The University of Texas at Dallas
CS 6322

Information Retrieval
Spring 2023

Class Project Report

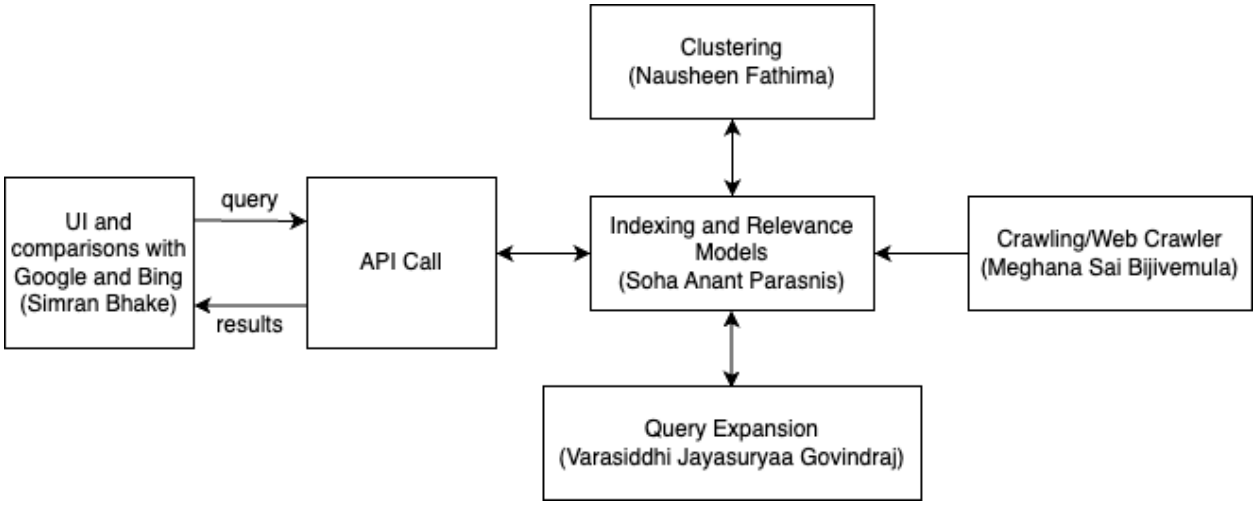
Project Title: Search Engine for Lakes
Team 10

Meghana Sai Bijivemula, mxb210011@utdallas.edu
Soha Anant Parasnis, sxp200044@utdallas.edu
Simran Bhake, sxb190165@utdallas.edu
Nausheen Fathima, NXF200000@utdallas.edu
Varasiddhi Jayasuryaa Govindraj, VXG190049@utdallas.edu

1. The Problem (5 points): Generate a search engine for Lakes

The focus of the search engine I have created, is **Lakes.**

<u>Architecture of the Search Engine</u>



Please find below the list of students in our team and which student was responsible for which part of the architecture,

Crawling - Meghana Sai Bijivemula
Indexing and relevance - Soha Anant Parasnis
User interface and comparisons with Google and Bing - Simran Bhake
Clustering - Nausheen Fathima
Query expansion and relevance feedback - Varasiddhi Jayasuryaa Govindraj

None of the team members had created a search engine before. This was a unique experience for all of us.

One of our biggest learnings was understanding how a search engine works and how our theoretical learnings can be implemented to develop it. Working with more than 100,000+ web pages was a challenge in multiple ways - right from deciding the proper seed URLs to use to get that amount at the end of crawling, to figuring out how to work on the huge amount of crawled data for the further steps in the development of this search engine.

One of the biggest challenges was integrating the results of the various tools we used in each step. It was especially difficult to brainstorm and settle on the relevance model algorithms while making sure they were effective for a variety of queries and options.

We learned a lot during the process of developing it. All in all, it was a fun yet challenging project.

Meghana Sai Bijivemula, mxb210011

2. Crawling (45 points): Meghana Sai Bijivemula, mxb210011

Crawling

Crawling developed by Meghana Sai Bijivemula

When I started researching about the crawling tools, Scrapy and Apache Nutch were the top tools that I found on the web. I did a proof of Concept with Scrapy and Apache Nutch and that is when I found out that Nutch has built-in support for page analysis for language identification, duplicate detection, and content analysis but Scrapy does not. Additionally, Nutch can handle dynamic web pages, but Scrapy is made for static web pages. So, I chose Apache Nutch over Scrapy.

