|  |  |  |
| --- | --- | --- |
| **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](#_Toc449472459)

[Focus of the Search Engine 5](#_Toc449472460)

[Architecture of the Search Engine 5](#_Toc449472461)

[Open Source Libraries Used in the Project 6](#_Toc449472462)

[Responsibilities 6](#_Toc449472463)

[Learnings & Experince 7](#_Toc449472464)

[Crawler 9](#_Toc449472465)

[webpages gathered 9](#_Toc449472466)

[How the webpages were gathered 9](#_Toc449472467)

[how passing collection to index creation 9](#_Toc449472468)

[describe clearly how many webpages were crawled in the search engine 9](#_Toc449472469)

[details of the webpages that were crawled 9](#_Toc449472470)

[how were duplicates handled 9](#_Toc449472471)

[how was hyper link information provided to the students that generated the index and the relevance model 10](#_Toc449472472)

[Indexing and Relevance 10](#_Toc449472473)

[How you assembled the index 10](#_Toc449472474)

[include a picture of how you assembled the index 10](#_Toc449472475)

[describe the web graph and how it was constructed 10](#_Toc449472476)

[show how information from the web graph was connected to the graph 10](#_Toc449472477)

[describe in detail two relevance models that you created and provide the weighting schemes that you have used 10](#_Toc449472478)

[give an example topic based page ranks computed 10](#_Toc449472479)

[discuss the hits score and show which webpages have obtained the largest score 10](#_Toc449472480)

[how interaction with user interface in generating queries to test the relevance models and to display the results of your search engine 11](#_Toc449472481)

[State clearly how many queries you have used 11](#_Toc449472482)

[how you have generated them 11](#_Toc449472483)

[how you have judged the results of your relevance models 11](#_Toc449472484)

[Collaboration with clustering to improve relevance models 11](#_Toc449472485)

[User Interface 12](#_Toc449472486)

[Design of user interface 12](#_Toc449472487)

[Collaboration with relevance models to provide the results in uI 12](#_Toc449472488)

[number of queries you have used for testing the search engine. 12](#_Toc449472489)

[collaboration to display Query Expansion Result 12](#_Toc449472490)

[collaborate with the student that produced clusters 13](#_Toc449472491)

[clustering information for relevance and presentation 13](#_Toc449472492)

[How do you think you search engine compares to Google and Bing 13](#_Toc449472493)

[results of clustering in user interface 13](#_Toc449472494)

[selectION the queries for the demonstration 13](#_Toc449472495)

[Provide three examples of the queries and the results produced by your search engine, as well as the results of Google and Bing 13](#_Toc449472496)

[Clustering 13](#_Toc449472497)

[how you have designed the flat clustering 13](#_Toc449472498)

[how many predefined clusters did you select 13](#_Toc449472499)

[What did you do with the results of clustering 13](#_Toc449472500)

[did you incorporate them in the relevance models 14](#_Toc449472501)

[did you provided to the user interface results that were obtained when clustering is used 14](#_Toc449472502)

[how did you use the results of agglomerative clustering 14](#_Toc449472503)

[How many clusters did you obtain 14](#_Toc449472504)

[How were they presented on the user interface? 14](#_Toc449472505)

[How many queries did you experiment with 14](#_Toc449472506)

[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 14](#_Toc449472507)

[Discuss how you have decided to select the queries for the demonstration of your search engine 14](#_Toc449472508)

[Provide three examples of the queries and the results produced by your search engine and the clusters that you have created 14](#_Toc449472509)

[Query Expansion 14](#_Toc449472510)

[Describe how you have selected 20 queries to test the Rocchio algorithm of your search engine 14](#_Toc449472511)

[Give examples of the web pages that you found relevant and those that you found irrelevant – and explain your judgments 15](#_Toc449472512)

[Show also the modified queries that resulted by applying Rocchio to your original queries 15](#_Toc449472513)

[Discuss the 50 queries that you have used for pseudo-relevance feedback 15](#_Toc449472514)

