|  |  |  |
| --- | --- | --- |
| **Search Engine for Commerce** | April 27 2016  **By**  **Stephen Balhoff (Crawling)**  **Ramprasadh Srivathsa (Relevance Model)**  **Subhasis Dutta [sxd150830] (User Interface)**  **Wyatt Lee Chastain (Clustering)**  **Matthew Bachelder (Query Expansion)** | |
| **Project Report was completed as a part of Course Work in**  **CS 6322 – Information Retrieval** | |  |

Contents

[Introduction 5](#_Toc449453955)

[Focus of the Search Engine 5](#_Toc449453956)

[Architecture of the Search Engine 5](#_Toc449453957)

[Responsibilities 5](#_Toc449453958)

[Learnings & Experince 6](#_Toc449453959)

[Crawler 7](#_Toc449453960)

[How many webpages gathered 7](#_Toc449453961)

[How the webpages were gathered 7](#_Toc449453962)

[how passed the collection to index creation 7](#_Toc449453963)

[describe clearly how many webpages were crawled in the search engine 7](#_Toc449453964)

[details of the webpages that were crawled 7](#_Toc449453965)

[how were duplicates handled 7](#_Toc449453966)

[how was hyper link information provided to the students that generated the index and the relevance model 7](#_Toc449453967)

[Indexing and Relevance 8](#_Toc449453968)

[How you assembled the index 8](#_Toc449453969)

[include a picture of how you assembled the index 8](#_Toc449453970)

[describe the web graph and how it was constructed 8](#_Toc449453971)

[show how information from the web graph was connected to the graph 8](#_Toc449453972)

[describe in detail two relevance models that you created and provide the weighting schemes that you have used 8](#_Toc449453973)

[give an example topic based page ranks computed 8](#_Toc449453974)

[discuss the hits score and show which webpages have obtained the largest score 8](#_Toc449453975)

[how interaction with user interface in generating queries to test the relevance models and to display the results of your search engine 8](#_Toc449453976)

[State clearly how many queries you have used 8](#_Toc449453977)

[how you have generated them 9](#_Toc449453978)

[how you have judged the results of your relevance models 9](#_Toc449453979)

[Collaboration with clustering to improve relevance models 9](#_Toc449453980)

[User Interface 9](#_Toc449453981)

[Design of user interface 9](#_Toc449453982)

[how you have worked with the student that has generated the index – how you have accessed the relevance models to provide the results in you user interface 9](#_Toc449453983)

[number of queries you have used for testing the search engine. 9](#_Toc449453984)

[How many were used in collaboration with the student that built the relevance models and how many did you generate on your own 9](#_Toc449453985)

[collaborate with the student that produced clusters 9](#_Toc449453986)

[how use the clustering information for relevance and presentation on the interface 9](#_Toc449453987)

[How do you think you search engine compares to Google and Bing 9](#_Toc449453988)

[Explain your judgments 9](#_Toc449453989)

[how did you use the results of clustering in presenting the results of your search engine in the user interface 10](#_Toc449453990)

[how you have decided to select the queries for the demonstration of your search engine 10](#_Toc449453991)

[Provide three examples of the queries and the results produced by your search engine, as well as the results of Google and Bing 10](#_Toc449453992)

[Clustering 10](#_Toc449453993)

[how you have designed the flat clustering 10](#_Toc449453994)

[how many predefined clusters did you select 10](#_Toc449453995)

[What did you do with the results of clustering 10](#_Toc449453996)

[did you incorporate them in the relevance models 10](#_Toc449453997)

[did you provided to the user interface results that were obtained when clustering is used 10](#_Toc449453998)

[how did you use the results of agglomerative clustering 10](#_Toc449453999)

[How many clusters did you obtain 10](#_Toc449454000)

[How were they presented on the user interface? 11](#_Toc449454001)

[How many queries did you experiment with 11](#_Toc449454002)