I gathered 200 seed URLs which contain hyperlinks of websites related to lakes or topics associated with lakes like activities, ecosystems, pollution. I started with Wikipedia websites which lists lakes by country and then also covered continent wise using Lakepedia website. Later, I moved on to Wikipedia pages related to Artificial Lakes and list of artificial lakes by country. Additionally, I added wiki websites which cover types of lakes, pollution in lakes, prehistoric lakes, shipwrecks, and film sets on lakes etcetera. I wanted to cover the science behind lakes like origin and ecosystems, for these I used websites like education.nationalgeography, Britannica, and waterencyclopedia. Furthermore, I added government websites like uslakes, nps, Greatlakescanada which talk about lakes in state wise and lakes in national parks along with it, Britannica pages which cover the science behind lakes, ecosystem in lakes. Added many more pages from different domains which cover information on lakes in specific countries or largest lakes in terms of area and volume, deepest lakes and also best lakes based on scenery. I also used url filter to remove domains like Instagram, twitter, YouTube, and amazon to avoid crawling a lot of irrelevant data. With the 200 seed URLs, I was able to crawl **138952 webpages** through 10 iterations and deleted 23716 duplicates as a result Soha indexed 115236 webpages.

Additionally, Nutch has a built-in feature to delete duplicates during crawling and this feature is known as the duplicate removal filter and it is responsible for identifying and eliminating duplicate URLs from the crawl frontier. When Nutch crawls a webpage, it extracts all the links present on the page and adds them to the crawl frontier. However, some of these links may be duplicates of URLs that have already been crawled or are in the process of being crawled. The duplicate removal filter ensures that such duplicate URLs are not added to the crawl frontier again so that the use of resources is done

efficiently and prevent the web crawler from getting stuck in infinite loops caused by repeatedly crawling the same URLs.

We found that we were getting a lot of irrelevant data after certain iterations when we crawled using the first set of links. So, we decided to add topic that are associated with lakes like boating, fishing, kayaking, lake view properties, overfishing and its effects and also lake conservation organizations. Adding links which covered the above topics helped us crawl 138K websites.

Here the crawling was done using Apache Nutch 1.19. Crawling was done through Nutch's crawl automation command shown below:

crawl -s /Users/hacker/Documents/IR/Project/crawling/apache-nutch-1.19/runtime/local/bin/urls /Users/hacker/Documents/IR/Project/crawling/apache-nutch-1.19/runtime/local/crawl 10

When we run the above command, it goes to the seedlist and performs bootstrapping from an initial seed list and creates a database with unfetched urls. Then it generates a fetch list for all of the pages due to be fetched. The fetch list is placed in a newly created segment directory. Every URL that Nutch is working with has information about it in the crawldb, including whether it was fetched and, if so, when. The list of known links to each URL, together with their source URL and anchor text, will be found in the linkdb database.

Later, it fetches and parses the contents of webpages in the segment. When this is complete, we update the database with the results of the fetch. Now the database contains both updated entries for all initial pages as well as new entries that correspond to newly discovered pages linked from the initial set. Each segment consists of a group of URLs that are fetched collectively. Directories called segments contain the following subdirectories: a *crawl_generate*, a list of URLs to be fetched is named, each URL's current fetching state is listed in *crawl_fetch,* raw content collected from each URL is contained in *content,* parsed text of each URL is contained in a parse_text, each URL's outlinks and metadata are processed into a *parse_data,* outlink URLs are contained in *crawl_parse* and are used to update the crawldb.

When Indexing with Apache Solr, the indexing is done on the segments generated after crawling with Apache Nutch. Once, Soha setup Solr, which is compatible with Nutch, Soha and I pointed Nutch to Solr. Once the crawling is done and the segments are

created, then I will run the command to invert all of the links. After this the segments are ready to be indexed through the indexing command. The Segments and its subdirectories generated after the above crawl command will be used for indexing in solr and also running vector relevance models like Link rank. This is how I provide the required segments to Soha for Indexing.

