The University of Texas at DallasCS 6322

Information Retrieval

Spring 2022

Class Project Report

Project TITLE: Search Engine for Soccer

Group: No. 12

Students: Kanamata Reddy, Vishnu Vardhan Reddy,vxk210042@utdallas.edu

Kotra, Sai Charan, sxk210083@utdallas.edu

Manthri, Satya Sai Bharadwaj, sxm210073@utdallas.edu

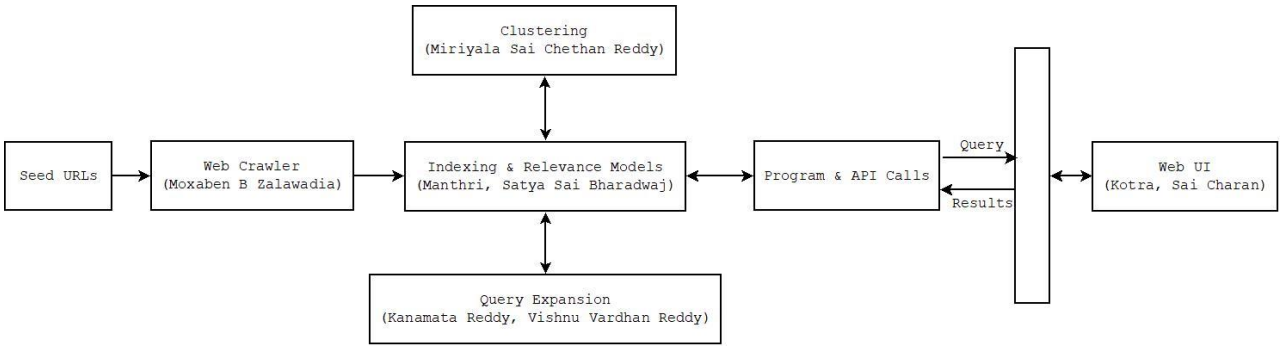
Miriyala Sai Chethan Reddy, sxm200225@utdallas.edu

Zalawadia Moxaben Bhupatbhai, mxz210014@utdallas.edu

1. Introduction

In this project we are going to make a search engine for soccer. This search engine is all about finding everything about soccer. We are going to use all concepts which we have learnt in the course Information retrieval. We are going to use:

- Clustering
- Web Crawling
- Query Expansion
- Relevance Models and other methods



Responsibilities:

Crawling: Zalawadia Moxaben Bhupatbhai (mxz210014)

Indexing and Relevance: Manthri, Satya Sai Bharadwaj (sxm210073)

User Interface and Comparisons with Google and Bing: Kotra, Sai Charan (sxk210083)

Clustering: Miriyala Sai Chethan Reddy (sxm200225)

Query Expansion and Relevance Feedback: Kanamata Reddy, Vishnu Vardhan Reddy (vxk210042)

Learning:

Through this project we all learnt following:

- Practical implementation of Concepts
- Team Work
- Creating a perfect Search Engine
- Proper Planning
- Time Management

Difficulties Faced:

We faced many difficulties while working on this project.

Problem	Solution
Initially, We didn’t understand what to do and which tools to select	We all sat together and we did some research to get clarity over everything
We were unable to run some large code due to our system memory and RAM Limitations	We tried to run in different large server which ran our programs
Working with API was tough	But we figured out throught tutorials and documentations

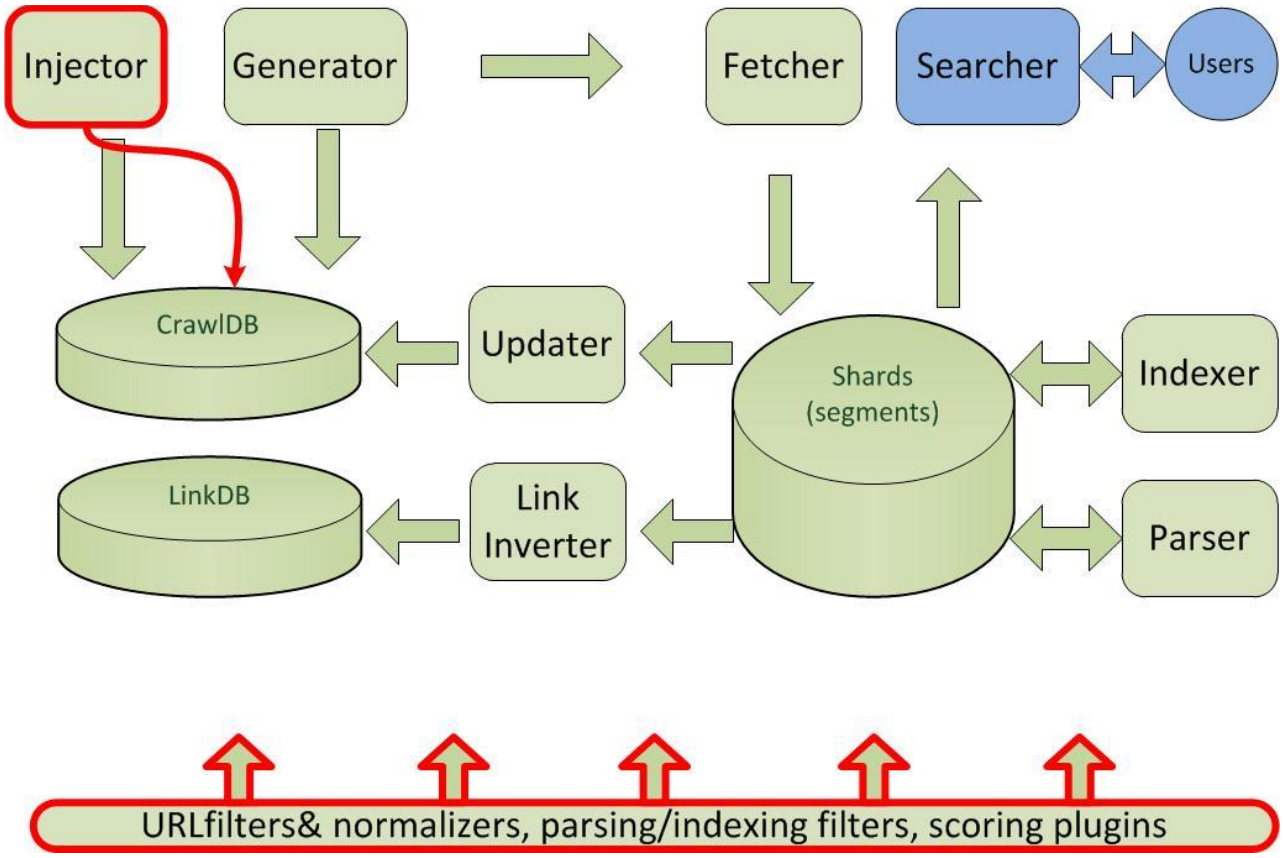
This project has challenged us a lot, and finally, we completed everything properly.

2. Crawling

TECHNOLOGY STACK

1. Apache Nutch-1.14
2. Solr-6.6.5
3. Java 8.0

NUTCH ARCHITECTURE



Details about how pages were gathered and how the collection was passed to the index creation.

URL seed list

- A URL seed list includes a list of websites, one-per-line, which nutch will look to crawl.

- Total of 67 seed links were crawled.
  Crawled seed list:

1. https://www.espn.com/soccer/
2. https://www.soccer.com/
3. https://www.espn.com/soccer/competitions
4. https://en.wikipedia.org/wiki/Association_football
5. https://www.ussoccer.com/
6. https://www.mlssoccer.com/
7. https://www.nytimes.com/section/sports/soccer
8. https://www.cbssports.com/soccer/
9. https://fifa.com
10. https://www.uslsoccer.com/
11. https://www.nbcsports.com/soccer
12. https://www.usatoday.com/sports/soccer/
13. https://twitter.com/hashtag/soccer
14. https://www.washingtonpost.com/sports/soccer/
15. https://sports.yahoo.com/soccer/
16. https://football.fandom.com/wiki/Football_Wiki
17. https://www.thesoccerworldcups.com/
18. https://www.imdb.com/list/ls021591480/
19. https://www.ussoccer.com/teams/
20. https://en.wikipedia.org/wiki/Soccer_in_the_United_States
21. https://en.wikipedia.org/wiki/Major_League_Soccer
22. https://www.uefa.com/nationalassociations/uefarankings/club/
23. https://www.globalfootballrankings.com/
24. https://www.realmadrid.com/en
25. https://www.manutd.com/
26. https://www.soccer24.com/
27. https://www.footballaustralia.com.au/
28. https://www.soccerstand.com/soccer/
29. http://www.australiafootball.com/
30. https://www.sportingnews.com/us/soccer/news
31. https://www.nbcolympics.com/soccer
32. https://olympics.com/en/olympic-games/tokyo-2020/results/football
33. https://www.sbnation.com/olympic-soccer
34. https://canadasoccer.com/
35. https://www.the-afc.com/en/national/afc_asian_cup.html
36. https://olympics.com/en/olympic-games
37. https://www.eurosport.com/football/olympic-games/
38. https://www.skysports.com/football
39. https://www.fifamuseum.com/en/
40. https://www.the-aiff.com/
41. https://www.conmebol.com/
42. https://www.oceaniafootball.com/
43. https://www.uefa.com/
44. https://www.cafonline.com/
45. https://www.concacaf.com/
46. https://www.teamusa.org/
47. https://www.britannica.com/browse/Soccer
48. https://www.reddit.com/r/soccer/
49. https://www.theguardian.com/football
50. https://7news.com.au/sport/soccer
51. https://www.ftbl.com.au/news
52. https://en.wikipedia.org/wiki/United_States_Soccer_Federation

53. https://paralympic.ca/paralympic-sports/football-5-side
54. https://www.usaba.org/sports/blind-soccer/
55. https://theathletic.com/football/
56. https://www.si.com/soccer
57. https://www.stingsoccer.com/
58. https://usclubsoccer.org/
59. https://www.underarmour.com/en-us/c/mens/sports/soccer/
60. https://us.puma.com/us/en/men/shoes/soccer
61. https://www.nike.com/soccer
62. https://worldsoccertalk.com/2022/04/14/apple-tv-plus-mls-match-made-in-soccer-heaven/
63. https://www.latimes.com/sports/soccer
64. https://apnews.com/hub/soccer
65. https://www.shmoop.com/ncaa/soccer/famous-athletes.html
66. https://www.statista.com/statistics/266636/best-paid-soccer-players-in-the-2009-2010-season/
67. https://www.wsj.com/articles/katie-meyer-stanford-womens-soccer-player-found-dead-in-campus-residence-11646315173

Nutch data is composed of:

1. The information about every URL known to Nutch, including whether it was fetched, and, if so, when is stored in a database named crawldb.
2. The list of known links to each URL, including both the source URL and anchor text of the link are stored in the link database named linkdb.
3. Each segment in set of segments is a set of URLs that are fetched as a unit. Segments are directories with the following subdirectories:

- a *crawl_generate* names a set of URLs to be fetched.
- a *crawl_fetch* contains the status of fetching each URL.
- a *content* contains the raw content retrieved from each URL.
- a *parse_text* contains the parsed text of each URL.
- a *parse_data* contains outlinks and metadata parsed from each URL.
- a *crawl_parse* contains the outlink URLs, used to update the crawldb

Seeding the crawldb with a list of URLs:

Bootstrapping from an initial seed list.

This option shadows the creation of the seed list.
bin/nutch inject crawl/crawldb urls

Fetching:

For fetching, first generate a fetch list from the database:
bin/nutch generate crawl/crawldb crawl/segments

A fetch list is generated for all of the pages that are need to be fetched. The fetch list is saved in a newly created segment directory. The segment directory is named by the time it's created. After that the name of this segment was saved in the shell variable s1:
s1=`ls -d crawl/segments/2* | tail -1`
echo $s1

Run the fetcher on this segment with:
bin/nutch fetch $s1

Then parse the entries:

bin/nutch parse $s1

When this is complete, the database is updated with the results of the fetch:
bin/nutch updatedb crawl/crawldb $s1

Updated entries for all initial pages and new entries that correspond to newly discovered pages linked from the initial set are stored in the database.

After that generate and fetch a new segment containing the top-scoring 1,000 pages:
bin/nutch generate crawl/crawldb crawl/segments -topN 1000
s2=`ls -d crawl/segments/2* | tail -1`
echo $s2

bin/nutch fetch $s2
bin/nutch parse $s2
bin/nutch updatedb crawl/crawldb $s2

Fetch one more round:
bin/nutch generate crawl/crawldb crawl/segments -topN 1000
s3=`ls -d crawl/segments/2* | tail -1`
echo $s3

bin/nutch fetch $s3
bin/nutch parse $s3
bin/nutch updatedb crawl/crawldb $s3

By this point few thousand pages have been fetched.

APACHE NUTCH CONFIGURATION

Modifications that were made to the nutch configuration file
Path: apache-nutch-1.1.15/conf/nutch-site.xml
Ignore outlinks to the same hostname: False
Ignore outlinks to the same domain: False
Limit to only a single outlink to the same page: False

We crawled total of 190,564 pages and after duplication deletion 105,463 useful pages were gathered using Nutch. We were aiming for more but had to cut it short due to running out of time. We ensured we did not have duplication in our crawl by using Nutch's built in remove duplicates feature.

To pass on the hyperlink information, I gave read permissions to all of the data I was able to collect and explained the output of Nutch to my teammates so that they would be able to find what they needed from me and copy all of the data I collected. Before indexing first invert all of the links, so that my teammates can index incoming anchor text with the pages.
bin/nutch invertlinks crawl/linkdb -dir crawl/segments

Indexing And Relevance

Indexing and Relevance are done by Satya Sai Bharadwaj Manthri.

Indexing:

Indexing is an important process in Information Retrieval (IR) systems. The practice of characterizing and identifying materials in terms of their topic matter is known as indexing. The ideas are retrieved from texts using the analysis method and then transcribed into indexing system elements like thesauri, categorization schemes, and so on.

Indexing is done using Solr. Solr is a Java-based open-source corporate search tool. Full-text search hit highlighting, faceted search, real-time indexing, dynamic clustering, database integration, NoSQL capabilities, and rich document handling are some of its key features.

I have taken the data of crawled web pages from **Zalawadia Moxaben Bhupatbhai** for indexing.

Using following:

bin/nutch solrindex http://localhost:8983/solr/soccer /home/reddy/SoccerCrawlFinal/crawldb/ -linkdb /home/reddy/SoccerCrawlFinal/linkdb/ -dir /home/reddy/SoccerCrawlFinal/segments/ -filter -normalize -deleteGone

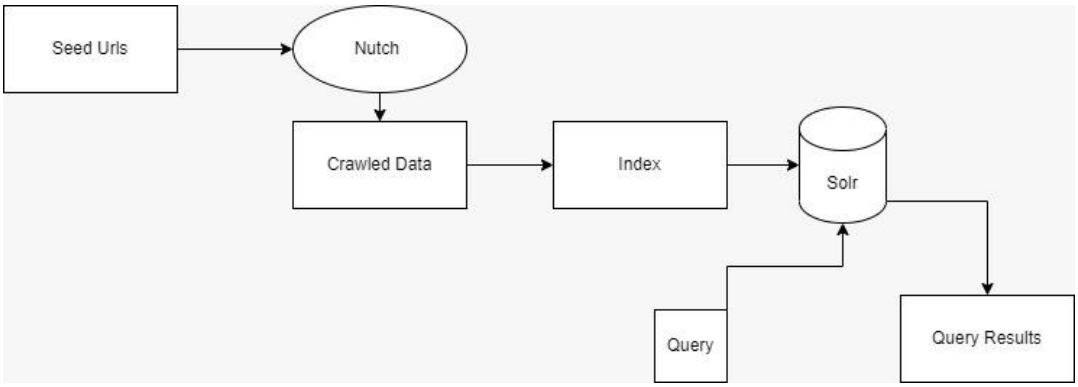
curl http:/localhost:8983/solr/soccer/select?q=*:*&wt=json&indent=true&rows=1000000000" > soccer_index.json

We make outputs into the following. They are:

1) Crawldb - contains all the links parsed by the Nutch
2) Linkdb - contains outgoing and the incoming URLs for each webpage
3) Segments - contains the list of URLs to be crawled or being crawled for incremental crawl
4) Soccer_index.json- Everything in the JSON file

All these are given as input to Solr for indexing. The following command is used to create the index of crawled pages in Solr with the help of nutch:

Apache Nutch's binary is bin/nutch. The index command collects material from one or more segments and transmits it to all IndexWriter plugins that have been activated, which then sends the documents to Solr.