[For each of the three methods, i.e. associative clustering, metric clustering and scalar clustering show 15](#_Toc449472515)

[examples of 3 queries, the local document set and 15](#_Toc449472516)

[the local vocabulary and set of local stems as well as their vocabularies; 15](#_Toc449472517)

[show the values of the correlations you computed for the queries, and discuss you selection of the clusters and 15](#_Toc449472518)

[show the resulting expanded queries. 15](#_Toc449472519)

[show the results of the search engine on your expanded queries and discuss them 15](#_Toc449472520)

[Elaborate on how you have collaborated with the student responsible for the user interface to expose the results of your expanded queries as well 15](#_Toc449472521)

[Discuss which queries and their expansion you selected for the demonstration of the project. 16](#_Toc449472522)

[Discussion – all team {Matthew} 16](#_Toc449472523)

[Assumptions 16](#_Toc449472524)

[Algorithms/ Data Structures 16](#_Toc449472525)

[Collaboration - How we collaborated 16](#_Toc449472526)

[Conclusion {Matthew} 16](#_Toc449472527)

# Introduction

## Focus of the Search Engine

In this project we implemented a fully working commerce search engine that has crawling, page ranking, indexing and an easy to use simple user interface to retrieve the indexed document.

Our design motivation is to build modular system, where each module will work independently. We tried to design such that failure of one single module won’t bring down the whole system.

## Architecture of the Search Engine

The project consists of following modules –

a) Web Crawler: Collects documents from mentioned websites and organizes the data.

b) Indexer: Creates an index of incoming and outgoing links. This creates the graph structure.

c) Search API Server: Allows user to search for a particular query asynchronously and get the result in JSON.

d) User Interface: Allows user to search for a term and visualize the resultant clustering.

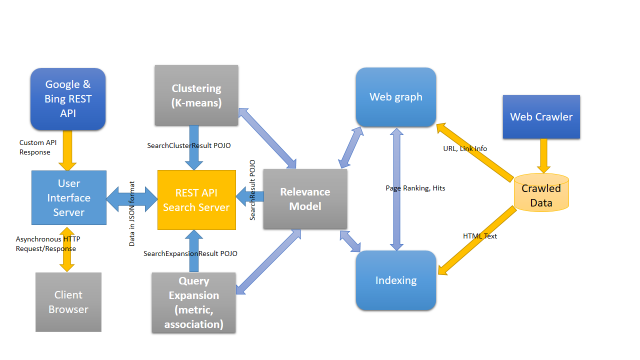
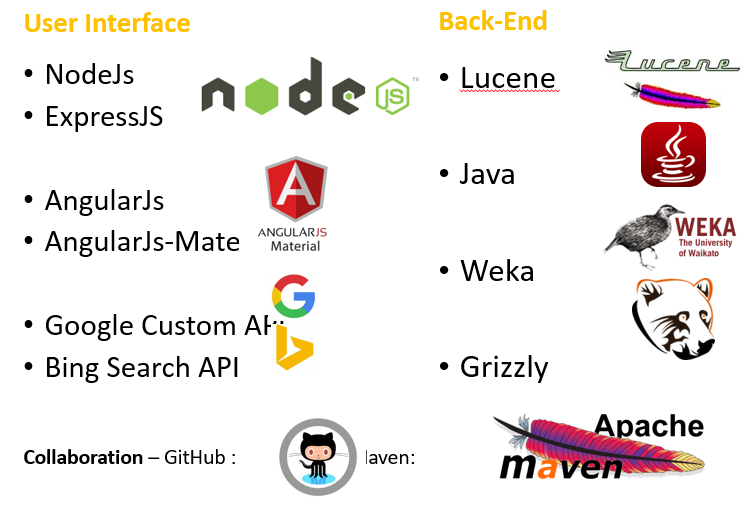