Below are **seed urls** used for crawling:

https://www.lakepedia.com/continent/africa-lakes.html

https://www.lakepedia.com/continent/asia-lakes.html

https://www.lakepedia.com/continent/australia-and-oceania-lakes.html

https://www.lakepedia.com/continent/europe-lakes.html

https://www.lakepedia.com/continent/north-america-lakes.html

https://www.lakepedia.com/continent/south-america-lakes.html

https://lakeaccess.org/

https://www.lakechamplaincommittee.org/learn/lake-ecology/

https://www.visitmammoth.com/things-to-do/lakes

https://en.wikipedia.org/wiki/Lake

https://en.wikipedia.org/wiki/Lists_of_lakes

https://en.wikipedia.org/wiki/List_of_lakes_by_area

https://en.wikipedia.org/wiki/Great_Lakes

https://en.wikipedia.org/wiki/List_of_lakes_of_the_United_States

https://en.wikipedia.org/wiki/List_of_lakes_of_Japan

https://en.wikipedia.org/wiki/List_of_lakes_of_China

https://en.wikipedia.org/wiki/List_of_lakes_of_Russia

https://en.wikipedia.org/wiki/List_of_lakes_of_India

Meghana Sai Bijivemula, mxb210011

https://en.wikipedia.org/wiki/List_of_lakes_of_Australia

https://en.wikipedia.org/wiki/List_of_lakes_of_New_Zealand

https://en.wikipedia.org/wiki/List_of_lakes_of_Argentina

https://en.wikipedia.org/wiki/List_of_lakes_of_Brazil

https://en.wikipedia.org/wiki/List_of_lakes_of_Peru

https://en.wikipedia.org/wiki/List_of_lakes_of_Chile

https://en.wikipedia.org/wiki/List_of_lakes_of_Bolivia

https://en.wikipedia.org/wiki/List_of_lakes_of_Uruguay

https://en.wikipedia.org/wiki/List_of_lakes_of_South_Africa

https://en.wikipedia.org/wiki/List_of_lakes_of_Canada

https://en.wikipedia.org/wiki/Underground_lake

https://en.wikipedia.org/wiki/Lake_ecosystem

https://www.nalms.org/home/basics-of-lake-management/

https://education.nationalgeographic.org/resource/lake/

https://www.britannica.com/science/lake/Basins-formed-by-tectonism-volcanism-and-landslides

https://what-when-how.com/water-science/lakes-water-science/
http://www.waterencyclopedia.com/Hy-La/Lake-Formation.html

https://a-z-animals.com/blog/how-were-the-great-lakes-formed-and-how-long-ago/

https://dec.vermont.gov/watershed/lakes-ponds/aboutlakes

https://www.ecoshape.org/en/lakes-environment/

https://www.climate-policy-watcher.org/lake-ecosystems/lake-ecology.html

https://www.uslakes.info/

Meghana Sai Bijivemula, mxb210011

https://www.glc.org/lakes/

https://search.epa.gov/epasearch/?querytext=lakes&areaname=&areacontacts=&areasearchurl=&typeofsearch=epa&result_template=#/

https://www.britannica.com/search?query=lakes&page=5

https://en.wikipedia.org/wiki/Category:Lists_of_lakes_by_country

https://whc.unesco.org/en/list/?search=lakes&order=country

https://en.wikipedia.org/wiki/List_of_lakes_of_Italy

https://en.wikipedia.org/wiki/List_of_lakes_of_Greece

https://en.wikipedia.org/wiki/List_of_lakes_of_Indonesia

https://en.wikipedia.org/wiki/List_of_lakes_of_Sweden

https://www.loc.gov/collections/world-digital-library/?q=lakes

https://www.nrdc.org/search?search_api_fulltext=lakes

https://fws.gov/search?$keywords=%22lakes%22

https://www.nps.gov/subjects/lakes/index.htm

https://www.britannica.com/science/lake

https://www.livescience.com/search?searchTerm=lakes

https://www.worldatlas.com/search?q=lakes

https://www.usgs.gov/special-topics/water-science-school/science/lakes-and-reservoirs

https://www.nature.com/search?q=lakes

https://www.watereducation.org/find/results/lakes

https://www.worldlakes.org/

https://www.nature.org/en-us/search/?q=lakes

https://www.oxfordlearnersdictionaries.com/definition/english/lake

Meghana Sai Bijivemula, mxb210011

https://www.sciencedirect.com/search?qs=lakes

https://www.worldwildlife.org/search?cx=003443374396369277624%3Av3nraqhmeyk&ie=UTF-8&x=lakes#gsc.tab=0&gsc.q=lakes&gsc.page=1