Index creation is done in the above sequence. Seed URLs are crawled after that crawled data is indexed with solr. With the indexed data in solr, we use different queries to get query results.

Web Graph Creation And Statistics:

The webgraph is a representation of the World Wide Web's directed links between pages. A web graph has been created from the crawled URLs. The nutch command was used to construct a web graph of all the crawled web pages.

We create web graph with a command:

bin/nutch org.apache.nutch.scoring.webgraph.WebGraph -segmentDir /home/reddy/SoccerCrawlFinal/segments/ -webgraphdb /home/reddy/SoccerCrawlFinal/webgraphdb/

Nutch's webgraph command processes multiple segments and requires an output directory to store the finished web graph components. An inlink database, an outlink database, and a node database are all created by the webgraph. The inlink database contains a list of urls and their inlinks. The outlink database contains a list of urls as well as all of their outlinks. The node database is a collection of URLs that includes node metadata such as the number of inlinks and outlinks.

Total Number of links = 955321
Total Number of inlinks = 91116
Total Number of outlinks = 864205

Total Number of nodes = 202131
The largest number of ingoing links = 10000
The largest number of outgoing links = 85

Web graph information connected to the index. We do link analysis through the following commands.

bin/nutch solrindex http://localhost:8983/solr/vector /home/reddy/SoccerCrawlFinal/crawldb/ -linkdb /home/reddy/SoccerCrawlFinal/linkdb/ -dir /home/reddy/SoccerCrawlFinal/segments/ -filter -normalize -deleteGone

bin/nutch index -D solr.server.url=http://localhost:8983/solr/vector /home/reddy/SoccerCrawlInitial/crawldb/ -linkdb /home/reddy/SoccerCrawlInitial/linkdb/ -dir /home/reddy/SoccerCrawlInitial/segments/

bin/nutch readlinkdb /home/reddy/SoccerCrawlFinal/linkdb/ -dump inlinks

bin/solr start -Dsolr.ltr.enabled=true

With the above queries we have connect information of from graph to index. Mainly with graph information we calculated pageranks and HITS.

We do the link analysis and generate the following results:

- Inlinks
- Outlinks etc

Combing the Relevance models and Link analysis:

We have used weighted sum to combine relevance models and link analysis. We did weighted sum of vector relevance and page ranks with weights of 0.6 for page rank and 0.4 for vector relevance.

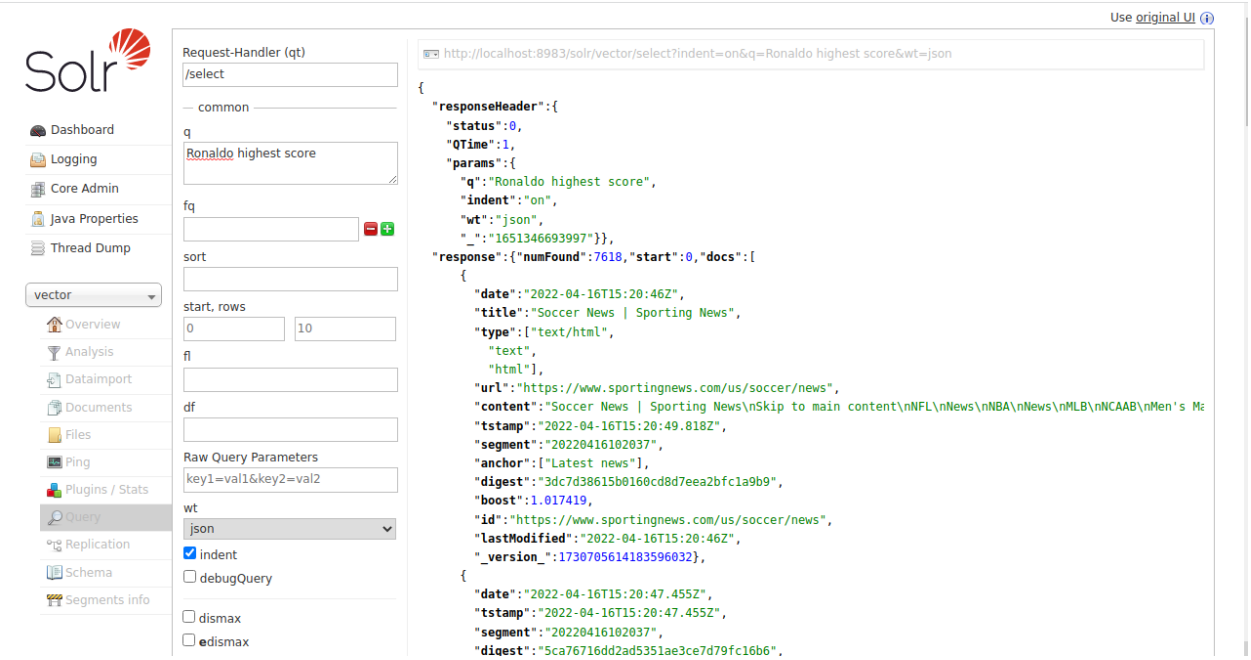
Relevance Models:

Vector Space Relevance model

To score webpages, We do the following:

- Solr provides a tf-IDF-based relevance model, which we used as a vector space relevance model. Documents and queries are both vectors in the vector space relevance model.

- The cosine of the angle between a document vector and a query vector can be used to calculate their similarity. The closeness between the document and the query was determined using the cosine similarity measure.
- This scoring model is based on a number of variables.
    - Term frequency (Tf)
    - Inverse document frequency (IDF)
    - Tf-IDF ($W_{d,t} = tf_{d,t} \times idf_t$)
    - The tf-IDF weighting scheme is the most common weighting scheme in the vector space relevance model.



Page Rank

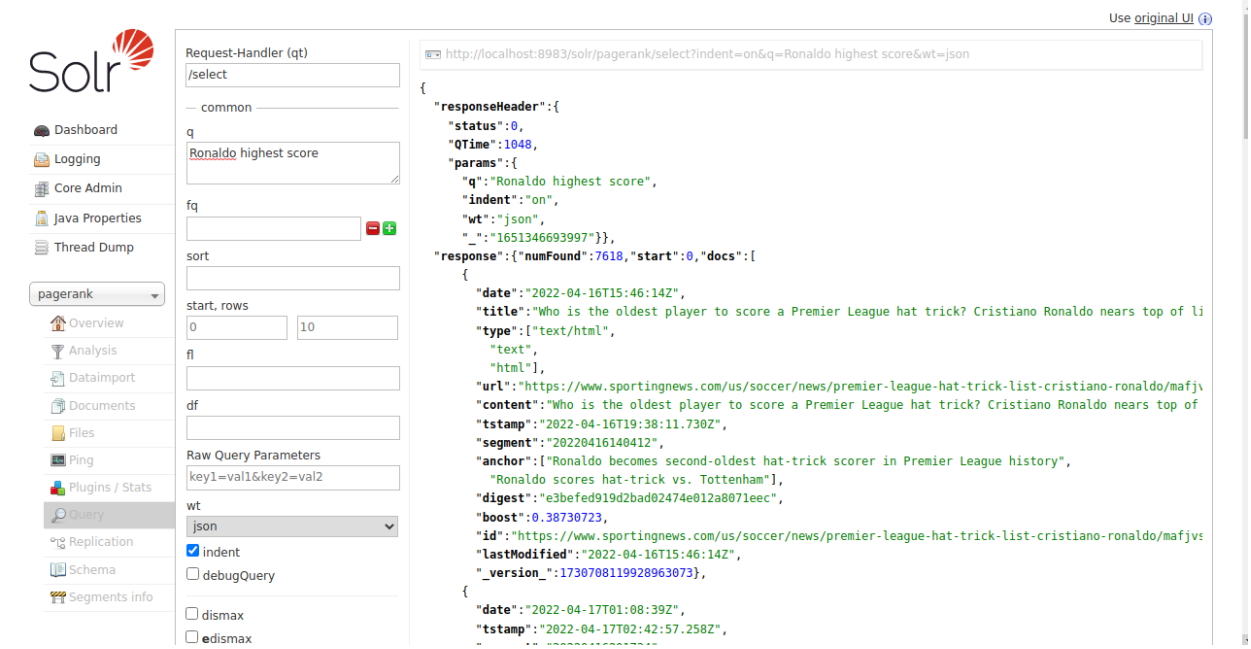
We utilized the built-in nutch function to create a relevance model based on Page Ranking. The procedures we took to apply PageRank scores to a crawled database so that query results would show web pages in descending order of page ranking values are listed below –

Damping factor - 0.85

bin/nutch org.apache.nutch.scoring.webgraph.WebGraph -segmentDir /home/reddy/SoccerCrawlFinal/segments/ -webgraphdb /home/reddy/SoccerCrawlFinal/webgraphdb/

bin/nutch org.apache.nutch.scoring.webgraph.LinkRank -webgraphdb /home/reddy/SoccerCrawlFinal/webgraphdb/

bin/nutch org.apache.nutch.scoring.webgraph.ScoreUpdater -crawldb /home/reddy/SoccerCrawlFinal/crawldb -webgraphdb /home/reddy/SoccerCrawlFinal/webgraphdb



HITS (Hyperlink Induced Topic Search)

Hyperlink Induced Topic Search (HITS) Algorithm is a Link Analysis Algorithm that rates webpages, developed by Jon Kleinberg. This algorithm is used in the web link structures to discover and rank the web pages relevant for a particular search. HITS uses hubs and authorities to define a recursive relationship between web pages.

Given a query to a Search Engine, the set of highly relevant web pages are called Roots. They are potential Authorities. Pages that are not very relevant but point to pages in the Root are called Hubs. I wrote a python program to generate the authority and hub score for the HITS score.

Networkx python library is used to create a graph and the hits algorithm to get hub and the authority scores. I used nutch to build a dump of all the inlinks of crawled pages in order to acquire these inlinks. I constructed outlinks using the inlinks dictionary. Based on page rank results, a root set of web pages is gathered based on the user's query. The root set is converted to a base set by adding the root set's outlinks and inlinks. Now calculations of the HIT score, and

relevant web pages are sorted and returned to the UI for display based on the maximum authority score. Among the websites we crawled were the following:

The highest hub score was assigned to:
0.0432895726356544311 (https://www.uslsoccer.com/)

The highest authority score was assigned to:
0.0027890365732591324 (https://www.horseracing24.com/)

Topic-Based Page Rankings:

I took the following topics for Topic-based page rankings:

1) Soccer Shopping
2) Soccer matches
3) FIFA

Soccer shopping

Pagerank : 4.0518418,
Link :"https://apps.apple.com/us/app/soccer-tv-schedules/id1543713555",
Title :"Soccer TV Schedules on the App Store",

Pagerank :3.927948,
Link :"https://cdcshoppingcart.uchicago.edu/Cart2/Cart",
Title : “Shopping Basket - Shopping Cart”

Pagerank :3.8730723,
Link :"https://www.yahoo.com/lifestyle/tagged/shopping/",
Title :"Shopping | Yahoo Life"

Soccer matches

Pagerank :3.8730723,
Link :"https://apnews.com/article/entertainment-sports-soccer-world-cup-arts-and-c61c2bd6b97ec19af607a94a513d64dc",
Title :"Fox to televise 35 of 64 World Cup matches on main network | AP News",

Pagerank :3.7730723,
Link :"https://www.cbssports.com/wwe/news/2022-wwe-wrestlemania-38-card-matches-rumors-match-card-start-time-results-location-date/",
Title :"2022 WWE WrestleMania 38 card, matches, rumors, match card, start time, results, location, date - CB",

Pagerank :3.340463,
Link :"https://www.sportingnews.com/us/soccer/news/how-watch-mls-tv-channels-streaming-2022-usa-canada-world/ms4vqvm6ebke9ciyb1kkctui",
Title :"How to watch MLS in 2022: TV channels & streaming in USA, Canada and around the world | Sporting New",

FIFA

Pagerank : 3.8730727,
Link :"https://www.fifa.com/",
Title :"FIFA World Cup 2026™",

Pagerank :3.4756375,
Link :"https://www.fifa.com/tournaments/mens/worldcup/canadamexicousa2026",
Title :"FIFA expresses hope for rapid cessation of hostilities and peace in Ukraine",

Pagerank :3.3746198,
Link :"https://www.fifa.com/about-fifa/organisation/fifa-council/media-releases/fifa-expresses-hope-for-rapid-cessation-of-hostilities-and-peace-in-ukraine",
Title :FIFA expresses hope for rapid cessation of hostilities and peace in Ukraine

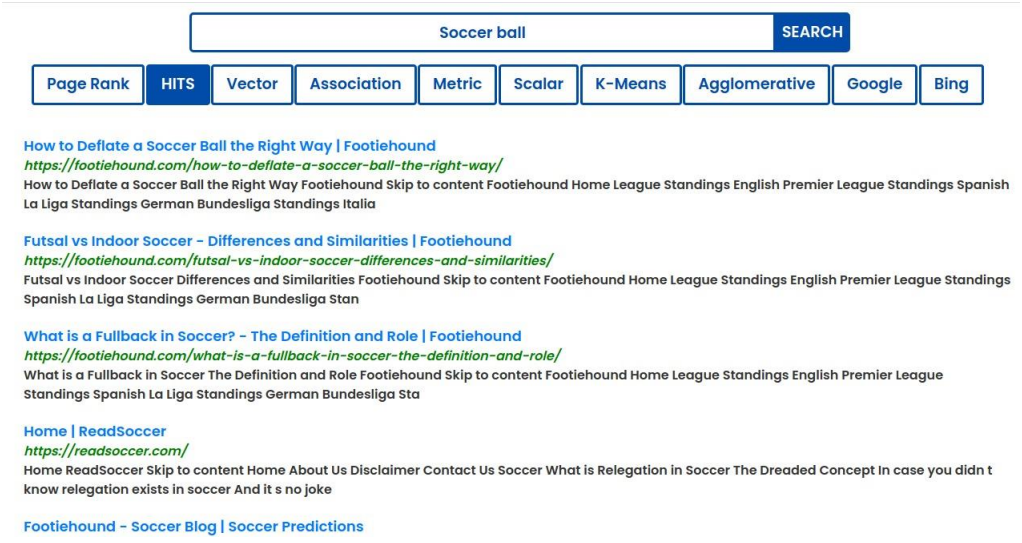
Collaboration with UI and test relevance models results

I have used **25 queries** to test the relevance model. To judge the quality of results we used the following queries:

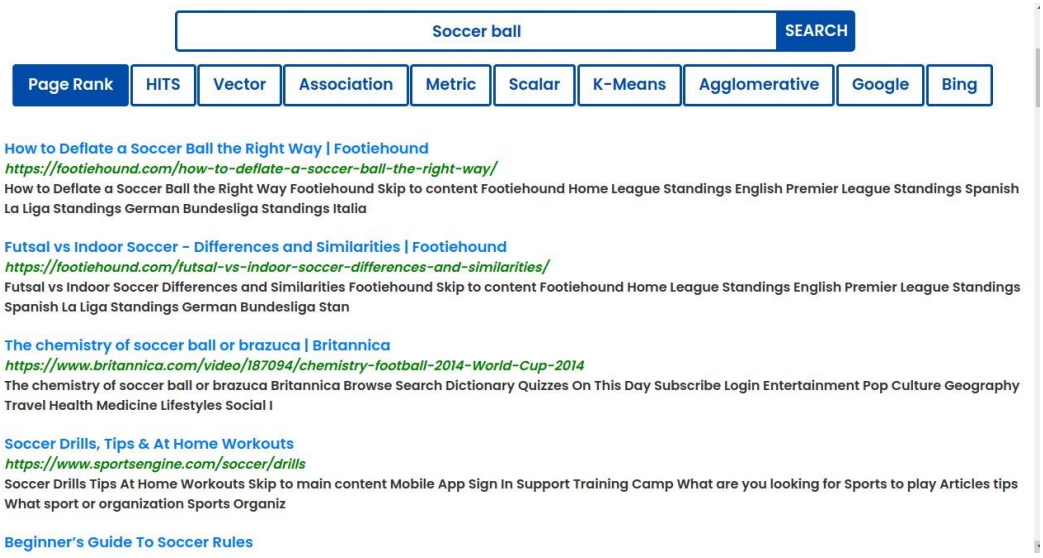
Query 1: Soccer ball
HITS:

Query 1: Soccer ball

HITS

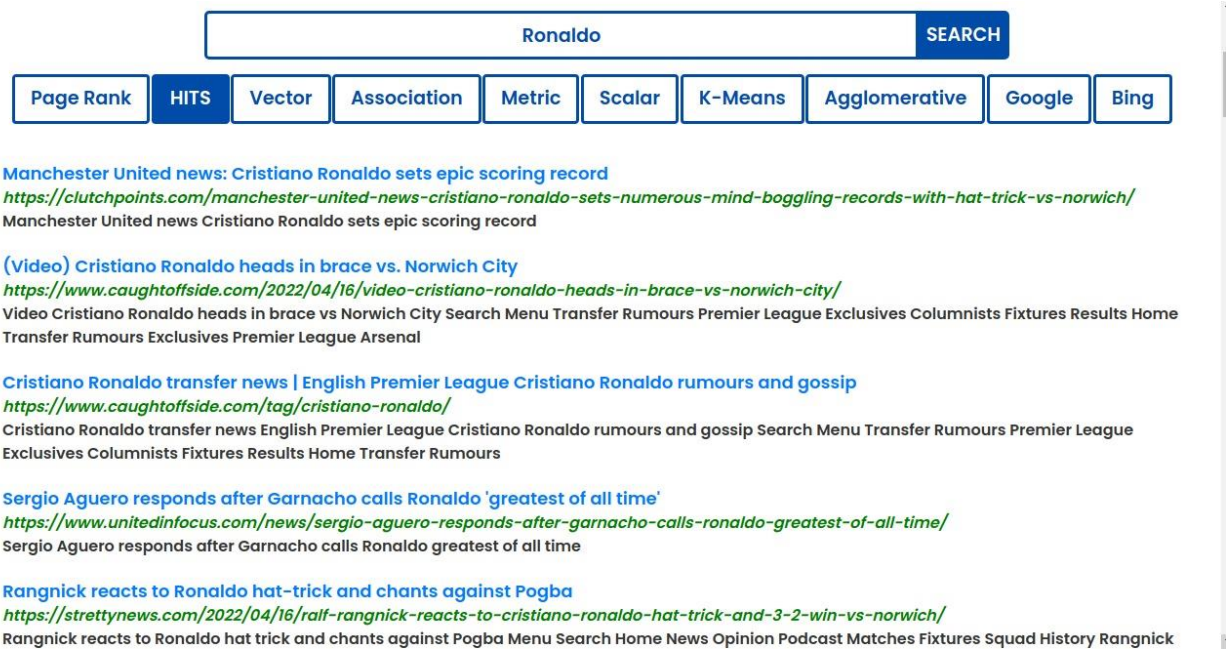


Pagerank

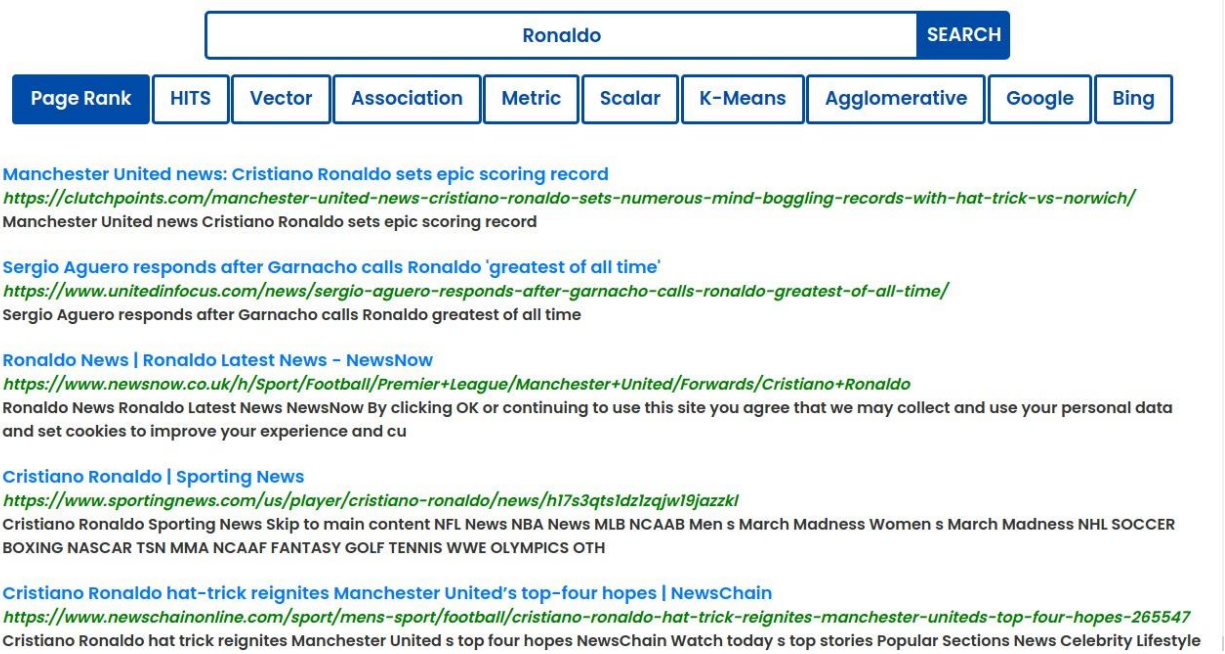


Query 2: Ronaldo

HITS

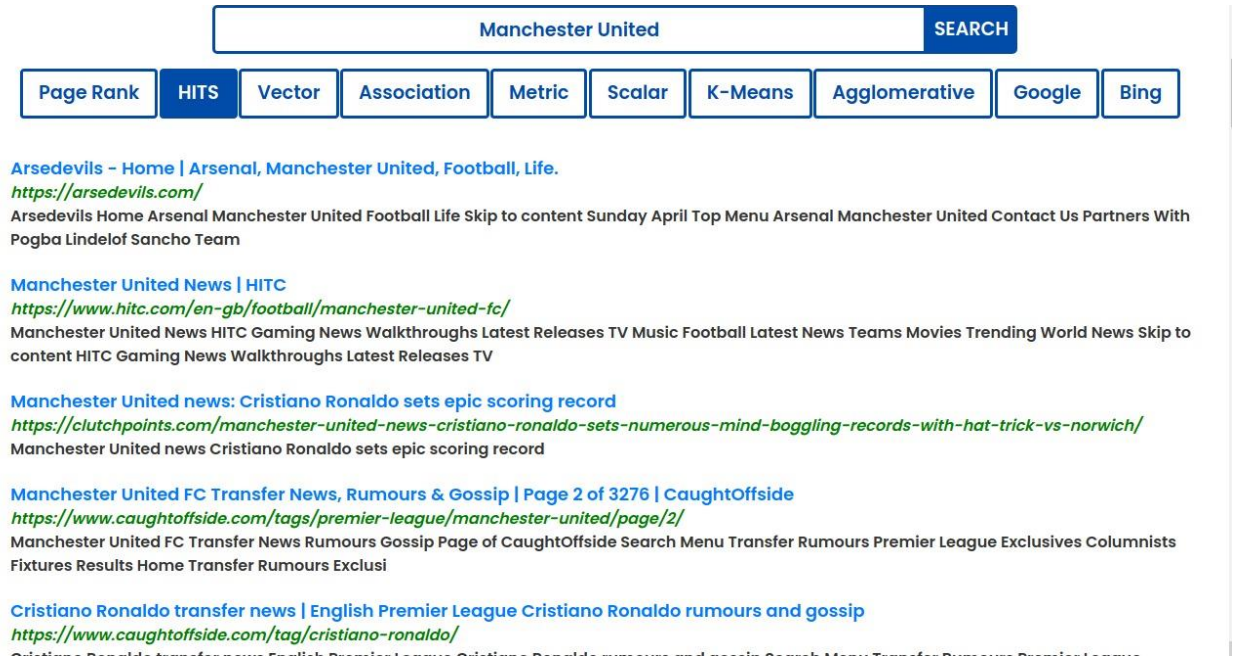


Pagerank

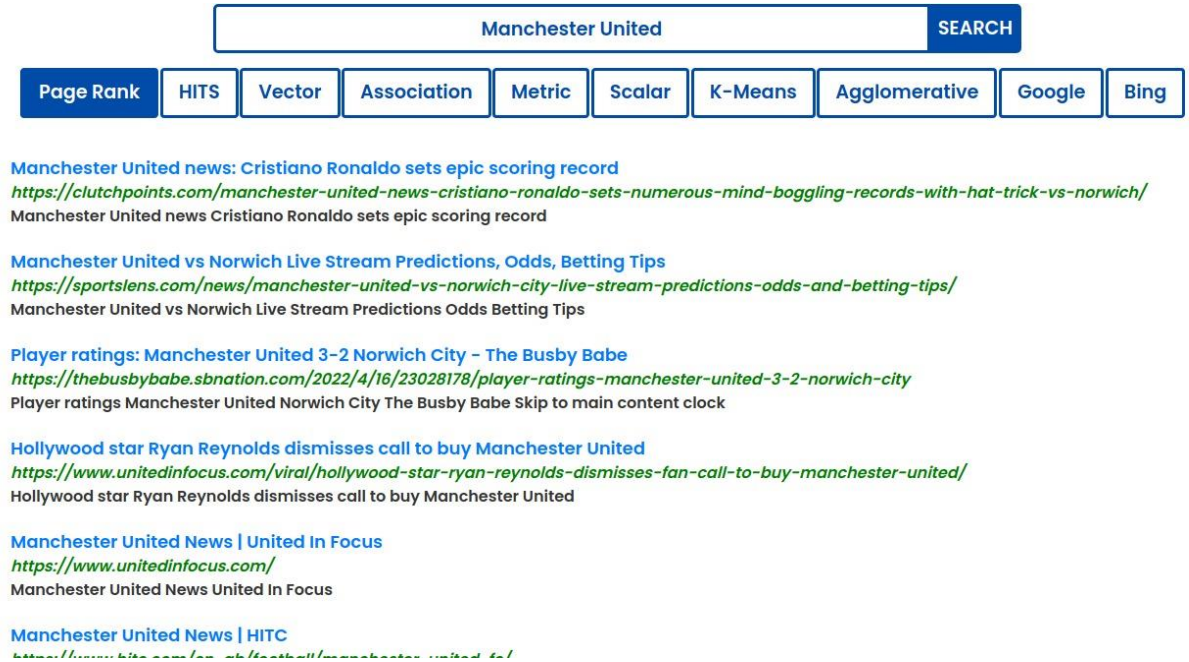


Query 3: Manchester United

HITS



Pagerank



Analysis:- When HITS are used, I am getting better results. Results of HITS are more relevant to the query.

Collaboration with Clustering

Miriyala Sai Chethan Reddy has done the clustering part. We collaborated to get the best out of the relevance models with clustering. We integrated relevance model results with the clustering by which the accuracy of the search engine increased. In most of the cases, clustering gave better results than PageRank. We decided on a process:

- For each query the user enters, the relevance model will give the results of the top 50 to Chethan's clustering program.
- Then, the clustering will rearrange the clusters based on the clusters which have the highly ranked results.

Sai Charan Kotra
SXK210083

User interface and comparisons with Google and Bing

UI design:

The front-end part of our project is prepared using HTML, and CSS, and we have used PHP, Python in the backend for our project.

The user interface has two parts:

- Searching
  - Search bar
  - Search button
  - Buttons for selecting different methods
- Results
  - Different tabs for each method having their respective results.

When a user searches for a query the system will:

- Parse the query entered.
- Send different requests to the API and get the results for that specific method (For google and bing, we present their results in the iframe present in their respective tabs).
- Push the results of json in a particular format with the help of PHP and Python onto the web page.
- Retrieve the pages of Google and Bing onto iframes with the results of the entered query in their respective tabs.
- Display all the results in their respective tabs.

Sai Charan Kotra
SXK210083

Collaboration with Relevance Models:

I have worked with Manthri, Satya Sai Bharadwaj to get results from relevance models. After clicking on the search button, the process starts. I make a request with parameters to the API that is connected to the relevance models. API generates results ordered by ranking.

I and Manthri Satya Sai Bharadwaj tested the search engine with **25 queries**. We tested the connection between APIs, query passing, results flows, and Result Accuracy. We tried to check whether the failure occurs or not in all possible ways.

Parameters passed to each API to build JSON file are as follows:

Page Rank	API returns JSON file with page rank based relevance model
HITS	API returns JSON file with HITS based relevance model
Vector Space	API returns JSON file with Vector space based relevance model
K-Means clustering	API returns JSON file with KMeans Clustering-based relevance model
Agglomerative Clustering	API returns JSON file with Agglomerative Clustering-based relevance model
Association Query Expansion	API returns JSON file with a relevance model based on the metric association query expansion
Metric Query Expansion	API returns JSON file with a relevance model based on metric cluster query expansion
Scalar Query Expansion	API returns JSON file with a relevance model based on metric scalar query expansion

I tried over **15 queries** independently to confirm that the API calls could handle several requests in a row and that each of the different arguments were appropriately provided to the API.

Sai Charan Kotra
SXK210083

Collaboration with Clustering:

Miriyala Sai Chethan Reddy has done the clustering part. He passed all the clustering results to me. We tried a procedure to incorporate clustering results for the query results into the system.

- For each query the user enters, Bharadwaj's relevance model will give the results of the top 50 to Chethan's clustering program.
- Then, the clustering will rearrange the clusters based on the clusters which have the highly ranked results and returns the results.

Presenting results to the Iframe is pretty much the same in terms of output format and presentation.

For the clustering component, I included options in the list for clustering also. Selecting one of the clustering options and clicking on the search button will trigger the API, which will run the process and return the output in a specific format.

Google vs Bing vs Our Search Engine:

Our search engine is not as reliable as major search engines such as Google or Bing. While our search engine can give better results than big search engines on some queries, but it can't compete with Google or Bing as we don't have the access to all web pages as they are having, and also, we are not having an option to maintain the results updated based on the real-time traffic of users.

Outputs:

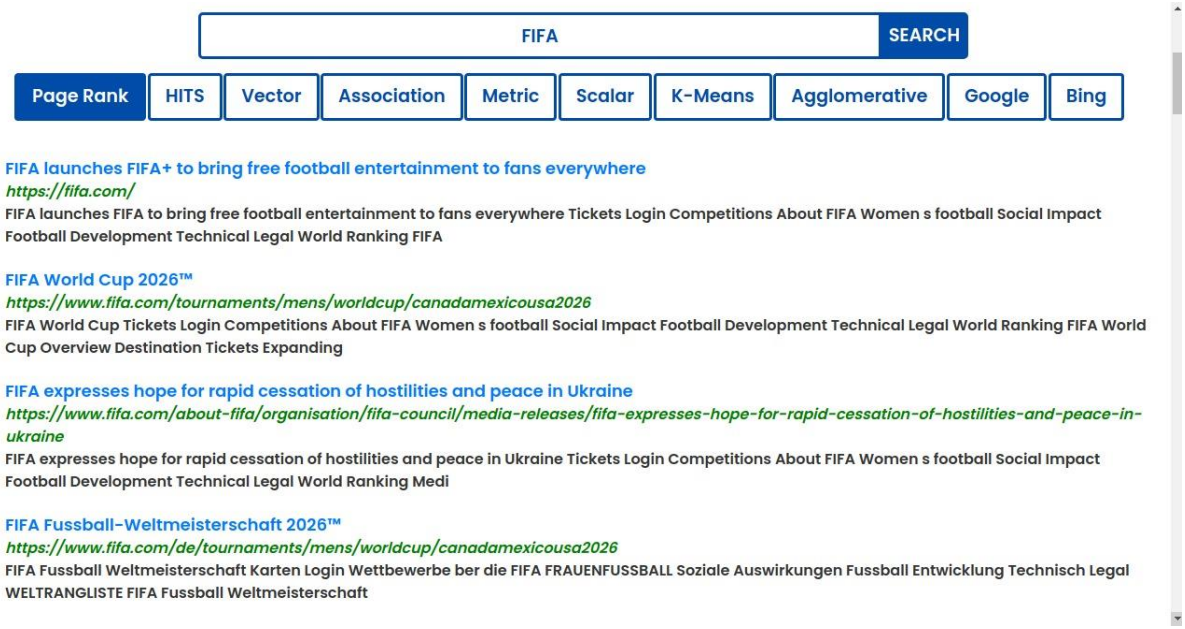
For the demonstration, we picked queries for the example that produced more accurate results and improved when we added clustering, query expansion, and relevance models.