Figure 1: Architecture of the Search Engine

## Open Source Libraries Used in the Project



## 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

|  |  |
| --- | --- |
| **Crawler** | |
| What you learned? |  |
| What was your experience? |  |
| What were difficulties you faced? |  |
| How did you resolve them? |  |
| **Relevance Models** | |
| What you learned? |  |
| What was your experience? |  |
| What were difficulties you faced? |  |
| How did you resolve them? |  |
| **User Interface** | |
| What you learned? | Using commercially available Search API of Google and Bing.  Build Single Page Applications using Material design.  Building asynchronous Web services with a Node server running on event loop.  Displaying search results in a responsive design that can be displayed on any device resolution. |
| What was your experience? | It was a good learning experience in developing the user interface and |
| What were difficulties you faced? | Initially there was difficulty in aggregating on a common data format. |
| How did you resolve them? | It was resolved by moving all interaction points to the back –end java objects which are very standard and flexible and can easily be converted to any document format like JSON or XML. |
| **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.

## 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 passing 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

## How you assembled the index

Sdsds ds

### include a picture of how you assembled the index

sdd

## describe the web graph and how it was constructed

Adasd sad

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

Sd sds s

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

Sd sd sds

## give an example topic based page ranks computed

Sdsds d

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

S ds ds ds d

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

S fdf sd f

### State clearly how many queries you have used

Dgfdg

### how you have generated them

Sdsd

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

Dgdfgd

## Collaboration with clustering to improve relevance models

Gdfgkdfj gk fdjgkkdg

# User Interface

In addition to the system retrieval capacity, we’re interested in the visualization of the search results and displaying them in a readable manner. To implement this we have created a search based user interface, where the user can search for a term and the results are displayed.

## Design of user interface

For implementation we have chosen NODEJS stack, to implement the project as a web application. We have used a basic MVC architecture with HTML pages as our view and AngularJS for all client side scripting. The controller is a Java HTTP Server built with JAVA Jersey. To render the cluster we have grouped them and displayed them in separate cards.

When a user enters a term in the search box and clicks search, the request is sent to the REST API controller, using an Ajax call. The controller delegates the request to a method in the DAO, which invokes a service to search the index files. The Index searcher returns the hierarchical structure of the cluster. This object is saved in the session bean and the top-level clusters are converted in the view object, with the help of the layout engine. The layout engine tries to draw the graph in an aesthetically pleasing way.

## Collaboration with relevance models to provide the results in uI

To make integration of the user interface with the relevance models we have used Java Models which are filled by the relevance models then this are converted to JSON format and obtained as response.

To collaborate with the relevance model we encapsulate the result in a **Search Result Java object**, which is converted and received as a response. The JSON file is converted to a JavaScript object which is monitored by the angular event and rendered.

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

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

## collaboration to display Query Expansion Result

To make integration of the user interface with the relevance models we have used Java Models which are filled by the relevance models then this are converted to JSON format and obtained as response.

To collaborate with the relevance model we encapsulate the result in **a Search Expansion Result** Java object, which is converted and received as a response. The JSON file is converted to a JavaScript object which is monitored by the angular event and rendered.

## collaborate with the student that produced clusters

To make integration of the user interface with the relevance models we have used Java Models which are filled by the relevance models then this are converted to JSON format and obtained as response.

To collaborate with the relevance model we encapsulate the result in **a Search Cluster Result** Java object, which is converted and received as a response. The JSON file is converted to a JavaScript object which is monitored by the angular event and rendered.

## clustering information for relevance and presentation

Currently the cluster information are presented in separate cards to display the groups detected by the K-means cluster. Then the relevant results are displayed as mentioned by the clustering.

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

In my estimate our search results compares very nicely with the custom Google results that shows results from only the websites we have crawled with a relevance of around 40%. There was low match with that of Bing results however that was mostly due to the method bing results were obtained which had results from all websites that reduced the common webpages.