https://lakelubbers.com/

https://www.climate.gov/search?query=lakes

https://www.treehugger.com/search?q=lakes

https://www.visittheusa.com/search?keys=lakes

https://www.who.int/home/search?indexCatalogue=genericsearchindex1&searchQuery=lakes&wordsMode=AnyWord

https://www.nationalparks.org/search?query=lakes

https://www.lakegenevawi.com/

https://www.nationalforests.org/our-forests/your-national-forests-magazine/great-lakes-great-forests

https://www.visitlakegeneva.com/

https://lakehub.com/

https://www.lakescientist.com/

https://theodora.com/encyclopedia/l/lake.html

https://www.thecanadianencyclopedia.ca/en/article/great-lakes

https://education.nationalgeographic.org/resource/lake/

https://en.wikipedia.org/wiki/Oxbow_lake

https://en.wikipedia.org/wiki/Dimictic_lake

https://guides.nynhp.org/search-results/?n=lakes

https://www.nationalgeographic.org/projects/okavango/expeditions/source-lakes-science/

Meghana Sai Bijivemula, mxb210011

https://www.usgs.gov/special-topics/water-science-school/science/lakes-and-reservoirs

https://tpwd.texas.gov/fishboat/fish/recreational/lakes/

https://www.glc.org/lakes/

https://www.pinetoplakesideaz.gov/224/Lakes

https://en.wikipedia.org/wiki/List_of_lakes_of_France

https://en.wikipedia.org/wiki/List_of_lakes_of_Latvia

https://en.wikipedia.org/wiki/List_of_lakes_of_Madagascar

https://en.wikipedia.org/wiki/List_of_lakes_of_Hungary

https://en.wikipedia.org/wiki/List_of_lakes_of_Belgium

https://en.wikipedia.org/wiki/List_of_lakes_of_Switzerland

https://en.wikipedia.org/wiki/List_of_lakes_of_Vietnam

https://en.wikipedia.org/wiki/Portal:Lakes

https://en.wikipedia.org/wiki/Monomictic_lake

https://en.wikipedia.org/wiki/Meromictic_lake

https://en.wikipedia.org/wiki/Amictic_lake

https://en.wikipedia.org/wiki/List_of_lakes_of_Armenia

https://en.wikipedia.org/wiki/List_of_lakes_of_Himachal_Pradesh

https://en.wikipedia.org/wiki/Category:Shipwrecks_in_lakes

https://en.wikipedia.org/wiki/Category:Films_set_on_lakes

https://lakenearme.com/

https://www.ducksters.com/searchducksters.php?q=lakes

https://dec.vermont.gov/search/node?keys=lakes

Meghana Sai Bijivemula, mxb210011

https://www.epa.gov/greatlakes

https://www.dec.ny.gov/searchresult.html#stq=lakes&stp=1 http://www.worldlakes.org/

https://en.wikipedia.org/wiki/Category:Artificial_lakes_of_the_United_States

https://en.wikipedia.org/wiki/Controlled_lake

https://en.wikipedia.org/wiki/Reservoir

https://en.wikipedia.org/wiki/Category:Artificial_lakes_by_country

https://en.wikipedia.org/wiki/List_of_reservoirs_by_volume

https://en.wikipedia.org/wiki/Category:Artificial_lakes

https://en.wikipedia.org/wiki/List_of_reservoirs_by_surface_area

https://www.usgs.gov/special-topics/water-science-school/science/lakes-and-reservoirs

https://www.lakesonline.com/
http://www.worldlakes.org/lakeprofiles.asp?anchor=deepest
http://www.worldlakes.org/lakeprofiles.asp?anchor=amazing
http://www.worldlakes.org/lakeprofiles.asp?anchor=ancient
http://www.worldlakes.org/lakeprofiles.asp?anchor=volume
http://www.worldlakes.org/lakeprofiles.asp?anchor=area

https://www.cs.mcgill.ca/~rwest/wikispeedia/wpcd/wp/l/List_of_world%2527s_largest_lakes.htm

