**Medi Crawl: A search engine for diseases**

**1. Introduction**

**1.1 What is Web Search Engine?**

Internet is our everyday source for all kinds of information, like news, broadcast, music, pictures, or even movies. We are immersed in the flood of big data every day. However, the information we need is only a small portion of it and for different individuals the importance of information is hardly the same. A search engine will make it convenient for us to absorb information efficiently. The search engine will do the MapReduce for us humans to get the useful pages.

**1.2 Crawling & Indexing**

Search Engines have automated robots called crawlers that use links to scour the Internet, find web pages, and decipher page data that are indexed to be included in search engine results.

**1.3 Ranking**

Each search engine has different ranking factors, but they all have a few factors in common: keywords, content, and links. Keywords and content are arguably two of the most important factors that search engines look for when ranking pages. Because of this, it is imperative that you know what keywords are in highest demand within your market and incorporate those keywords into the content on your website. All your website’s content will naturally create a collection of links; search engines use link analysis algorithms that look at the sources, number, and anchor texts of links to help determine their relevance in search queries. TF-IDF is one such method for ranking.

**2. Scope of Work**

**2.1 Problem**

Improvement to the existing web search engine(FindZebra) for rare diseases.

**2.2 Functionality**

The main functionality of our prototype:  
Medi Crawl: A search engine for most of the diseases

(ref.: <https://www.findzebra.com/>)  
FindZebra: A search engine for rare diseases - <https://arxiv.org/abs/1303.3229>

**2.3 Modules**  
1. Definition (Wikipedia and WebMD) - other names, types  
2. Symptoms/ Effects  
3. Medicines / Vaccines (shopping?)  
4. Treatment (hospitals / clinics?)  
5. List of Doctors / researchers  
6. News / Research / Books

**2.4 Vision**

We are going to use Nutch - an open-source effort to build a web search engine, as our web crawler to download the data from the internet, and we are using MongoDB to store the data, Pig to perform MapReduce on the database, and we are going to build a simple web interface as front-end layer for users to search text using Node & Express JS.

**2.4.1 Strategy & Keywords**

The first step is to determine how your website is currently ranked. We will do this by searching keywords that are associated with your business and finding out how your website stacks up against your competition.

**2.4.2 Accessibility & Indexing**

For your website to be visible in search results, search engines must be able to find it. They do this by crawling the web and looking for relevant and indexable content, such as link structures and HTML features like alt tags for images.

**2.4.3 Ranking factors**

Once we determine that your website is being found by search engines and is being properly indexed, we can begin looking further to improve your ranking. The different factors that we will be looking at are content to make sure that your content is relevant and utilizes keywords without being duplicated on multiple pages, keywords, and HTML markup to make sure that your source code is relevant to crawlers.

**2.4.4 Competitive analysis**

Once we have audited your website, we will run a competitive analysis to compare your content and other SEO factors against those of your competitors. This will help us come with a plan for deliverable items that we can then present to you and implement on your website to improve your rank and drive more visitors to your site.

**2.5 Benefits**

**2.5.1 Data Management**

The data management plan to get the data for our prototype. We are planning to implement a search engine by building our own indexed database to collect webpages via Web crawling.

**2.5.2 Query Implementation**

* Query should be returned and displayed within 2s.
* Accuracy (confidence of results) should be more than 75%

**3. Resources**

OSS libraries + Free APIs - bootstrap, anaconda, beautiful soup, json, NumPy, sklearn, plotly etc.

**3.1 Server**

Express Server

**3.2 Programming tools**

Visual Code

**3.3 Platform**

MongoDB, NodeJS, MapReduce

**4. Role of each member:**

**4.1** Devharsh Trivedi ([**dtrived5@stevens.edu**](mailto:dtrived5@stevens.edu)) will be working mostly on Nutch to build WebCrawler for the engine.

* Design you own Web "spider" to go through several levels deep starting from a specified page.
* Retrieved documents should be automatically indexed (inverted file format) and stored on a server.

**4.2** Vaishnavi Gopalakrishnan ([**vgopalak@stevens.edu**](mailto:vgopalak@stevens.edu)) will be working on MongoDB and Node JS to store and process the data.

* On the client-side, a simple query interface should provide the capability to perform simple Boolean queries.
* Results displayed should be ranked.

In terms of other work, like web interface, or some unpredictable small problems to fix, we will be working together.

**5\_Schedule:**

Planning/Design: Mar 15

Deploying Nutch to download data: Apr 5

Use MongoDB to build our Database: Apr 10

Implementing and Testing MapReduce on the Database: Apr 20

Deal with the web interface & Ranking: May 4

Performance and Accuracy improvements: May 8

Documentation, Presentation & Project Report: May 12