## results of clustering in user interface

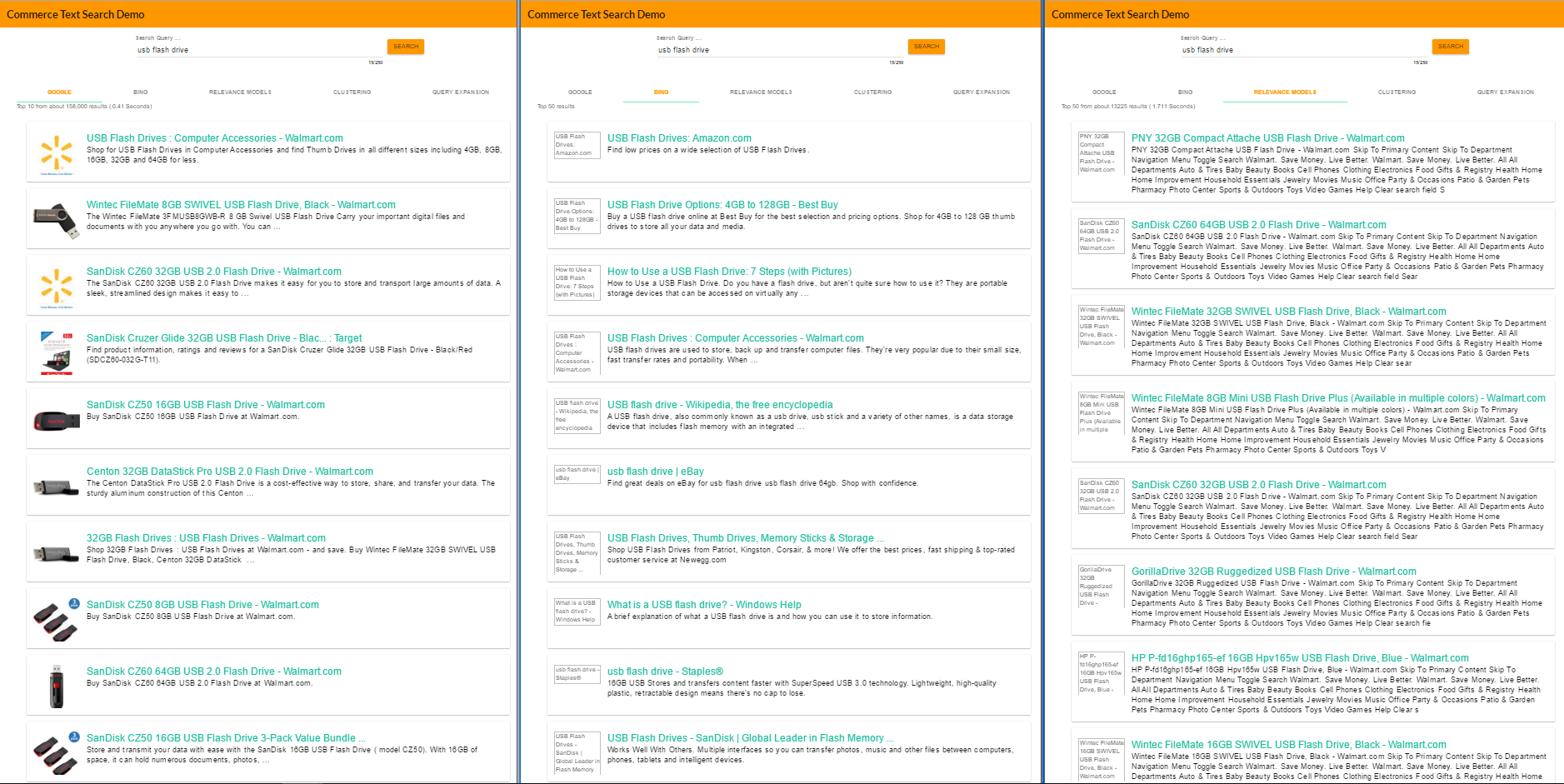
The results of clustering are displayed as normal search results as displayed for relevance models. Along with this we are displaying the cluster groups.