https://en.wikipedia.org/wiki/Ancient_lake

https://en.wikipedia.org/wiki/List_of_prehistoric_lakes

https://www.webuildvalue.com/en/infrastructure-news/artificial%20lakes.html

https://www.privatewaterfishing.com/properties

https://www.bestfishinginamerica.com/texas-bass-fishing.html

https://fishbrain.com/fishing-waters/popular

https://outdoorsman.guide/texas-fishing-lakes/

Meghana Sai Bijivemula, mxb210011

https://outdoorsman.guide/americas-best-fishing-locations/

https://www.aa-fishing.com/tx/texas-fishing-lakes.html

https://www.aa-fishing.com/fishing-lakes.html

https://fishedthat.com/

https://ksoutdoors.com/Fishing/Fishing-Regulations/State-Fishing-Lakes

https://dnr.maryland.gov/fisheries/pages/hotspots/index.aspx

https://www.iowadnr.gov/Fishing/Where-to-Fish/Lakes-Ponds-Reservoirs

https://outdoornebraska.gov/fish/where-to-fish/family-friendly-lakes/

https://portal.ct.gov/DEEP/Fishing/CT-Fishing

https://www.boatsetter.com/boating-resources/best-lakes-in-texas-for-boating

https://www.extraspace.com/blog/outdoor-recreation/best-boating-lakes-in-the-us/

https://www.boat-ed.com/

https://www.funlake.com/boating

https://lakelandboating.com/

https://www.michigan.gov/dnr/

https://www.privatecommunities.com/lakefront-homes-communities.htm
http://www.myalamedaparade.com/10-fun-things-lake-summer/

https://www.lifeinminnesota.com/lake-activities/

https://www.fortworth.com/blog/post/fort-worth-lakes-recreational-activities/

https://www.touropia.com/most-beautiful-lakes-in-the-world/

https://www.busytourist.com/things-to-do-in-big-bear-lake-ca/

https://getbybus.com/en/blog/top-lakes-in-europe/

https://www.strel-swimming.com/best-lakes-swimming-europe/

Meghana Sai Bijivemula, mxb210011

https://www.ridestore.com/mag/best-lakes-in-europe/

https://manntravel.co.uk/best-swimming-adventures-across-europe/

https://www.europeanbestdestinations.com/best-of-europe/best-natural-pools-in-europe/

https://trulyexperiences.com/blog/most-gorgeous-lakes-in-europe/

https://malawiplus.com/watersports/

https://www.ugandasafariexperts.com/beta/lake-bunyonyi-activities/

https://www.africangreatlakesinform.org/page/african-great-lakes

https://en.wikipedia.org/wiki/African_Lakes_Corporation

https://www.society19.com/things-to-do-at-a-lake/

https://www.des.nh.gov/sites/g/files/ehbemt341/files/documents/2020-01/bb-49.pdf

https://en.wikipedia.org/wiki/Living_Lakes_Network

https://en.wikipedia.org/wiki/Great_Lakes_Commission

https://a-z-animals.com/blog/lake-vs-pond-the-3-main-differences-explained/

https://nhlakes.org/state-of-nh-lakes/

https://www.solitudelakemanagement.com/blog/how-do-lakes-differ-from-ponds/
http://www.differencebetween.net/science/nature/difference-between-lake-and-lagoon/

https://www.nature.org/en-us/about-us/where-we-work/priority-landscapes/great-lakes/stories-in-the-great-lakes/great-lakes-fisheries/

https://iiseagrant.org/fishing-may-lead-to-rapid-changes-in-great-lakes-fish/

https://www.greatlakesnow.org/2019/11/great-lakes-commercial-fishing-history/

https://wildewoodonlakesavant.com/is-lake-fishing-harmful-to-the-environment/

https://www.motherjones.com/environment/2015/01/climate-change-lake-victoria-overfishing/

https://fishingbooker.com/blog/history-of-fishing-on-the-great-lakes-part-2/

Meghana Sai Bijivemula, mxb210011

https://greatlakes.guide/ideas/the-benefits-of-lake-swimming

https://ijc.org/en/what/glwq

https://greatlakes.org/campaigns/defending-the-great-lakes-compact/
http://www.worldlakes.org/lakes.asp

https://www.globalnature.org/livinglakes

https://www.unep.org/news-and-stories/story/interlinked-threats-facing-lakes-and-why-we-need-protect-them

https://phys.org/news/2023-03-scientists-urgent-action-world-lakes.html

https://www.nature.org/en-us/about-us/where-we-work/united-states/michigan/stories-in-michigan/great-lakes-african-women-science/