Sai Charan Kotra
SXK210083

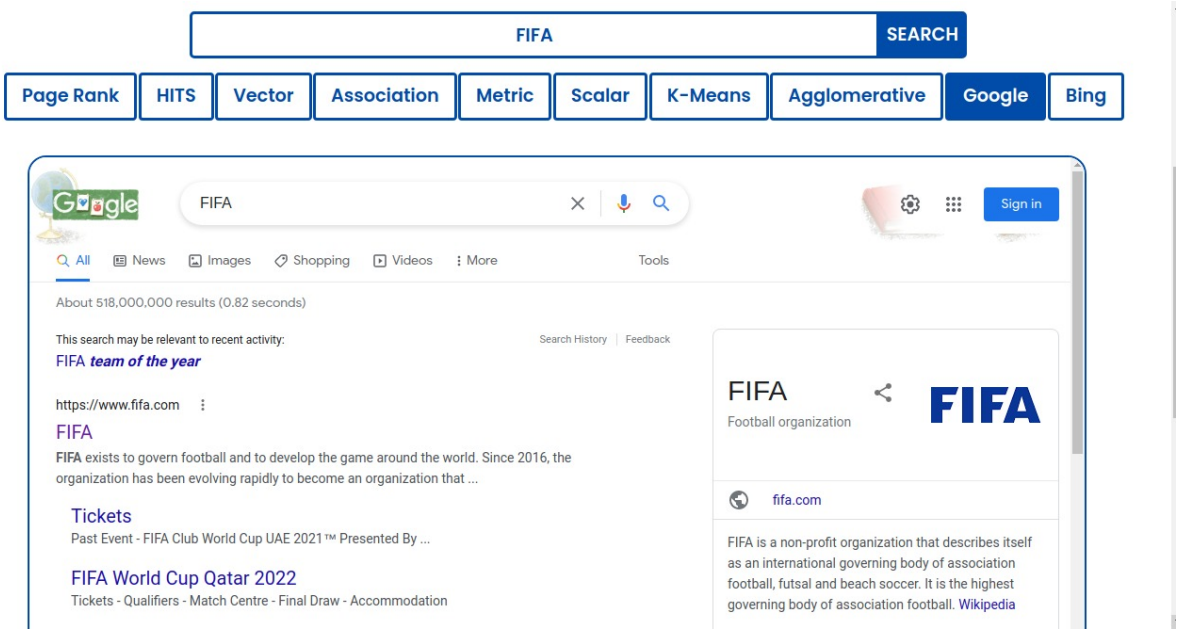
Results of some of the queries which I tried on the search engine are as follows:

Query - 1: FIFA

Our search result:

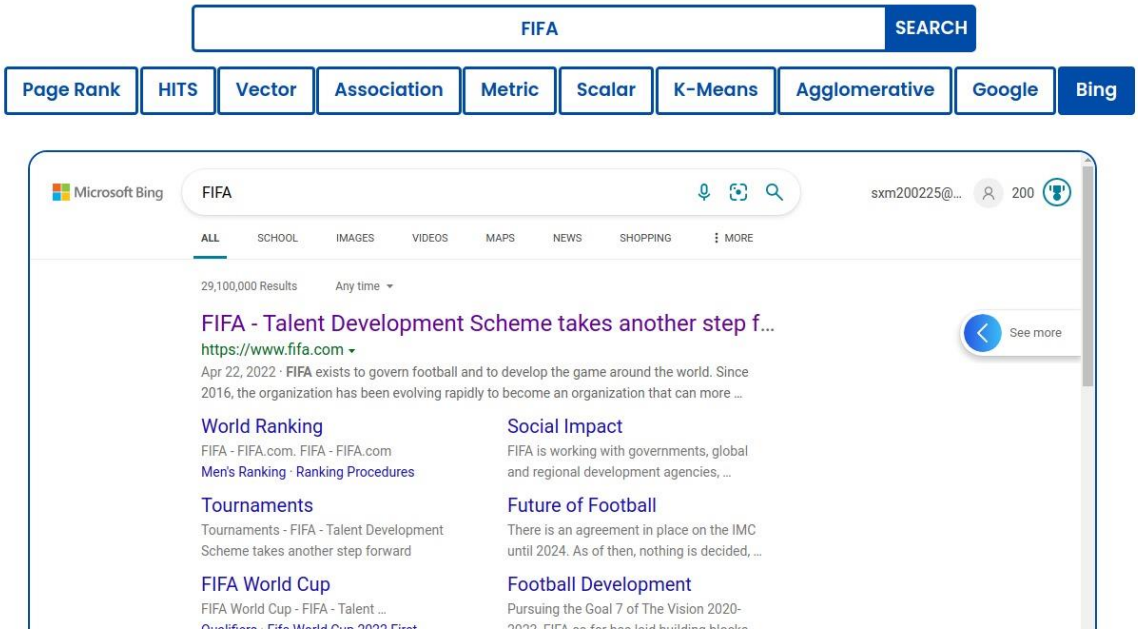


Google's search result:



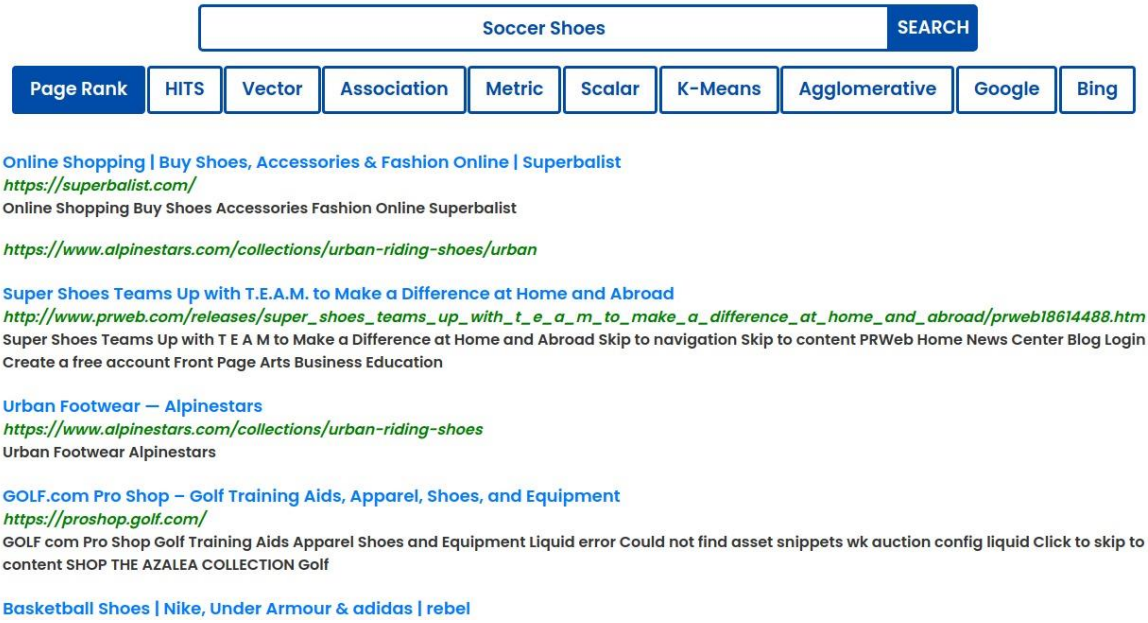
Sai Charan Kotra
SXK210083

Bing's search result:



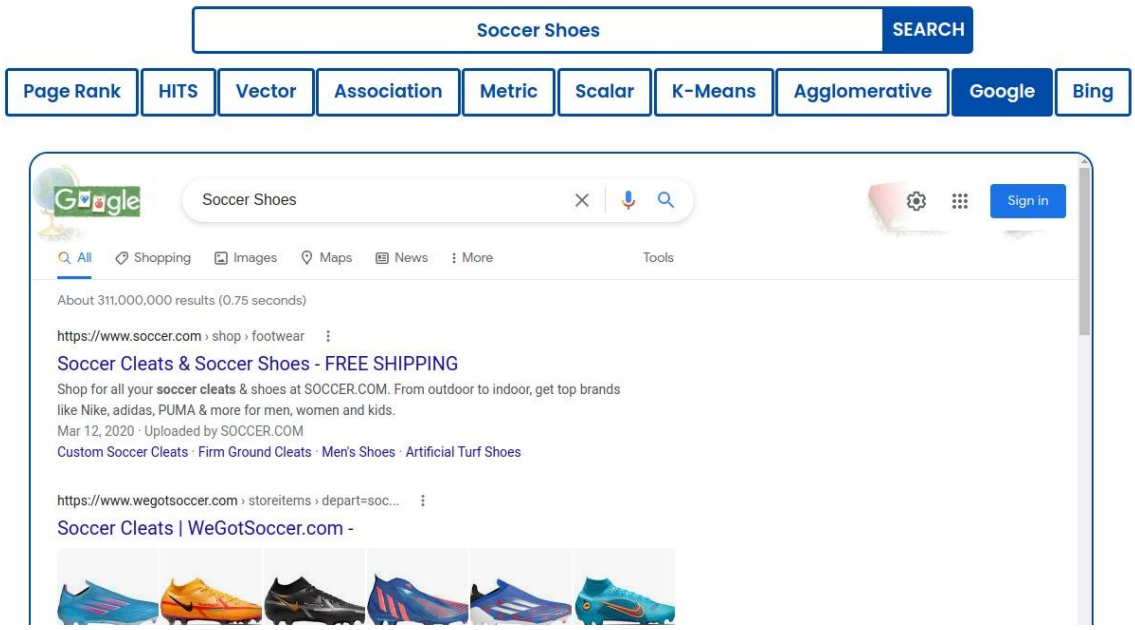
Query - 2: Soccer shoes

Our search result:

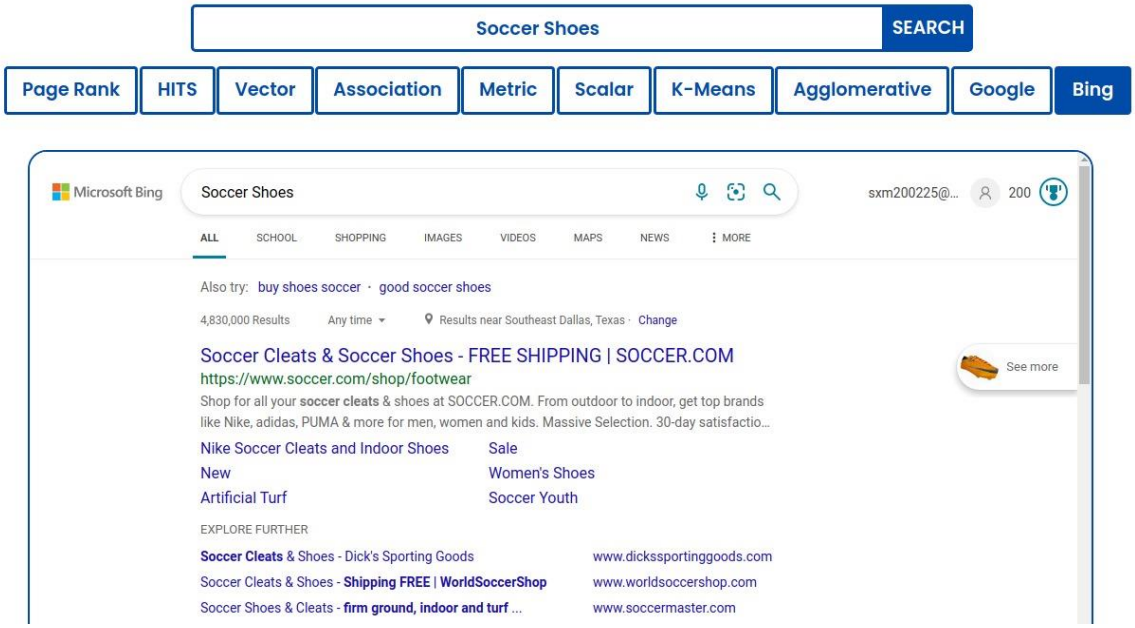


Sai Charan Kotra
SXK210083

Google's search result:



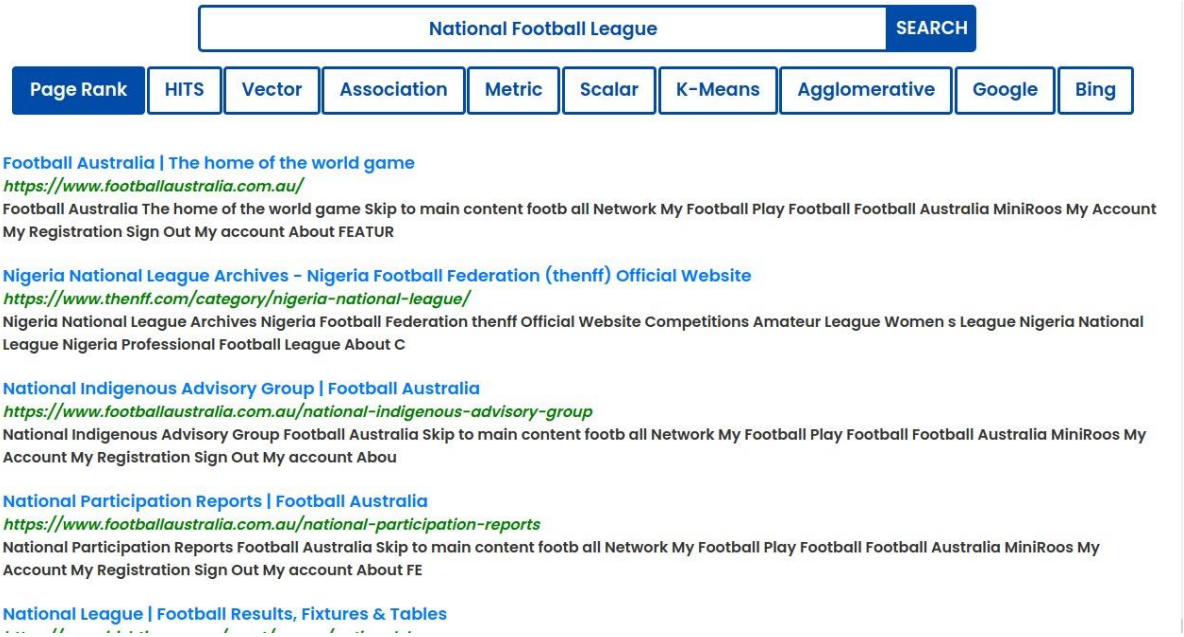
Bing's search result:



