if the user enters ‘capitalofdhaka’ then the engine will not give accurate result because it will think ‘capitalofdhaka’ a single word. But this can be fixed for each i (0<=i<l), j(i<j<l) if we consider str(i,i+1…j) as a single word with O(l^2) added complexity. But it will be ignorable because of word length, l can not be that big.

Problems Faced in This Project And Their Solution:

1. Major problems were faced in making a crawler:
    1. Crawling same website more than once. I fixed it by using a SET structure where I put all the websites crawled so far.
    2. Crawling a dynamic website. Actually it has not fixed. However most of the newspaper site are of these kind. So a separate check list can be created to update this kind of site.
    3. In parsing The challenge is that there is always strange markups, URLs etc. in the HTML code and it’s hard to cover all corner cases. But fortunately java’s build in library could handle most of the problems.
    4. Since the index database was for monogram and diagram, I had to make sure that no other symbol was added because it would make the query result poor.
    5. I had to use try catch block accurately so that any thread does not stop unintentionally.
    6. Multiple thread was used so that crawling could be fast.
2. In search program a major problem was how to rank the wabsite. For this I used value of a website as : monogramfreq*1 + diagramfreq*100. Frequency of a word in a website name was given 30 for monogram, 10000 for diagram. It made the search result most relevant. Most valued website was ranked first.

User Manual:

1. To start Crawling run CoolCrawler.java in mysearchengine package

2. To start searching run MySearchWithGui.java in AnterosEngine package.
3. Before stopping CoolCrawler newly added website are not written in any file.