[State clearly how many queries you have used to test the impact of the results of each clustering method, how you have generated them and how you have judged the results of your relevance models 11](#_Toc449454003)

[Discuss how you have decided to select the queries for the demonstration of your search engine 11](#_Toc449454004)

[Provide three examples of the queries and the results produced by your search engine and the clusters that you have created 11](#_Toc449454005)

[Query Expansion 11](#_Toc449454006)

[Describe how you have selected 20 queries to test the Rocchio algorithm of your search engine 11](#_Toc449454007)

[Give examples of the web pages that you found relevant and those that you found irrelevant – and explain your judgments 11](#_Toc449454008)

[Show also the modified queries that resulted by applying Rocchio to your original queries 11](#_Toc449454009)

[Discuss the 50 queries that you have used for pseudo-relevance feedback 11](#_Toc449454010)

[For each of the three methods, i.e. associative clustering, metric clustering and scalar clustering show 11](#_Toc449454011)

[examples of 3 queries, the local document set and 12](#_Toc449454012)

[the local vocabulary and set of local stems as well as their vocabularies; 12](#_Toc449454013)

[show the values of the correlations you computed for the queries, and discuss you selection of the clusters and 12](#_Toc449454014)

[show the resulting expanded queries. 12](#_Toc449454015)

[show the results of the search engine on your expanded queries and discuss them 12](#_Toc449454016)

[Elaborate on how you have collaborated with the student responsible for the user interface to expose the results of your expanded queries as well 12](#_Toc449454017)

[Discuss which queries and their expansion you selected for the demonstration of the project. 12](#_Toc449454018)

[Discussion – all team {Matthew} 12](#_Toc449454019)

[Assumptions 12](#_Toc449454020)

[Algorithms/ Data Structures 12](#_Toc449454021)

[Collaboration - How we collaborated 13](#_Toc449454022)

[Conclusion {Matthew} 13](#_Toc449454023)

# Introduction

## Focus of the Search Engine

To take advantage of this template’s design, use the Styles gallery on the Home tab. You can format your headings by using heading styles, or highlight important text using other styles, like Emphasis and Intense Quote. These styles come in formatted to look great and work together to help communicate your ideas.

Go ahead and get started.

## Architecture of the Search Engine

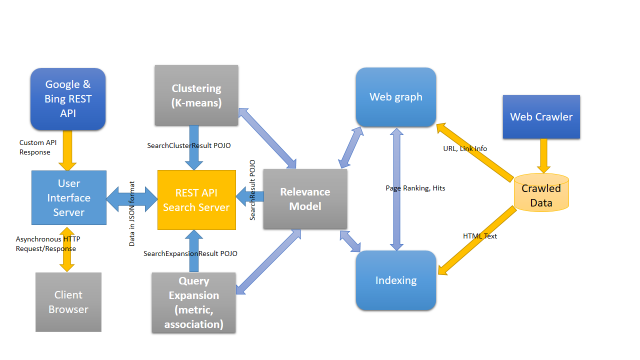


Figure 1: Architecture of the Search Engine

## Responsibilities

Based on the main modules of the search engine depicted in Figure 1, the individual responsible are:

* Web Crawler - Stephen Balhoff
* Relevance Model - Ramprasadh Srivathsa
* User Interface – Subhasis Dutta
* Clustering - Wyatt Lee Chastain
* Query Expansion - Matthew Bachelder

## Learnings & Experince

Need to write some crap….