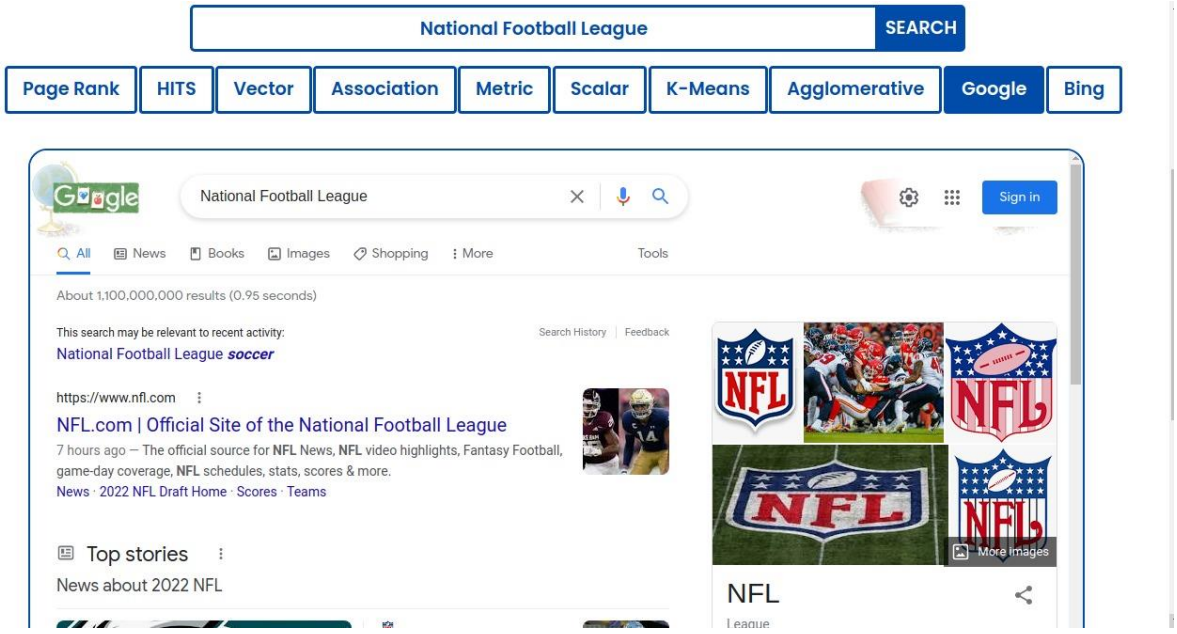
Sai Charan Kotra
SXK210083

Query - 3: National Football league

Our search result:

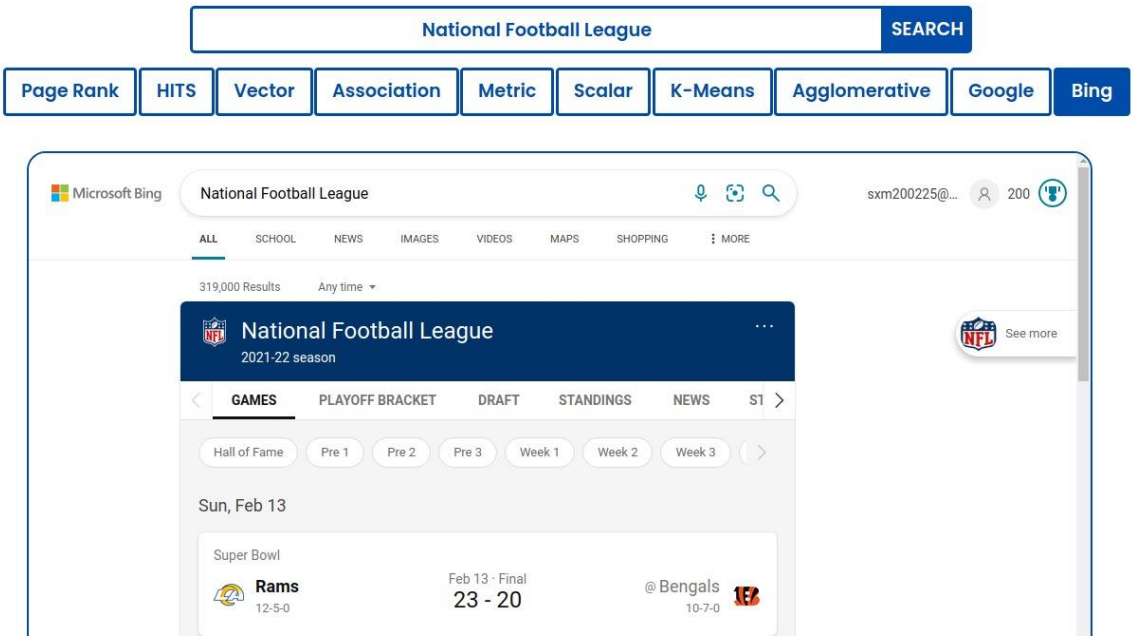


Google's search result:



Sai Charan Kotra
SXK210083

Bing's search result:



Miriyala Sai Chethan Reddy
SXM200225

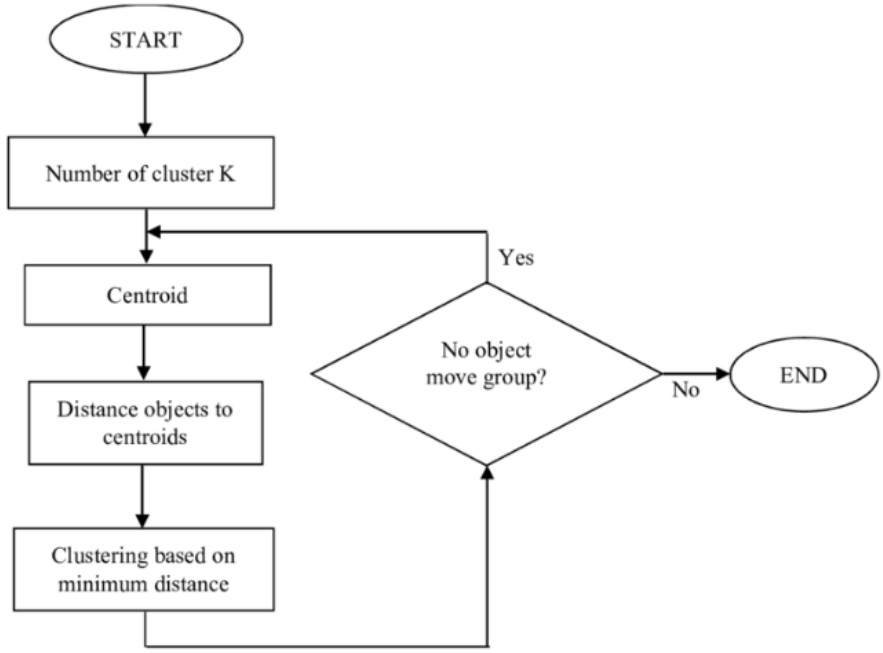
5. Clustering (Sai Chethan Reddy)

Flat Clustering

Flat clustering creates a flat set of clusters without any explicit structure that would relate clusters to each other. For this project, I have designed flat clustering with the K-means centroid algorithm.

K-Means Algorithm

K-means is the most important flat clustering algorithm. Its objective is to minimize the average squared Euclidean distance of documents from their cluster centers where a cluster center is defined as the mean or centroid of the documents in a cluster



Given a set of observations (x1, x2, ..., xn), where each observation is a d-dimensional real vector, K-means clustering aims to partition the n observations into k (≤ n) sets S = {S1, S2, ..., Sk} so as to minimize the within-cluster sum of squares (WCSS). This WCSS is the sum of the squares of distances of all the observations belonging to a cluster from the centroid of the cluster. Formally, the objective is to find:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

where μi is the mean/centroid of points in Si.

Miriyala Sai Chethan Reddy
SXM200225

clustering

I used the response generated by Apache Solr when a query “*” was given. This query retrieved all the crawled documents in a JSON format, and the results were dumped into a file.

```
curl
"http:/localhost:8983/solr/soccer/select?q=*:*&wt=json&indent=true&rows=100
0000000" > soccer_index.json
```

Hierarchical Clustering

Hierarchical clustering (or hierarchic clustering) outputs a hierarchy, a structure that is more informative than the unstructured set of clusters returned by flat clustering. Agglomerative clustering algorithm with the **single-link** is implemented for hierarchical clustering of webpages

Agglomerative clustering

Agglomerative clustering is a bottom-up approach of hierarchical clustering. Here, each observation, document here, starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. The results of hierarchical clustering are usually represented by a dendrogram. For computing distance, there are various distance metrics to choose from Euclidean, Manhattan, etc..,. For the project, I have used the Euclidean distance metric. For computing the linkage criteria, to compute the distance between different clusters, there are a lot of metrics to choose from: maximum or complete link, minimum or single link, unweighted average link, weighted average link, centroid link, ward, etc. For the project, I have used a single link metric.

```
SIMPLEHAC(d_1, ..., d_N)
 1  for n ← 1 to N
 2  do for i ← 1 to N
 3     do C[n][i] ← SIM(d_n, d_i)
 4     I[n] ← 1  (keeps track of active clusters)
 5  A ← []  (assembles clustering as a sequence of merges)
 6  for k ← 1 to N − 1
 7  do ⟨i, m⟩ ← arg max_{⟨i,m⟩:i≠m∧I[i]=1∧I[m]=1} C[i][m]
 8     A.APPEND(⟨i, m⟩)  (store merge)
 9     for j ← 1 to N
10     do C[i][j] ← SIM(i, m, j)
11        C[j][i] ← SIM(i, m, j)
12     I[m] ← 0  (deactivate cluster)
13  return A
```

Miriyala Sai Chethan Reddy
SXM200225

clustering

```
clustering_model = AgglomerativeClustering(n_clusters=None, distance_threshold=1.15, linkage='single')
clustering_model = AgglomerativeClustering(n_clusters=10, distance_threshold=None)
clustering_model = KMeans(n_clusters=10)
```

Note: Agglomerative clustering has both distance_threshold and no of clusters options only one of them can be None.

For both flat and hierarchical clustering I both used TF-IDF vector space and sentence transformers for sentence embeddings. Clustering using sentence transformers generated better clusters, both results are present in the Clustering folder in the source code.

Values of k, ranging from 5 to 15 were tested on a sample of the dataset and the observed clusters were compared to the relevancy of the documents. The Elbow method was also used to determine an appropriate value of k. After observing the results and testing different values of k, a value of **k = 10** gave the most relevant results. As a result, all the fetched documents are stored in one of the **10 clusters**. Similarly, with a distance threshold of 1.15 and a single linkage using agglomerative clustering, a total of **6 clusters** were created.

A sample of the generated clusters from the text file is shown below using both flat clustering(Kmeans) and hierarchical clustering(Agglomerative) are shown below respectively. Each line has both {URL, CLUSTER No}.

#Kmeans

```
http://www.alfiekohn.org/article/case-grades/,3
http://www.australiafootball.com/national-premiers-leagues/news/14/,0
http://www.crdsc-sdrcc.ca/eng/documents/2022-04-05_Sport_Integrity_Commissioner
_Announced_Final_EN.pdf,3
http://www.encyclopedia.chicagohistory.org/pages/665.html,3
https://americanhistory.si.edu/smithsonian-jazz/education/what-jazz,3
https://amp.nine.com.au/article/9ea6811c-038e-45e3-9993-fe2955cb1ee0,10
```

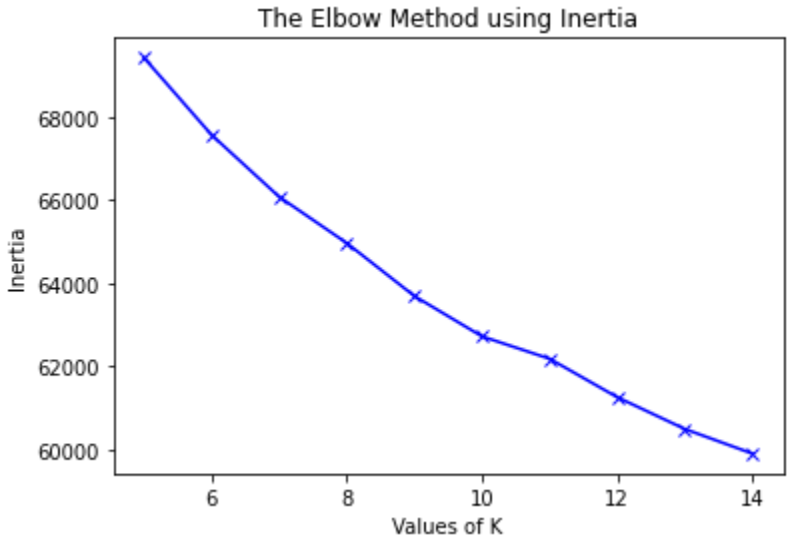
#Agglomerative

```
http://www.australiafootball.com/national-premiers-leagues/news/14/,0
http://www.crdsc-sdrcc.ca/eng/documents/2022-04-05_Sport_Integrity_Commissioner
_Announced_Final_EN.pdf,2
http://www.encyclopedia.chicagohistory.org/pages/665.html,1
https://alphahistory.com/chineserevolution/mao-zedong/,1
https://amp.nine.com.au/article/9ea6811c-038e-45e3-9993-fe2955cb1ee0,0
https://amp.usatoday.com/amp/7035438001,1
https://amp.usatoday.com/amp/7333061001,1
```

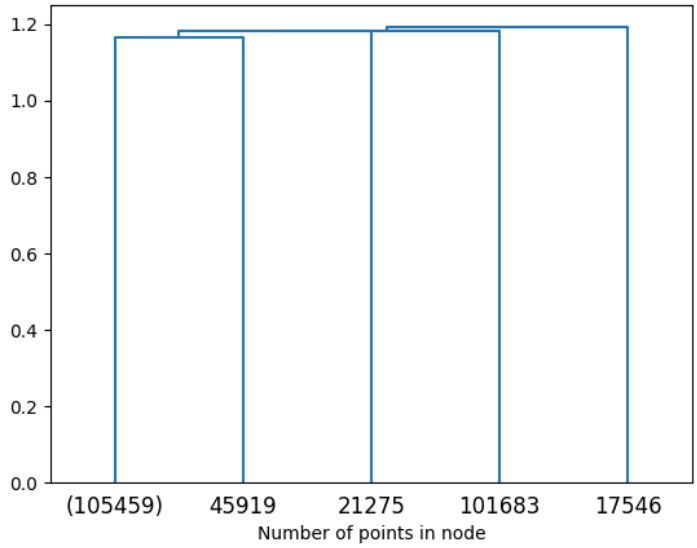
Miriyala Sai Chethan Reddy
SXM200225

clustering

Elbow Method & Dendrogram output



Note: on closer look, we can see there is a bend at k = 10



Miriyala Sai Chethan Reddy
SXM200225

clustering

Number of clusters obtained

Using K-Means: From the elbow method, we obtained **10 clusters**
Using Agglomerate: Using distance threshold and single link we obtained **6 clusters**

Clustering results incorporation and integration with UI

I have used page rank index(solr_index.json) generated by **Bharadwaj** as starting point for clustering the URLs. After I have performed both flat clustering and hierarchical clustering, I created two readable texts (each line contains URL and Cluster assignment) and a cluster reranking program in the app.py file.

In UI done by **Sai Charan** user is presented with two options, K-means & Agglomerative. Two types of values for the “type” parameter in the request query from the UI, are “flat_clustering” and “hierarchical_clustering”. Depending on the clustering mode selected, that particular mode’s clusters are selected for the re-computation of ranks of retrieved pages.

Note: In the clustering folder outputs of K-means clustering using k=5, 10, 15 are present and for agglomerative clustering using 10 clusters, distance_threshold=1.15 are present

Query testing generation and impact on results and relevancy

Around 25 queries were used for each of the two clustering methods: flat and hierarchical, to test the impact of the results of each clustering method. The queries were generated manually, depending on the initial results obtained from the original relevance model, using page rank and HITS. The results obtained after applying clustering seem relevant, as similar results are grouped together to be shown earlier than the non-relevant results.

Observations are also provided below the screenshots which show the impact of clustering on improving the relevancy of the retrieved results.

