

Introduction

Search engines is a tool used to look for information on the internet. They work by retrieving and ranking content based on user queries. The main goal of a search engine is to provide relevant and accurate results in response to user input. A search engine is a complex tool requiring multiple components.

Basic Principles of a Search Engine

1. **Crawling:** This is the process by which search engines discover and index web pages. A crawler systematically browses the web to collect content from websites. It follows links from one page to another and downloads the content. Crawlers are responsible for gathering large amounts of data from the internet, which later gets indexed.
2. **Indexing:** Once a page is crawled, the content is analyzed and stored in a massive database known as the index. The index is a structured collection of all the content a search engine has gathered. It includes keywords, metadata, and the relationships between various web pages. Indexing helps search engines retrieve relevant pages when a user enters a query.
3. **Ranking:** The search engine needs to rank the results. Ranking is determined by several factors, including the relevance of the content, the quality of the web page, and how well the page matches the query. Algorithms such as Google's PageRank consider various signals, such as the number and quality of backlinks, page content, user engagement, and more.
4. **Results** After ranking the results, the search engine presents them to the user in a list, typically with a title, a snippet of the content, and a URL. The results are sorted by relevance, with the most relevant results appearing at the top.

Methodology

The required tools and steps are described below:

At first all the required functions were imported

Content.csv and graph.csv are selected and showed

Graph was created from graph.csv and content was loaded from content.csv

Networkx library was used to create the graph

It was then visualized using matplotlib

Nltk library was downloaded and used for tokenizing the contents and removing articles, prepositions and conjunctions.

An inverted index was created

Pagerank was used for ranking webpages.

Results:

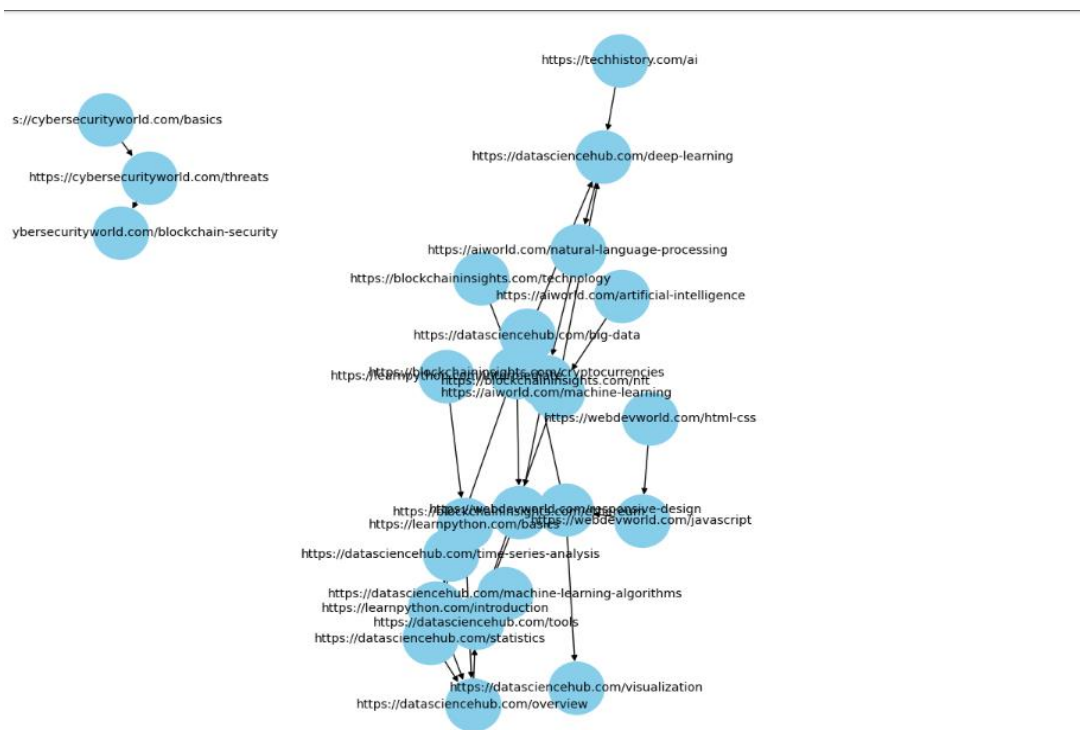
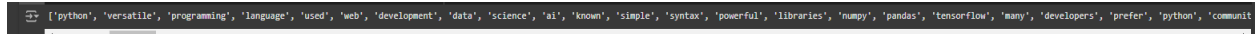
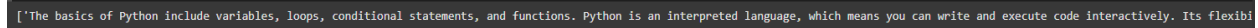


Figure: Graph Visualization.

A horizontal bar chart showing token frequencies. The tokens are listed in descending order of frequency. The first token is 'python', followed by 'versatile', 'programming', 'language', 'used', 'web', 'development', 'data', 'science', 'ai', 'known', 'simple', 'syntax', 'powerful', 'libraries', 'numpy', 'pandas', 'tensorflow', 'many', 'developers', 'prefer', 'python', 'communit'.

```
['python', 'versatile', 'programming', 'language', 'used', 'web', 'development', 'data', 'science', 'ai', 'known', 'simple', 'syntax', 'powerful', 'libraries', 'numpy', 'pandas', 'tensorflow', 'many', 'developers', 'prefer', 'python', 'communit']
```

Figure: Token Result.

A single line of text representing a search result: "The basics of Python include variables, loops, conditional statements, and functions. Python is an interpreted language, which means you can write and execute code interactively. Its flexibi".

```
['The basics of Python include variables, loops, conditional statements, and functions. Python is an interpreted language, which means you can write and execute code interactively. Its flexibi']
```

Figure: Single Search Result.

Conclusion

This search engine uses PageRank to rank pages while also processing content efficiently using natural language processing tools. This design while not perfect gives us a good understanding on search engine and its approach. Using the right tools and method we can make a functional search engine.

Github Link : <https://github.com/SofiqulSohan/SearchEngine>