https://www.ecosuperior.org/protecting-lake-superior

https://www.goldbarrealty.com/lakes/

https://documents.deq.utah.gov/water-quality/standards-technical-services/great-salt-lake-advisory-council/activities/DWQ-2019-009999.pdf

https://whc.unesco.org/en/list/?search=lakes&type=natural&order=country

https://www.circleofblue.org/2015/world/biggest-lakes-in-the-world-under-pressure-from-human-and-environmental-threats/

https://www.fllt.org/our-projects-to-save-lakes-streams-and-drinking-water/

https://en.wikipedia.org/wiki/Category:Top-importance_Lakes_articles

https://en.wikipedia.org/wiki/Lake_Malawi#Fishing

https://tpwd.texas.gov/landwater/water/aquaticspecies/inland.phtml

https://a-z-animals.com/blog/the-12-deadliest-lakes-in-the-world/

https://wldb.ilec.or.jp/Search/Lakename

https://www.barlettapontoonboats.com/blog/the-top-lakes-in-texas-for-boating

Meghana Sai Bijivemula, mxb210011

https://gl.audubon.org/thenest/great-lakes-migratory-birds

https://greatlakes.org/2019/06/birding-in-the-great-lakes-region/

https://councilgreatlakesregion.org/the-great-lakes-economy-the-growth-engine-of-north-america/

https://www.britannica.com/place/East-African-lakes/People-and-economy

Soha Anant Parasnis, sxp200044

3. Indexing and relevance (45 points): Soha Anant Parasnis, sxp200044

Indexing

For indexing, we decided to use Apache Solr (version 8.11.2). Apache Solr is an open-source enterprise-search platform, written in JAVA.

As a result of crawling, we can see three major folders in the runtime/local/crawl folder: crawldb, linkdb, segments. Now, while creating an index for this crawled data in Solr, we give the path to these folders in the commands so it gets used as the input. I referred to the official Nutch Documentation and Tutorial on Confluence. I used the following command to index the crawled pages,

bin/nutch index crawl/crawldb/ -linkdb crawl/linkdb -dir crawl/segments/ -filter -normalize -deleteGone

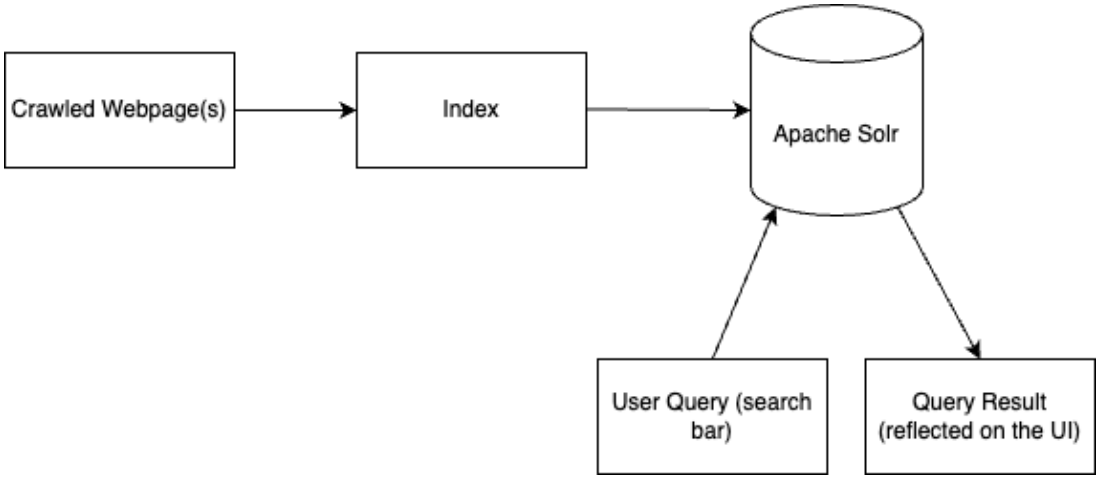
Let's discuss what paths we give for this command.

1. bin/nutch - path of the Apache Nutch under the runtime/local folder.
2. crawl/crawldb/ - path to the crawldb folder which got created after crawling. This is one of the three major folders we got as a result of crawling. It contains all the links parsed by Nutch.
3. crawl/linkdb - path to the linkdb folder which got created after crawling. This is one of the three major folders we got as a result of crawling. It contains the outgoing and incoming links/URLs for each URL.
4. crawl/segments/ - path to all the segments which got created after crawling. This is one of the three major folders we got as a result of crawling.