Miriyala Sai Chethan Reddy
SXM200225

clustering

Incorporation of clusters in the relevance models

Algorithm 1 Using Clustering to Improve Search Results

```
Get URL and Cluster mapping from the precomputed txt files
Set rank = 1
Initialize output = [ ]
for every response in API responses do
    Set status as False
end for
for response in response list do
    if response[status] is False then
        Set response[rank] = rank
        Initialize cluster = response[cluster]
        Increment rank
        set response[status] = True
        Append response to output
        for leftout in response list do
            if leftout[status]=False and leftout[cluster]=cluster then
                Set leftout[rank] = rank
                Set leftout[status] = True
                Increment rank
                Append leftout to output
            end if
        end for
    end if
end for
```

P.S: Above algorithm works the same for both flat and hierarchical clustering, only the mapping changes

Selection of Queries

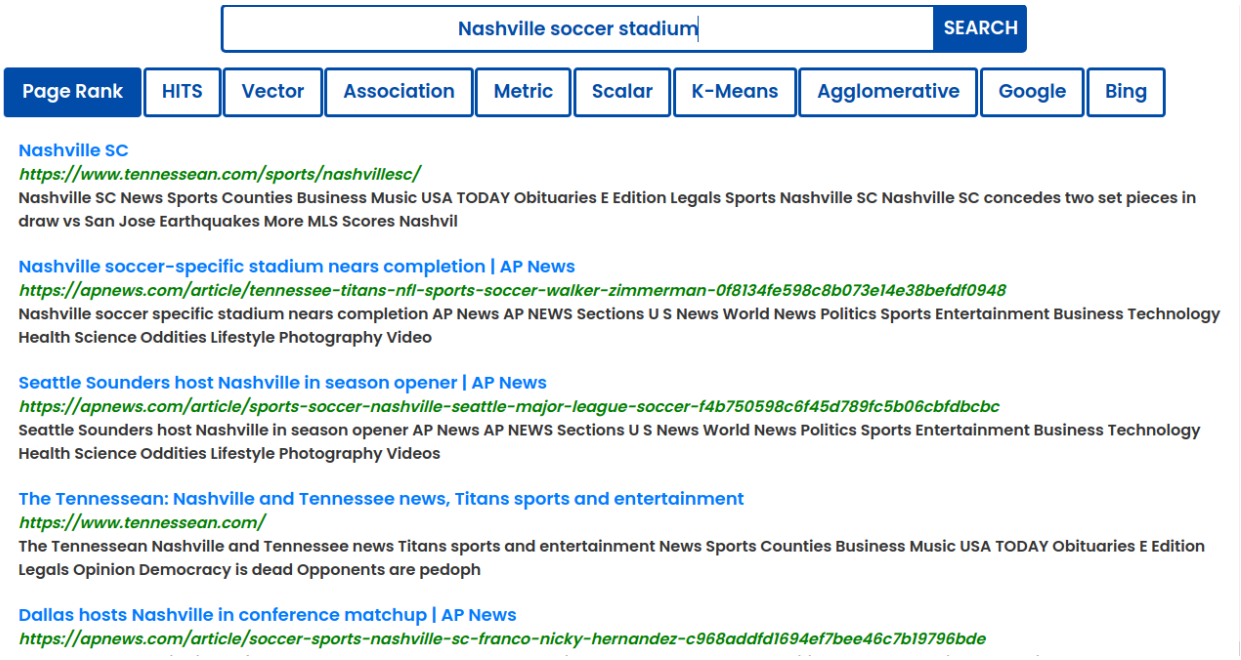
The criteria for the selection of queries is based on the observation obtained from the type of clusters formed for both the clustering methods used. As shown in the sample subset of the results obtained above, one of the query selection criteria such as team, or schedule or event, etc.., For example, official sites are ranked together and news sites on that topic are grouped together.