|  |  |
| --- | --- |
| **Crawler** | |
| What you learned? |  |
| What was your experience? |  |
| What were difficulties you faced? |  |
| How did you resolve them? |  |
| **Relevance Models** | |
| What you learned? | Use of Apache Lucene for indexing and searching purposes.  Efficient use of data structures and algorithms to enhance the performance and efficiency of the search engine.  Writing code that is loosely coupled with other people’s code so that any change in my code doesn’t affect others drastically. |
| What was your experience? | The learning experience was good especially when trying to optimize the code to reduce time complexity. |
| What were difficulties you faced? | Initially, the response time recorded was a little bit high than expected and coming up with a common format to be sent to the UI. |
| How did you resolve them? | Most of the time consuming process were moved to be run during the startup of the application so that the actual searching would take less time.  In the end, a common JSON format was designed which would suit all kinds of data to be displayed on the UI. |
| User Interface | |
| What you learned? |  |
| What was your experience? |  |
| What were difficulties you faced? |  |
| How did you resolve them? |  |
| Clustering | |
| What you learned? |  |
| What was your experience? |  |
| What were difficulties you faced? |  |
| How did you resolve them? |  |
| **Query Expansion** | |
| What you learned? |  |
| What was your experience? |  |
| What were difficulties you faced? |  |
| How did you resolve them? |  |

# Crawler

This is a crawler built using <https://github.com/jbrady42/crawl> that crawls commerce pages[.](https://github.com/jbrady42/crawl.T) The crawled pages are piped to a file, and the pages to be crawled are sent to a Postgres database. As worker queues diminish, more links are pulled from the database. The pages crawled are from Walmart, Target, and Dillards.

## How many webpages gathered

There were 570,000 webpages downloaded. Some of these were duplicates and had to be thrown out. I was using URLs to detect if the URL had been crawled already, but walmart had a lot of parameters on the end of its URLs which caused some duplication. There were also pages gathered that were browsing pages instead of product pages. Most of the pages, however, are valid, unique product pages. There are 18947092 in the crawler database.

## How the webpages were gathered

The webpages were gathered using the open source project located at <https://github.com/jbrady42/crawl>. To find pages to crawl, the crawler selects from a postgres databases of pages which includes the URL, along with whether the URL has been visited and some other important metadata. The crawled html file is piped to a file, and another program extracts the links and loads them into the database if their URL does not already exist. The crawler maintains a certain number of worker queues, and a watching thread pulls more links to crawl from the database when a worker queue runs out of work. Each worker queue is specific to a single IPv4 address.

## how passed the collection to index creation

The files were given to the indexing team member on a hard drive in gzip format. They were broken up into multiple files because there was a lot of data and I did not want to store one giant file. These files contained JSON that had the URL of the page and the html page data.

## describe clearly how many webpages were crawled in the search engine

There were 570,000 pages crawled during this search. This includes pages that did not have the same URL but were duplicates, and pages that were browsing pages for certain categories of products.

## details of the webpages that were crawled

The pages crawled all fell under three domains: walmart.com, target.com, and dillards.com. The root of each domain was crawled, as were category pages to find more products. Along with these, the actual product pages were crawled.

## how were duplicates handled

Originally, I intended to use sketches to detect duplicates. I was creating the permutations as needed, so they were never fully created. As can be expected, I ran out of space after a couple hours because there is not nearly enough room to create permutations over the space of 64 bit integers. I switched to using URLs to detect duplication. If the URL was the same, I marked it as a duplicate and didn’t crawl it again. In hindsight, I should have removed the parameters from the query and used that modified URL to check for duplicates. Once the pages were downloaded, I used the title of the webpages to detect duplication.

## how was hyper link information provided to the students that generated the index and the relevance model

The gzipped files that contained the JSON were given to the indexer and the project member who did the relevance model. This JSON contained the URLs and the HTML.

# Indexing and Relevance

Indexing and relevance models have been implemented by using Apache Lucene.

## How you assembled the index

Apache Lucene was used for indexing. Lucene is an open source software for indexing and searching text documents in an efficient way. Lucene goes through each file word by word and indexes it. Lucene makes use of incremental indexing. The main advantage of using Lucene to index is that it is as fast as batch indexing. The created index will be compressed and will roughly be around 20-30% of the size of the total data.