## selectION the queries for the demonstration

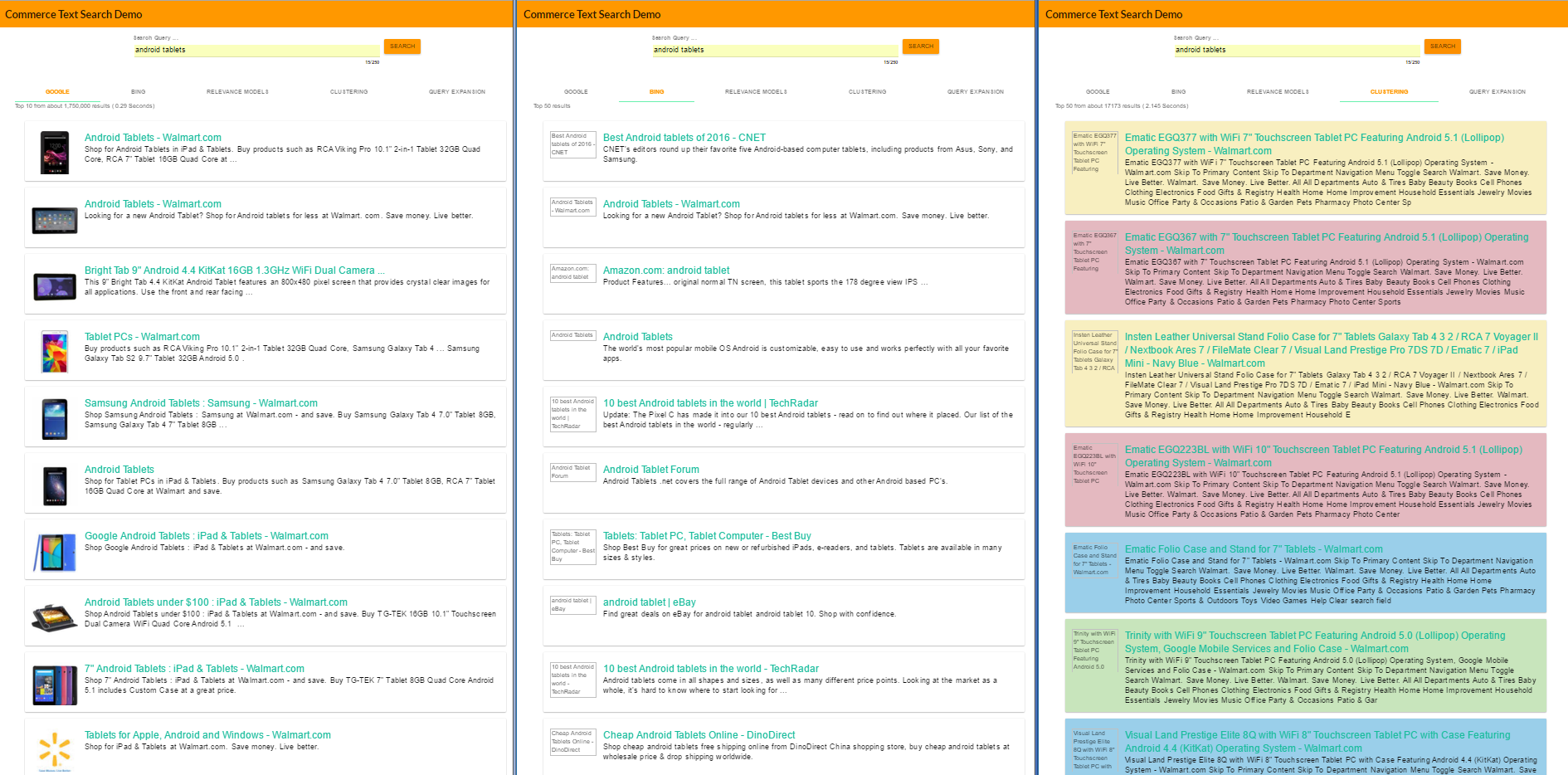
The selection of queries was done empirically based on the pages that were displayed as a part of the results. They were selected by choosing randomly the products that we thought were part of the product portfolio of the company.