Miriyala Sai Chethan Reddy
SXM200225

clustering

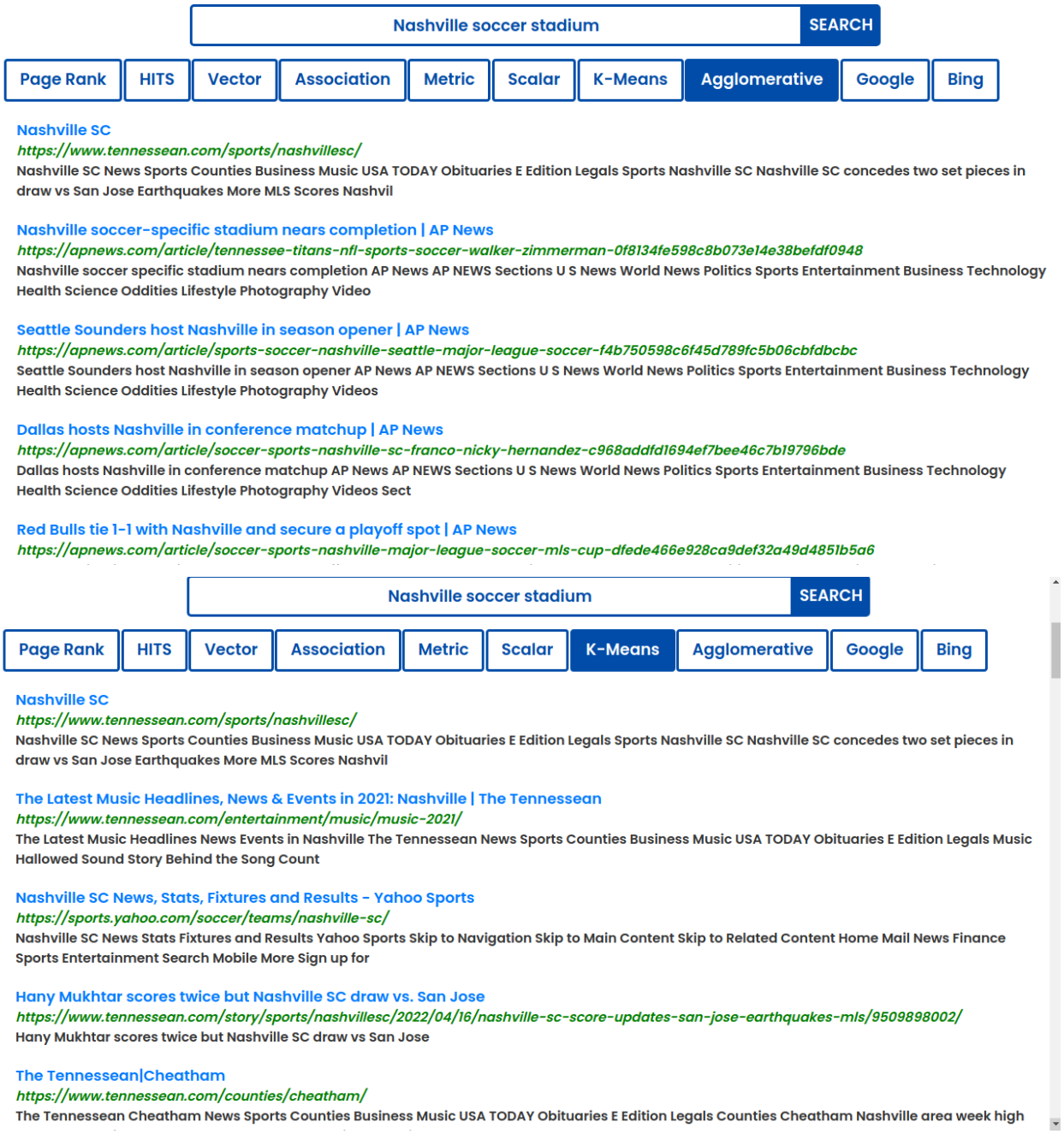
Query examples with clustering

Query #1:Nashville soccer stadium



Miriyala Sai Chethan Reddy
SXM200225

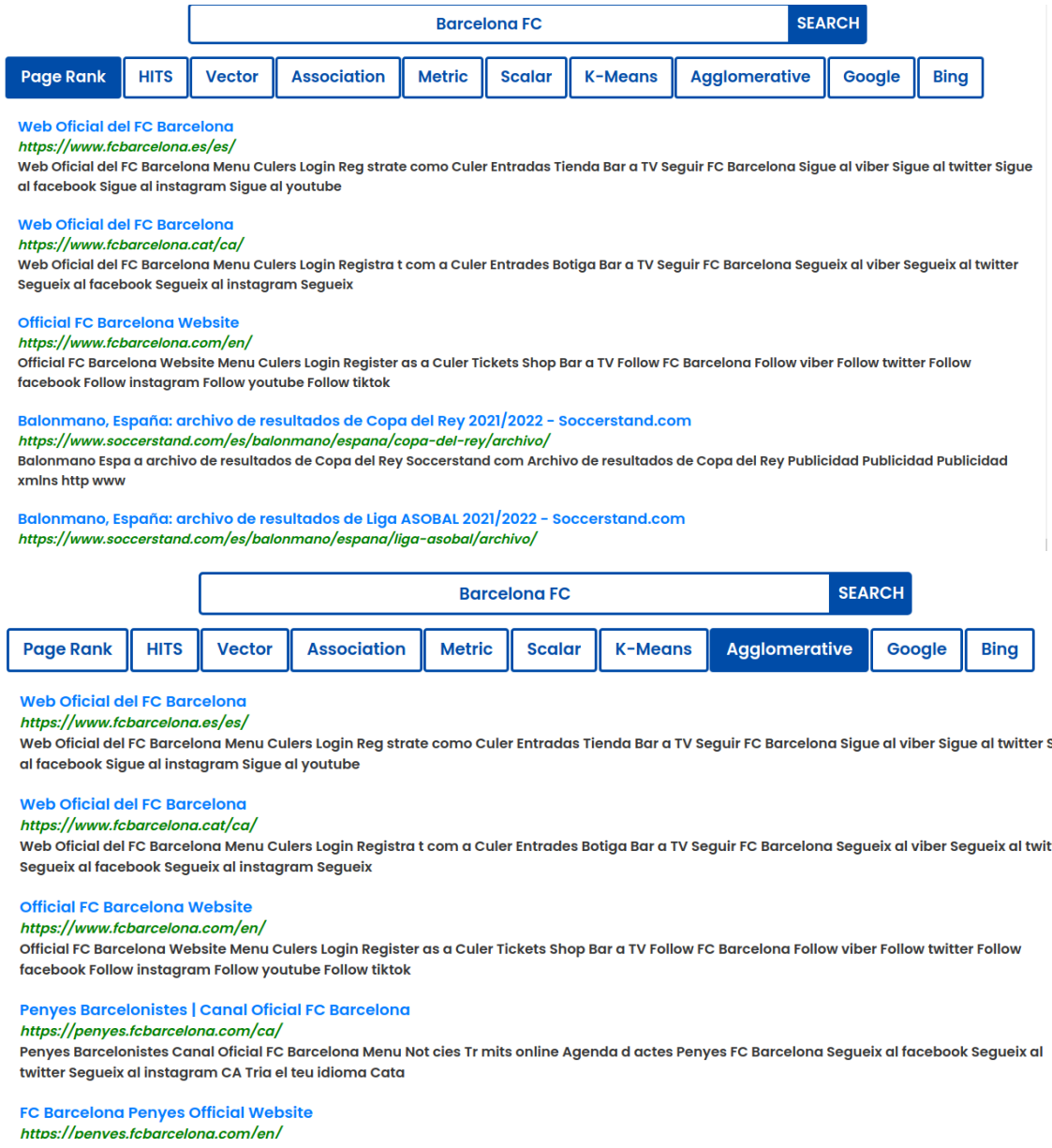
clustering



Query #2: Barcelona FC

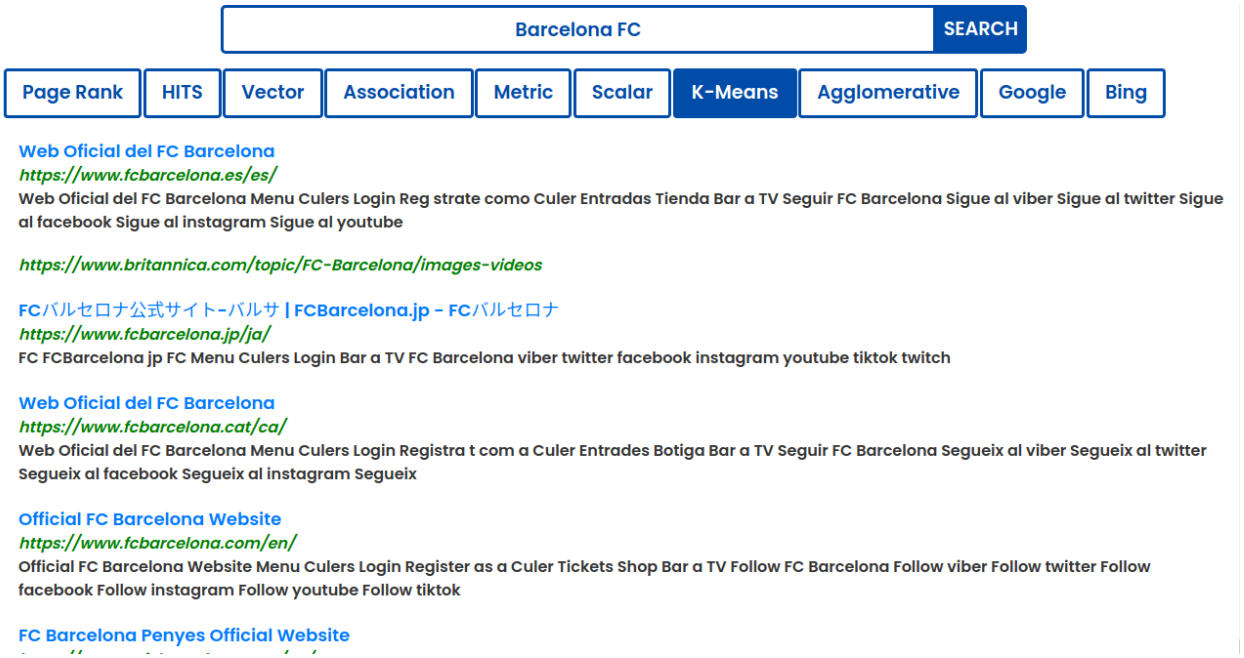
Miriyala Sai Chethan Reddy
SXM200225

clustering

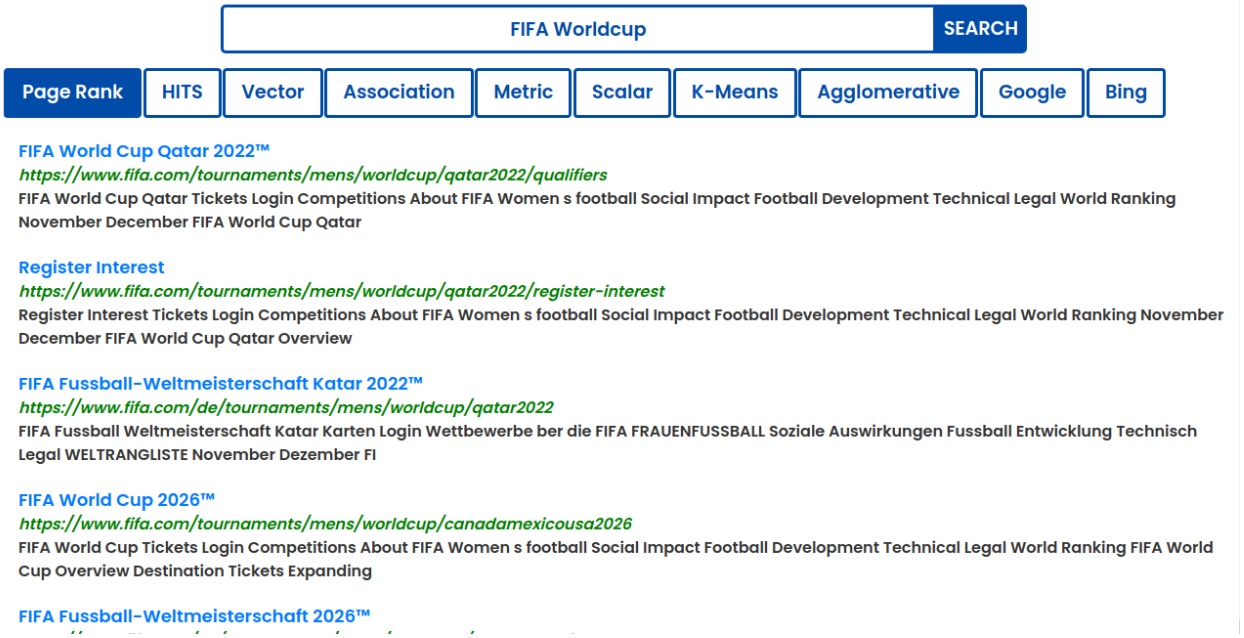


Miriyala Sai Chethan Reddy
SXM200225

clustering

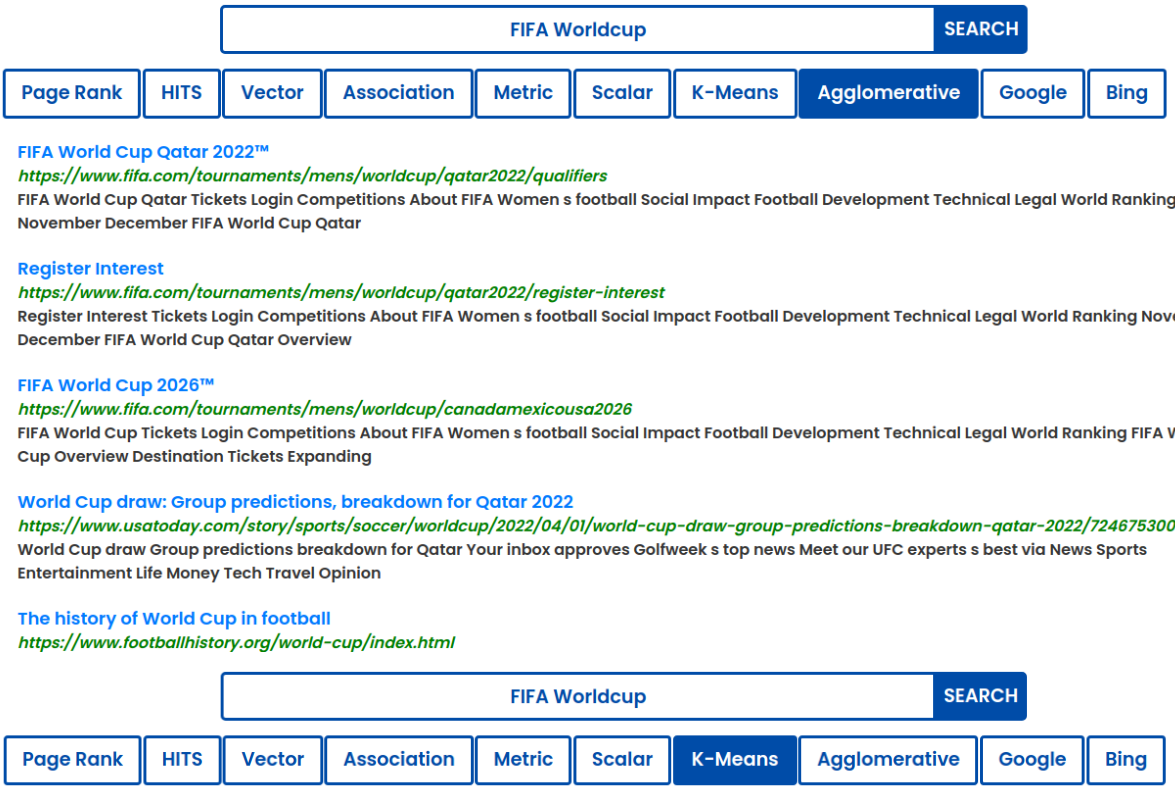


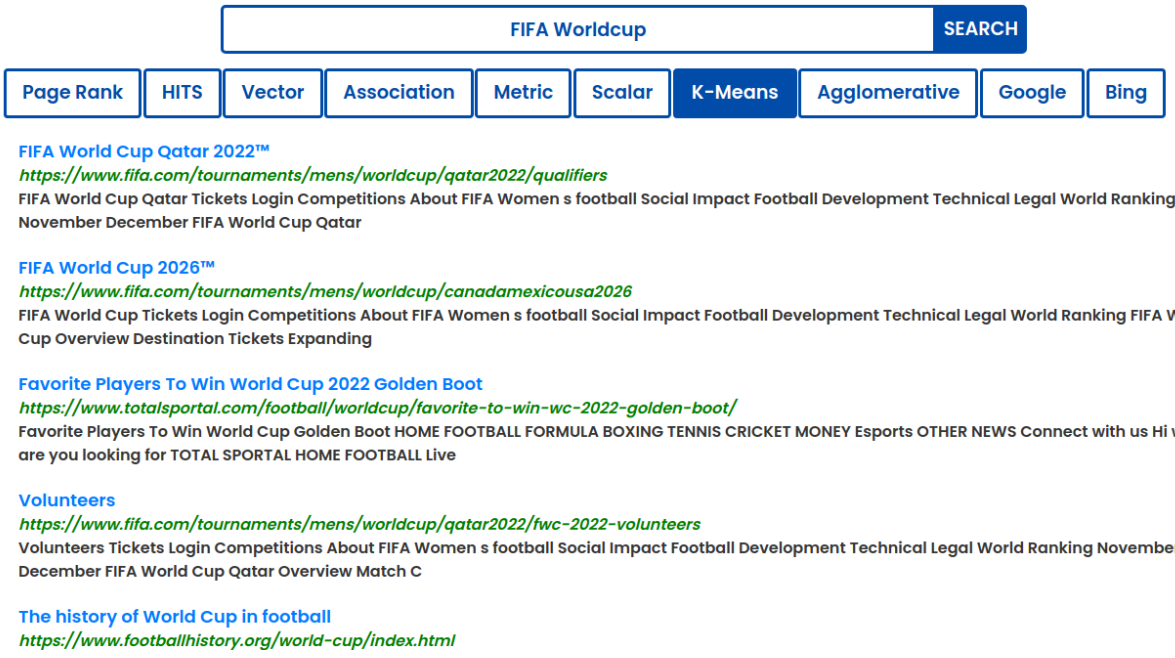
Query #3: FIFA Worldcup



Miriyala Sai Chethan Reddy
SXM200225

clustering





Query #4: US Soccer Team

Miriyala Sai Chethan Reddy
SXM200225

clustering

US Soccer Team SEARCH

Page Rank | HITS | Vector | Association | Metric | Scalar | K-Means | Agglomerative | Google | Bing

United States men's national soccer team all-time leading goal scorers
https://www.usatoday.com/picture-gallery/sports/2017/03/15/us-mens-national-team-all-time-leading-goal-scorers/99235450/
United States men s national soccer team all time leading goal scorers

Read U.S. Soccer - MLS & US national team, news, opinions, articles and interviews
https://readussoccer.com/
Read U S Soccer MLS US national team news opinions articles and interviews Menu Toggle search Account Sign in Register Terms and Partners About Us Contact Us Terms Conditions Privacy Policy

Belize National Soccer Team Stopped by Armed Gang Before World Cup Qualifier
https://www.insider.com/belize-national-soccer-team-stopped-gang-world-cup-qualifier-2021-3
Belize National Soccer Team Stopped by Armed Gang Before World Cup Qualifier

Top photos of the U.S. women's national soccer team through the years
https://www.usatoday.com/picture-gallery/sports/soccer/2019/05/29/uswnt-top-photos-through-years/1265161001/
Top photos of the U S women s national soccer team through the years

Kessler to replace injured Zimmerman on US Gold Cup roster | AP News
https://apnews.com/article/2020-tokyo-olympics-soccer-sports-international-soccer-jamaica-olympic-team-ec86b4c62e85cad32bf95826e019731d
Kessler to replace injured Zimmerman on US Gold Cup roster AP News AP NEWS Sections U S News World News Politics Sports Entertainment Business Technology Health Science Oddities Lifestyle Photograp

US Soccer Team SEARCH

Page Rank | HITS | Vector | Association | Metric | Scalar | K-Means | Agglomerative | Google | Bing

United States men's national soccer team all-time leading goal scorers
https://www.usatoday.com/picture-gallery/sports/2017/03/15/us-mens-national-team-all-time-leading-goal-scorers/99235450/
United States men s national soccer team all time leading goal scorers

Read U.S. Soccer - MLS & US national team, news, opinions, articles and interviews
https://readussoccer.com/
Read U S Soccer MLS US national team news opinions articles and interviews Menu Toggle search Account Sign in Register Terms and Partners About Us Contact Us Terms Conditions Privacy Policy

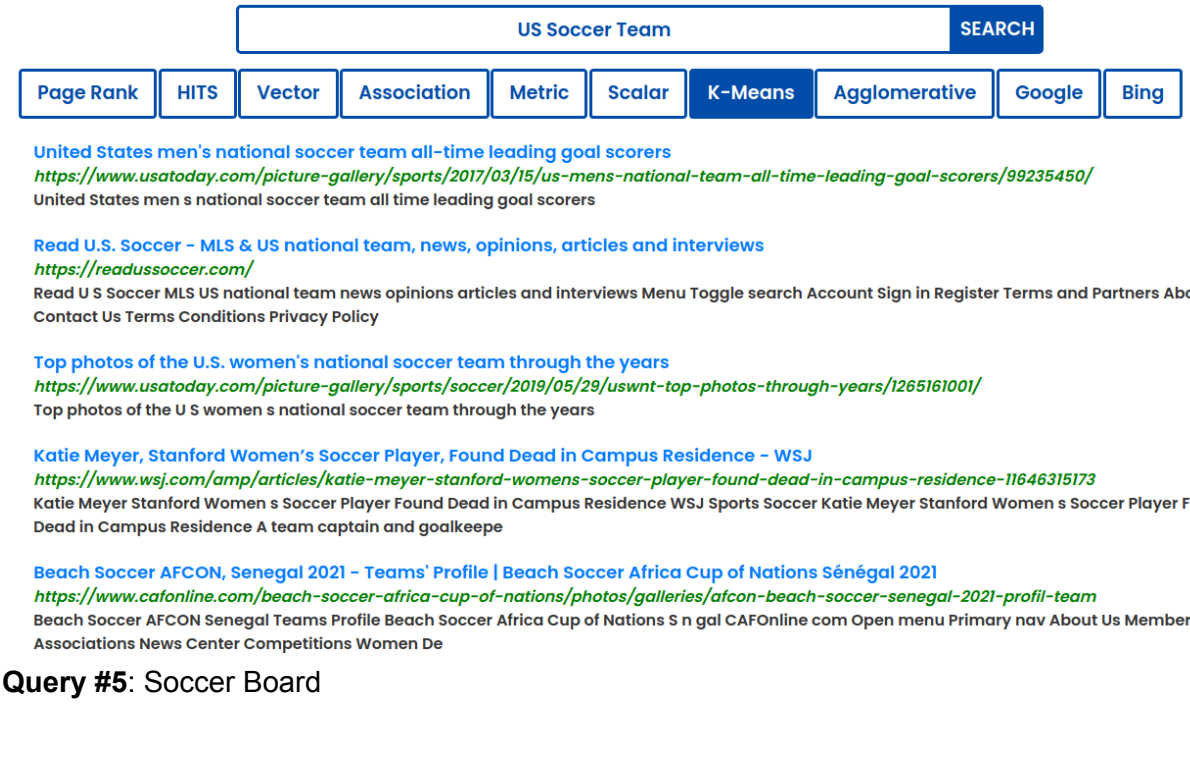
Lewandowski & Batshuayi lead FIFA 18 Bundesliga Team of the Season | Sporting News
https://www.sportingnews.com/us/soccer/list/lewandowski-batshuayi-lead-fifa-18-bundesliga-team-of-the-season/1ra2n7lovmziu1b8ntxsdfh77n
Lewandowski Batshuayi lead FIFA Bundesliga Team of the Season Sporting News Skip to main content NFL News NBA News MLB NCAAB Men s March Madness Women s March Madness NHL SOCCER BOXING NASCAR T

Home | ReadSoccer
https://readsoccer.com/
Home ReadSoccer Skip to content Home About Us Disclaimer Contact Us Soccer What is Relegation in Soccer The Dreaded Concept In case you didn t know relegation exists in soccer And it s no joke

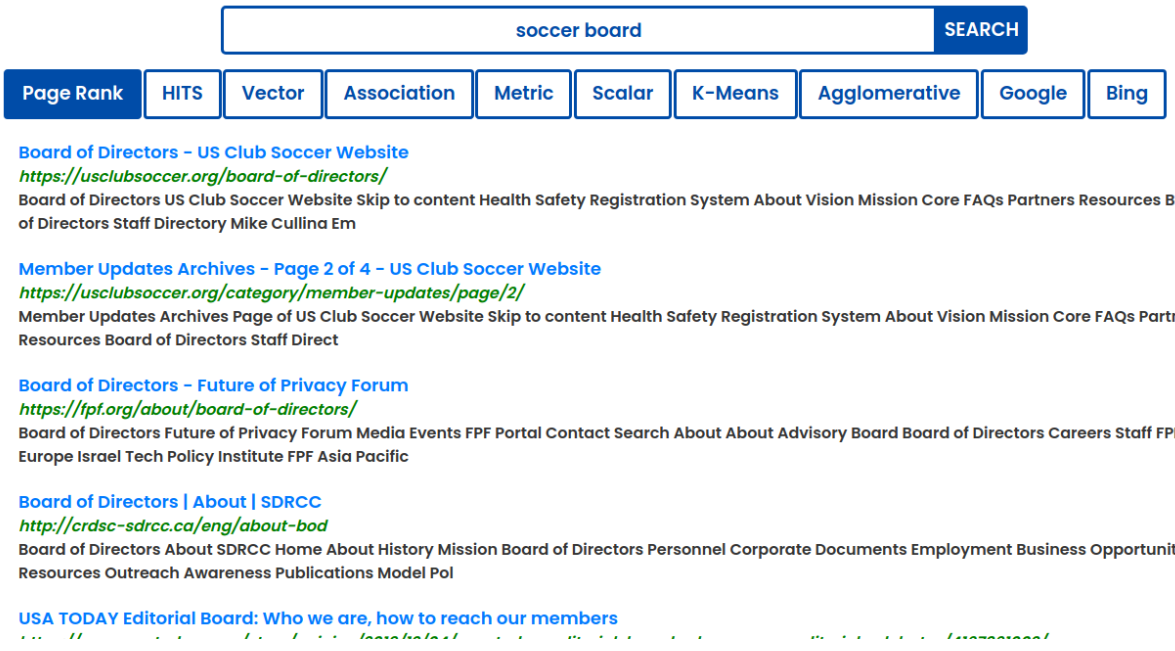
Olympics | Sporting News
https://www.sportingnews.com/us/competition/olympics/news/39fbqm5w9yu86agk4g5mkxl1x
Olympics Sporting News Skip to main content NFL News NBA News MLB NCAAB Men s March Madness Women s March Madness NHL SOCCER BOXING