## describe the web graph and how it was constructed

For any page, there are many pages that are leading to it and there are many pages that this page leads to. The web graph is constructed mainly by the process mentioned above. By checking all the possible pages that are connected to the page in a way, the web graph can be easily determined. For instance, assume there are pages p1, p2 and p3. Page p1 is the parent page and leads to p2 and p3. Page p3 leads back to p1 while page p2 is a dead end. It can be seen from the above instance that, p1 leads to p2, p1 leads to p3, and p3 leads to p1. In a similar way, the web graph can be calculated.

## show how information from the web graph was connected to the graph

From the web graph constructed, it is pretty easy to get the information that is required. It can be seen, from the web graph, the number of outlinks and inlinks of any page. The graph can be mapped by mapping the outlinks and inlinks to the corresponding pages.

## describe in detail two relevance models that you created and provide the weighting schemes that you have used

The relevance models used were vector space model and Boolean model. Lucene is tightly bound with both the models and thus making the implementation of either of the models very easy. The use of the relevance models to determine the relevancy of the documents is very highly dependent on how the documents are indexed. Without proper indexing, the result of the relevance models might not be as relevant as it is supposed to be. Lucene does the scoring by default. Lucene uses a combination of vector space model and Boolean model to determine the relevancy of the documents to the query. The vector space model is based on the comparison of how many times a term appears in a document to the total number of times the term appears in the entire documents in the collection. The Boolean model is used to narrow down the documents that need to be scored by using Boolean logic on the query. Though, Lucene uses both the models to score the documents, it can be seen that vector space model is more dominant than the Boolean model.

## give an example topic based page ranks computed

Apache Lucene allows the developer to customize its application easily. Lucene allows each document to be boosted by setting a score to each of them. For a query, the matching documents are obtained and based on the boosting scores that were set to them during indexing, one can determine the documents with the highest page ranking for that query. To do this, initially during indexing, a score is set to each document based on the page ranks computed by analyzing the web graph that was created. For example, for the query “lotion”, out of the many results that is obtained, we can see that the URL <http://www.walmart.com/ip/Johnson-s-Baby-Wash-Honey-Apple-28-fl-oz./14675930?action=product_interest&action_type=title&beacon_version=1.0.1&bucket_id=irsbucketdefault&category=&client_guid=9ef4ced7-044b-4261-8938-c6f7664259ed&config_id=2&customer_id_enc=&findingMethod=p13n&guid=4e5d9e4c-a2fb-4b76-b88a-5f668906d348&item_id=14675930&parent_anchor_item_id=10294048&parent_item_id=10294048&placement_id=irs-2-m3&strategy=PWBAB&visitor_id>=

gets the highest page ranking and thus it is displayed at the first because of its high priority.

## discuss the hits score and show which webpages have obtained the largest score

The relevant documents that are matching a query are obtained and the HITS score is calculated by analyzing the document that are linking from it and some of the document that linking to it. Since our search engine is based on commerce platform, there were a large number of hubs, however we have classified only the pages with the very highest HITS scores as hubs and considered other pages as any other web page. Some of the web pages with high HITS scores are given below:

<http://www.target.com/c/garage-storage-organization-home/-/N-5xtny#?lnk=bnav_t_spc_4_0>

<http://www.target.com/c/power-tools-tool-sets-home-improvement/power-screwdriver/-/N-5xtsfZ5dc9x>

<http://www.walmart.com/browse/baby/baby-skin-care-grooming/5427_486190_1042639>

## interaction with user interface in generating queries to test the relevance models and to display the results of your search engine

The query entered by the user is sent from the UI to the indexing part, which in turn goes through the index, searches for the term, incorporates the relevance models and gets the most relevant documents. The relevant document information are stored in a class and sent it to the user interface as a JSON object so that it is easier to display on the screen making it very user friendly. The user interface and the indexing part are loosely coupled so that even if one fails, the other keeps running. Put shortly, the UI sends a string query to the backend and in turn receives a JSON object containing the most relevant documents from the backend.