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

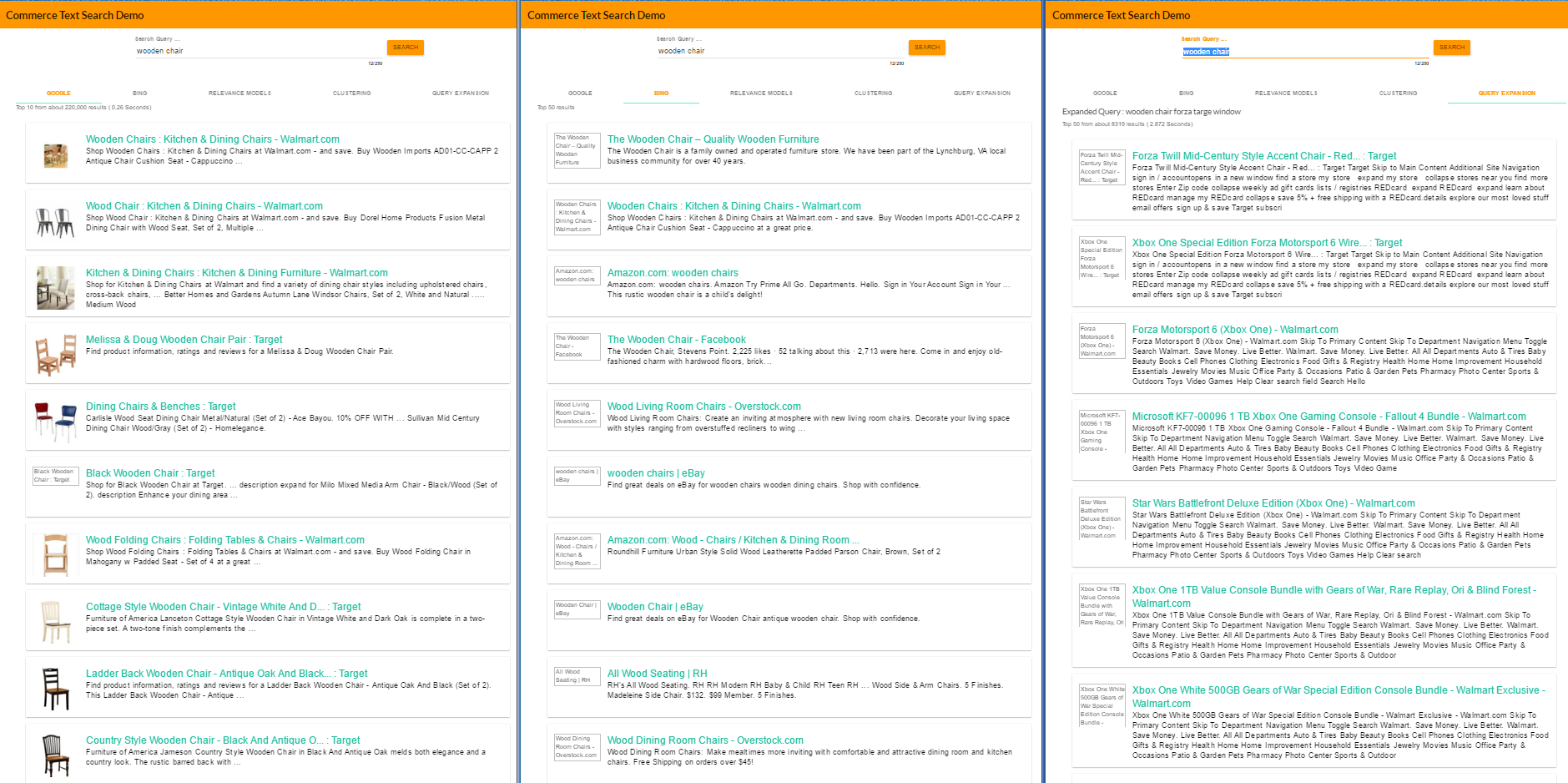
### Query1 – usb flash drive



### Query 2 – Android Tablet



### Query 3 – Wooden chairs



# 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…