The crawled pages can also be indexed by using a similar command for all segments individually, the command just changes minutely for that. The bin/nutch is the path to the binary of Apache Nutch under the runtime/local folder. IndexWriter is set to "solr", and when I run this command, these plugins takes the crawled pages from the segments and passes them to Solr. The option of "-filter" is to skip documents with URL rejected by configured URL filters. The "-normalize" option normalizes URLs before indexing.The "-deleteGone" option sends deletion requests for 404s, redirects, duplicate.

Soha Anant Parasnis, sxp200044

Index Creation/Assembly Diagram



Webgraph

After indexing, I generated a webgraph of the existing segments. This can be done using the following Nutch command, as outlined in the official Nutch Command Line Options documentation,

bin/nutch webgraph -segmentDir crawl/segments/ -webgraphdb crawl/webgraphdb

This command is an alias for the WebGraph class which creates 3 databases - one for inlinks, one for outlinks and a node database that holds the number of in and outlinks to a url and the current score for the url. The webgraph is an update-able database.

Statistics

Please find below the statistics for the webgraph I have generated,

1. The number of nodes = 130765
2. The number of links = 5725370
3. The largest number of ingoing links = 10000
4. The largest number of outgoing links = 100

Connecting Webgraph Information to Index

When we create an index in Solr, we also run the LinkRank program provided by Nutch. This performs iterative link analysis with the generated webgraph. The score for the

URL which is stored in the nodes database, is generated/given by this LinkRank analysis program. The LinkRank program starts with a common score for all URLs and then creates a global score for each URL based on the number of inlinks, the scores for those links, and the number of outlinks.

This is how we connect the webgraph information to the index in Solr.

Relevance Models

Vector Space Relevance Model

Apache Solr provides an in-built tf-idf based relevance model to score the webpages. In a vector space relevance model, documents and queries are both vectors. A Vector Space Relevance Model is an algebraic model for representing text documents (and any objects) as vectors of identifiers (such as index terms). Similarity of a document vector to a query vector can be computed by taking the cosine of the angle between them. Solr uses this cosine similarity between the document and the query. This scoring model depends on tf (term frequency) and idf (inverse document frequency),

Term Frequency (tf) - The frequency with which a term appears in a document. Given a search query, the higher the term frequency, the higher the document score.

Inverse Document Frequency (idf) - The rarer a term is across all documents in the index, the higher its contribution to the score.

$$W_{d,t} = tf_{d,t} \times idf_t$$

tf-idf weighing scheme is the most common weighing scheme in vector space relevance models.

Page Rank

For this relevance model, I used the in-built Nutch Page Rank scoring. According to official Nutch documentation, the damping factor is 0.85. Please find below the steps with the nutch commands used, to update our crawldb with the Page Rank scores, so that when we query the indexed pages, we'll have the results in the descending order of their Page Ranks -

1. Now, we have our indexed pages. Let's create a webgraph using the crawled segments.

   bin/nutch webgraph -segmentDir crawl/segments/ -webgraphdb crawl/webgraphdb

2. Use nutch's linkrank command to run the Page Rank algorithm iteratively until the score converges.

   bin/nutch linkrank -webgraphdb crawl/webgraphdb/

3. Update the page rank score into the crawldb.

   bin/nutch scoreupdater -crawldb crawl/crawldb -webgraphdb crawl/webgraphdb/

4. Now, index the updated crawldb.

   bin/nutch index crawl/crawldb -linkdb crawl/linkdb -dir crawl/segments/

HITS (Hyperlink Induced Topic Search)

HITS is an algorithm used for link analysis. It is used to discover and rank webpages relevant for a particular search. The two main components of HITS are Hubs and Authority. They are used to define a recursive relationship between webpages. A node is high quality if many high quality nodes link to it; such a node or a page is called an "Authority." A Hub is a node which links to many high quality nodes. Each "node" is assigned a hub and authority score/weight.