### State clearly how many queries you have used

The search engine has been tested for about 50 different queries like iPhone, laptop below $1000, android tabs, lotion, etc.

### how you have generated them

The queries have been generated in random in order to check the correctness of the search engine. There is no clear process in generating the queries as similar queries have a higher possibility to return similar results. It is easier to find errors by entering random queries than queries which are similar. For each query entered, a few similar queries were generated to check the correctness of the query and some random queries to check the correctness of the search engine.

### how you have judged the results of your relevance models

There were several ways used by us to determine the correctness of the relevance models. For every query, there are a lot URLs that are displayed in the UI. During initial part of the testing, we went through many of the URLs to see if the query terms were present in the web page and if they were appearing predominantly in those web pages. We also compared some of our results with Google and Bing results. Although, they didn’t match exactly, there were many results that were common between our results and those of Bing and Google.

## Collaboration with clustering to improve relevance models

Once the relevance documents have been determined for the query specified, the clustering information needs to be integrated with the relevance documents. The relevant documents are sent to the clustering part of the application. The clusters are determined before hand during the startup of the application and hence once the relevant documents are passed to the clustering information is integrated with them and sent back to be displayed.

# User Interface

Dg dfgfdg

## Design of user interface

Dfg fd g

## how you have worked with the student that has generated the index – how you have accessed the relevance models to provide the results in you user interface

Ssf sfs f

## number of queries you have used for testing the search engine.

S fsfsd f

## How many were used in collaboration with the student that built the relevance models and how many did you generate on your own

D gfdgfd g

## collaborate with the student that produced clusters

Gdfgfd gfd

## how use the clustering information for relevance and presentation on the interface

D gfd gfd gd

## How do you think you search engine compares to Google and Bing

Dg fdgfdg

## Explain your judgments

F gfd g

## how did you use the results of clustering in presenting the results of your search engine in the user interface

D gfdgdf gfd

## how you have decided to select the queries for the demonstration of your search engine

Dg fdg fdjghdfj g

## Provide three examples of the queries and the results produced by your search engine, as well as the results of Google and Bing

Hdfjghdfjghj dg

# Clustering

Fkd fjkd

## how you have designed the flat clustering

### how many predefined clusters did you select

## What did you do with the results of clustering

## did you incorporate them in the relevance models

## did you provided to the user interface results that were obtained when clustering is used

## how did you use the results of agglomerative clustering

## How many clusters did you obtain

## How were they presented on the user interface?

## How many queries did you experiment with

## State clearly how many queries you have used to test the impact of the results of each clustering method, how you have generated them and how you have judged the results of your relevance models

## Discuss how you have decided to select the queries for the demonstration of your search engine

## Provide three examples of the queries and the results produced by your search engine and the clusters that you have created

# Query Expansion

## Describe how you have selected 20 queries to test the Rocchio algorithm of your search engine

i. List them in your reports.

## Give examples of the web pages that you found relevant and those that you found irrelevant – and explain your judgments

## Show also the modified queries that resulted by applying Rocchio to your original queries

## Discuss the 50 queries that you have used for pseudo-relevance feedback

## For each of the three methods, i.e. associative clustering, metric clustering and scalar clustering show

### examples of 3 queries, the local document set and

### the local vocabulary and set of local stems as well as their vocabularies;

### show the values of the correlations you computed for the queries, and discuss you selection of the clusters and

### show the resulting expanded queries.

### show the results of the search engine on your expanded queries and discuss them

### Elaborate on how you have collaborated with the student responsible for the user interface to expose the results of your expanded queries as well

### Discuss which queries and their expansion you selected for the demonstration of the project.

# Discussion – all team {Matthew}

## Assumptions

## Algorithms/ Data Structures

## Collaboration - How we collaborated

# Conclusion {Matthew}

In conclusion…