Miriyala Sai Chethan Reddy
SXM200225

clustering

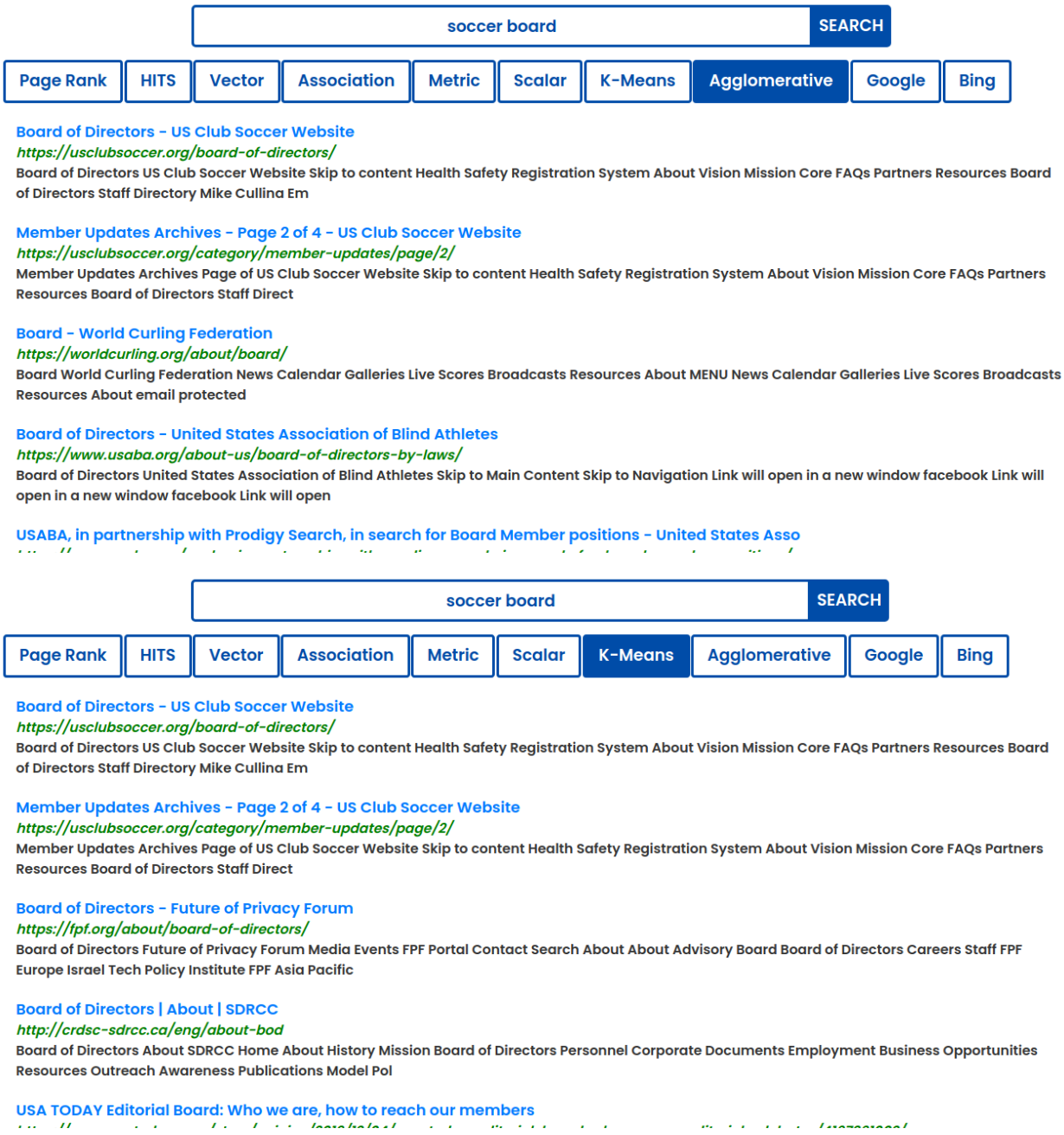


Query #5: Soccer Board



Miriyala Sai Chethan Reddy
SXM200225

clustering



Observation: From the above images, we can see that links/URLs that belong to the same cluster are presented together

Vishnu Vardhan Reddy Kanamata Reddy
[VXK210042]

6.Query Expansion and Relevance Feedback:

Query expansion is used to improve search results by following local methods:
1) Relevance Feedback (Roccio Algorithm)
2) Pseudo Relevance Feedback
    a) Association cluster
    b) Metric cluster
    c) Scalar cluster

The following formula is used to implement Roccio algorithm:

- Used in practice:

$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}}\vec{d}_j$$

- D_r = set of known relevant doc vectors
- D_{nr} = set of known irrelevant doc vectors
    - Different from C_r and C_{nr} 
- q_m = modified query vector; q_0 = original query vector; α,β,γ: weights (hand-chosen or set empirically)
- New query moves toward relevant documents and away from irrelevant documents

The Roccio algorithm was tried with alpha =1.0, beta = 0.9, and gamma =0.1 values, in order to find a set of weights that can differentiate important texts from irrelevant materials, using the 20 queries we selected as input.

The 20 queries were chosen using sporting events, team names, and well-known soccer players as criteria.

- ronaldo records
- messi
- fifa
- football stadium
- beach soccer
- Barcelona FC
- Manchester united
- UEFA champions
- olympics
- commonwealth
- soccer shopping
- soccer men diet
- fifa 2021 scores
- football playoffs
- football field
- soccer schedule
- football matches
- soccer score
- football leagues
- soccer men rankings

Examples of web pages that were found relevant to sample queries:

Query	**Relevant Web Pages**
fifa	https://www.fifa.com/
	https://www.fifa.com/tournaments/mens/worldcup/canadamexicousa2026
	https://www.fifa.com/about-fifa/organisation/fifa-council/media-releases/fifa-expresses-hope-for-rapid-cessation-of-hostilities-and-peace-in-ukraine
football stadium	https://www.fifa.com/tournaments/mens/worldcup/qatar2022/destination/ahmed-bin-ali-stadium
	https://www.oceaniafootball.com/projects/freshwater-football-stadium-project/

	https://www.fifa.com/tournaments/mens/worldcup/2018russia/news/saint-petersburg-stadium-2664617
Barcelona FC	https://www.fcbarcelona.es/es/
	https://www.fcbarcelona.cat/ca/
	https://www.fcbarcelona.com/en/

Examples of web pages that were found irrelevant to sample queries:

Query	**Irrelevant Webpages**
fifa	https://www.oceaniafootball.com/news/
	https://www.oceaniafootball.com/governance/
	https://worldsoccertalk.com/
football stadium	https://www.oceaniafootball.com/technical/womens-football/contacts/
	https://www.oceaniafootball.com/technical/womens-football/programmes/
	https://footballleagueworld.co.uk/11-quickfire-quiz-questions-about-sunderlands-stadium-that-all-black-cats-supporters-should-get-correct/amp/
Barcelona FC	https://www.caughtoffside.com/tags/la-liga/
	https://www.soccerstand.com/es/balonmano/espana/copa-del-rey/archivo/
	https://kids.britannica.com/students/article/FC-Barcelona/544996/related#nodeId=main&page=1

Queries after applying the Roccio algorithm:

Original Query	**Query after Roccio algorithm**
fifa	fifa score
ronaldo	ronaldo records

fifa 2021	fifa 2021 worldcup
football	football field
messi	messi score
nike soccer	nike soccer shoes
Barcelona	Barcelona FC
Manchester	manchester united
UEFA champions	UEFA champions league
famous soccer	famous soccer teams

For the above-mentioned 20 inquiries, this method was employed. The results were not promising, and there was no way to come up with a universal set of weights. Some of the findings were as follows:

1. Some queries returned no results – for example, "soccer ball" "football athlete diet," and so on – indicating that the data we crawled did not contain the pages or any references to these terms. In such circumstances, the relevance model failed.
2. Some queries returned completely irrelevant documents - for example, "soccer movies" returned results for "soccer movies worldcup games" As a result, the query was changed to "fifa world cup."
3. Some inquiries yielded highly relevant results, such as "fifa" which returned data on the fifa world cup in 2021 as well as 2020.
4. Problems with performance Because some queries required too much time to obtain stabilized weights, they could not be completed on the fly
5. Each query necessitated its own set of weights.

Because of the aforementioned concerns, it was decided not to publish the code for the Rocchio algorithm for extending the query; instead, the observations' findings were used in pseudo relevance feedback approaches to filter out stop words, eliminate spelling mistakes, and so on.

Pseudo Relevance Feedback:

Vishnu Vardhan Reddy Kanamata Reddy
[VXK210042]

1. Association cluster: The concept is that stems that appear frequently in documents share a synonymy relationship.
2. Metric cluster: The concept is that stems that appear widely apart in the document have a lower correlation with terms that appear close together, such as in the same sentence.
3. Scalar cluster: The theory is that if two stems have similar surroundings, they will be more associated.

The 54 queries used for pseudo relevance feedback are listed below. (As indicated before, each query was ran three times with three different cluster methods):

Original Query	**Relevant results fetched after expansion**	**Most relevant expanded query**	**Cluster method used**
ronaldo records	5	ronaldo records worldcup football	metric
fifa	17	fifa worldcup soccer	metric, association, scalar
football stadium	8	football stadium worldcup	metric
messi	6	messi worldcup score	association, metric
beach soccer	12	beach soccer football score	scalar
Barcelona FC	7	Barcelona FC score athelete	association
Manchester united	6	manchester united score football	association
UEFA champions	5	UEFA champions score soccer	scalar
olympics	10	olympics games football	association
commonwealth	4	commonwealth soccer games	association

Based on the foregoing findings, the association cluster approach was chosen as the optimum method for implementing pseudo relevance feedback.
Some issues to consider:

1. We can't do well because scalar clustering takes too long to run.
2. In certain circumstances, metric clustering took a long time.

Query 1: fifa
local document set(ids):
491d54c4-885c-11ea-bc55-0242ac130003
491d571c-885c-11ea-bc55-0242ac130003
491d57e4-885c-11ea-bc55-0242ac130003
491d58a2-885c-11ea-bc55-0242ac130003
491d5ab4-885c-11ea-bc55-0242ac130003
491d5b90-885c-11ea-bc55-0242ac130003
491d5c58-885c-11ea-bc55-0242ac130003
491d5d16-885c-11ea-bc55-0242ac130003
491d5f00-885c-11ea-bc55-0242ac130003
491d5fd2-885c-11ea-bc55-0242ac130003
491d6090-885c-11ea-bc55-0242ac130003
491d614e-885c-11ea-bc55-0242ac130003
491d620c-885c-11ea-bc55-0242ac130003
491d62ca-885c-11ea-bc55-0242ac130003
491d6392-885c-11ea-bc55-0242ac130003
491d65cc-885c-11ea-bc55-0242ac130003
491d66a8-885c-11ea-bc55-0242ac130003
491d6766-885c-11ea-bc55-0242ac130003
491d682e-885c-11ea-bc55-0242ac130003
491d68ec-885c-11ea-bc55-0242ac130003
491d69b4-885c-11ea-bc55-0242ac130003
491d6a72-885c-11ea-bc55-0242ac130003
491d6b30-885c-11ea-bc55-0242ac130003
491d6d6a-885c-11ea-bc55-0242ac130003
491d6e32-885c-11ea-bc55-0242ac130003
491d6ef0-885c-11ea-bc55-0242ac130003
491d6fae-885c-11ea-bc55-0242ac130003
491d7076-885c-11ea-bc55-0242ac130003

491d71ac-885c-11ea-bc55-0242ac130003
491d7288-885c-11ea-bc55-0242ac130003

fifa_worldcup_local_doc.txt

local vocabulary set: local_vocabulary_set1.txt
local stem set: local_stem_set1.txt

Query 2: olympics
local document set(ids):
987d15ba-885e-11ea-bc55-0242ac130003
987d193e-885e-11ea-bc55-0242ac130003
987d1b0a-885e-11ea-bc55-0242ac130003
987d1c72-885e-11ea-bc55-0242ac130003
987d1db2-885e-11ea-bc55-0242ac130003
987d1eca-885e-11ea-bc55-0242ac130003
987d205a-885e-11ea-bc55-0242ac130003
987d219a-885e-11ea-bc55-0242ac130003
987d229e-885e-11ea-bc55-0242ac130003
987d23fc-885e-11ea-bc55-0242ac130003
987d253c-885e-11ea-bc55-0242ac130003
987d264a-885e-11ea-bc55-0242ac130003
987d27d0-885e-11ea-bc55-0242ac130003
987d2906-885e-11ea-bc55-0242ac130003
987d2a1e-885e-11ea-bc55-0242ac130003
987d2b7c-885e-11ea-bc55-0242ac130003
987d2cc6-885e-11ea-bc55-0242ac130003
987d2e60-885e-11ea-bc55-0242ac130003
987d2faa-885e-11ea-bc55-0242ac130003
987d30fe-885e-11ea-bc55-0242ac130003
987d3252-885e-11ea-bc55-0242ac130003
987d3392-885e-11ea-bc55-0242ac130003

987d3518-885e-11ea-bc55-0242ac130003
987d36e4-885e-11ea-bc55-0242ac130003

987d38b0-885e-11ea-bc55-0242ac130003
987d39fa-885e-11ea-bc55-0242ac130003
987d3b3a-885e-11ea-bc55-0242ac130003
987d3c98-885e-11ea-bc55-0242ac130003
987d4350-885e-11ea-bc55-0242ac130003
987dac96-885e-11ea-bc55-0242ac130003

olympics_football_games_local_doc.txt

local vocabulary set: local_vocabulary_set2.txt
local stem set: local_stem_set2.txt

Query 3: ronaldo records
local documents (ids) :
f7300812-8861-11ea-bc55-0242ac130003
f7300a10-8861-11ea-bc55-0242ac130003
f7300b0a-8861-11ea-bc55-0242ac130003
f730115e-8861-11ea-bc55-0242ac130003
f73014e2-8861-11ea-bc55-0242ac130003
f730160e-8861-11ea-bc55-0242ac130003
f73017d0-8861-11ea-bc55-0242ac130003
f7302130-8861-11ea-bc55-0242ac130003
f730241e-8861-11ea-bc55-0242ac130003
f730250e-8861-11ea-bc55-0242ac130003
f73025e0-8861-11ea-bc55-0242ac130003
f730269e-8861-11ea-bc55-0242ac130003
f730275c-8861-11ea-bc55-0242ac130003
f730281a-8861-11ea-bc55-0242ac130003
f73029b4-8861-11ea-bc55-0242ac130003

f7302a90-8861-11ea-bc55-0242ac130003
f7302b62-8861-11ea-bc55-0242ac130003
f7302c20-8861-11ea-bc55-0242ac130003
f7302e78-8861-11ea-bc55-0242ac130003
f7302f54-8861-11ea-bc55-0242ac130003
f730301c-8861-11ea-bc55-0242ac130003
f73030e4-8861-11ea-bc55-0242ac130003
f73031ac-8861-11ea-bc55-0242ac130003
f730326a-8861-11ea-bc55-0242ac130003
f7303328-8861-11ea-bc55-0242ac130003
f730362a-8861-11ea-bc55-0242ac130003
f7303710-8861-11ea-bc55-0242ac130003
f73037ce-8861-11ea-bc55-0242ac130003
f7303896-8861-11ea-bc55-0242ac130003
f730395e-8861-11ea-bc55-0242ac130003

ronaldo_scores_local_doc.txt

local vocabulary set: local_vocabulary_set3.txt

local stem set: local_stem_set3.txt

The table above shows that when the association cluster approach is employed, the number of relevant pages obtained increases while the time spent on association cluster decreases. As a result, when it comes to performance, we can state that association cluster is the best option. It also corrects any spelling errors. The correlations for the queries are shown by the number of relevant pages collected from the table. For each initial query, it also displays the enlarged query.

Collaboration with UI and relevance model:

- We get the original question and the cluster method type to utilize from the API request.
- The relevant model gives the query expansion python code with the top 50

documents based on these parameters. These findings are then utilized as local documents to create local stem sets and vocabulary.

- The cluster method from association, metric, and scalar is called based on the query parameter to return the enlarged query to the relevance model.
- After then, the relevance model goes to solar in order to get results for the extended query.
- These findings are subsequently sent to the front end, where they are presented.

Query selection for demonstration:

We have selected “olympics’ for demonstration. It is tested for association cluster method.

Discussions (Kanamata Reddy, Vishnu Vardhan Reddy, vxk210042, Kotra, Sai Charan, sxk210083, Manthri, Satya Sai Bharadwaj, sxm210073, Miriyala Sai Chethan Reddy, sxm200225, Zalawadia Moxaben Bhupatbhai, mxz210014):

We learned many different aspects required to create a simple and functional search engine after completing this project. Using many approaches we learnt in class, we were able to develop a search engine that could crawl webpages and deliver relatively relevant results. The experiment taught us a lot about how to build a search engine and demonstrated that we have barely scratched the surface of information retrieval. When we compared the results, we worked so hard to get with the work that goes into creating and maintaining popular an up-to-date search engines like Google and Bing, we realized that our efforts were pale in comparison to what it takes to create and maintain these popular search engines.

Conclusion (Kanamata Reddy, Vishnu Vardhan Reddy, vxk210042, Kotra, Sai Charan, sxk210083, Manthri, Satya Sai Bharadwaj, sxm210073, Miriyala Sai Chethan Reddy, sxm200225, Zalawadia Moxaben Bhupatbhai, mxz210014):

Finally, for this project, we developed a search engine that focuses on soccer-related information, utilizing the skills we obtained during the course and various free source applications. The five of us divided the jobs necessary to develop the search engine and worked to integrate our separate components into a solution that can deliver somewhat accurate soccer information. We were able to overcome several obstacles along the road and construct a strong search engine. This project allowed us to put our classroom learning into practice and improved our comprehension of theoretical issues.