For implementing this relevance model of HITS, I utilized the Python networkx library. First, I created a readable dump of all the inlinks of the crawldb using a nutch command. I used this readable inlinks dump to create the outlinks in my HITS program. After forming the networkx Graph using these inlinks and outlinks, I applied the HITS algorithm on it, which then gave me the Hub Scores and Authority Scores for the given data. Using the generated Authority Scores (the maximum at the top), the relevant webpages are sorted and passed on to the UI for displaying it.

Soha Anant Parasnis, sxp200044

1. The highest Hub Score was assigned to: https://lakelubbers.com/lake/slacks-pond-rhode-island-usa/#shop

   Hub Score: 0.00020986891270129002

2. The highest Authority Score was assigned to: https://lakelubbers.com/

   Authority Score: 0.07408443899339986

Topic Based Page Ranking

For topic based page ranking, the following two topics were chosen - lakes by artificial, lakes by fishing.

Lakes by artificial

1. "url" : "https://www.aa-fishing.com/aa-bass-shallow-worms.html"

   "title" : "AA Bass Fishing - Guide To Shallow Artificial Worms"

   score: 0.2775076

2. "url" : "https://www.webuildvalue.com/en/infrastructure-news/artificial%20lakes.html"

   "title" : "Artificial Lakes: What are they? - We Build Value"

   Score : 0.1500041

3. "url" : "https://lakeaccess.org/how-are-man-made-lakes-created/"

   "title" : "How are Man-Made Lakes Created? Types and Uses of Artificial Lakes - Lake Access"

   score: 0.1500041

4. "url" : "https://dnr.maryland.gov/fisheries/pages/reefs/index.aspx"

   "title" : "Artificial Reef Program", score: 0.1500041

Soha Anant Parasnis, sxp200044

Lakes by fishing

1. “url” : “https://outdoorsman.guide/block/”

   “title” : “block - The Outdoorsman Fishing Lakes, Reports & Guides”

   score: 0.3046021

2. “url” : “https://outdoorsman.guide/arizona-fishing-lakes/”

   “title” : “Arizona State Fishing Lakes & Rivers - The Outdoorsman Fishing Lakes, Reports & Guides”

   score: 0.21375585

3. “url” : “https://outdoorsman.guide/flathead-lake-fishing-guide/”

   “title” : “Flathead Lake Fishing Guide - The Outdoorsman Fishing Lakes, Reports & Guides”

   score: 0.1500041

4. “url” : “https://outdoorsman.guide/kachess-lake-fishing-guide/”

   “title” : “Kachess Lake Fishing Guide - The Outdoorsman Fishing Lakes, Reports & Guides”

   score: 0.1500041

Soha Anant Parasnis, sxp200044

Collaboration with the UI Task Student and Testing Relevance Models' Results

I collaborated with Simran, who is responsible for UI, to brainstorm queries that would help in testing the relevance models from all angles. This included choosing a good variety of content in the queries, and we tested around 20+ queries. We observed that compared to Page Rank, the HITS algorithm gave better, more relevant results.

I also made sure to request Simran to test multiple queries back-to-back to make sure that the API could handle those many requests. We also made sure that the communication between these relevance models' implementations and the UI is smooth and functional.

To judge the results of the queries, I checked the first few results and saw how relevant they were to the query entered.

I collaborated with Nausheen, who is responsible for Clustering, by providing her with the indexed pages in Solr. She takes that and performs clustering on that to improve the results we get for any particular query. We observed that clustering significantly improved the results for a query.

Simran Bhake, sxb190165

4. User interface and comparisons with Google and Bing (45 points): Simran Bhake,SXB190165

The front end is built using JavaScript, React.js and Material UI. React is a free and open-source front-end JavaScript library for building user interfaces based on components used to build dynamic single page applications.

The application interface has a search bar for entering the query and buttons for selecting options for relevance model, clustering and query expansion. After making the selections, on clicking search the search results using our search engine are fetched and displayed on the front end. The query can also be searched using Bing or Google, iframe embedding is used to display these. Thus the UI will have 3 frames, results from our search engine, results from Google and results from Bing.

For integration and to provide the results in the user interface, I make API calls. I make a GET request which has multiple parameters, depending on that it fetches the results from the relevant model. The relevance model (Page rank and Hits) is provided by Soha. The API endpoint take, query and type parameters to get the results in form of JSON. Relevance_model param takes values, Page rank or hits. Clustering_model param takes values, flat clustering, hierarchical clustering single and hierarchical clustering average. Query_expansion param takes values, association, metric and scalar.

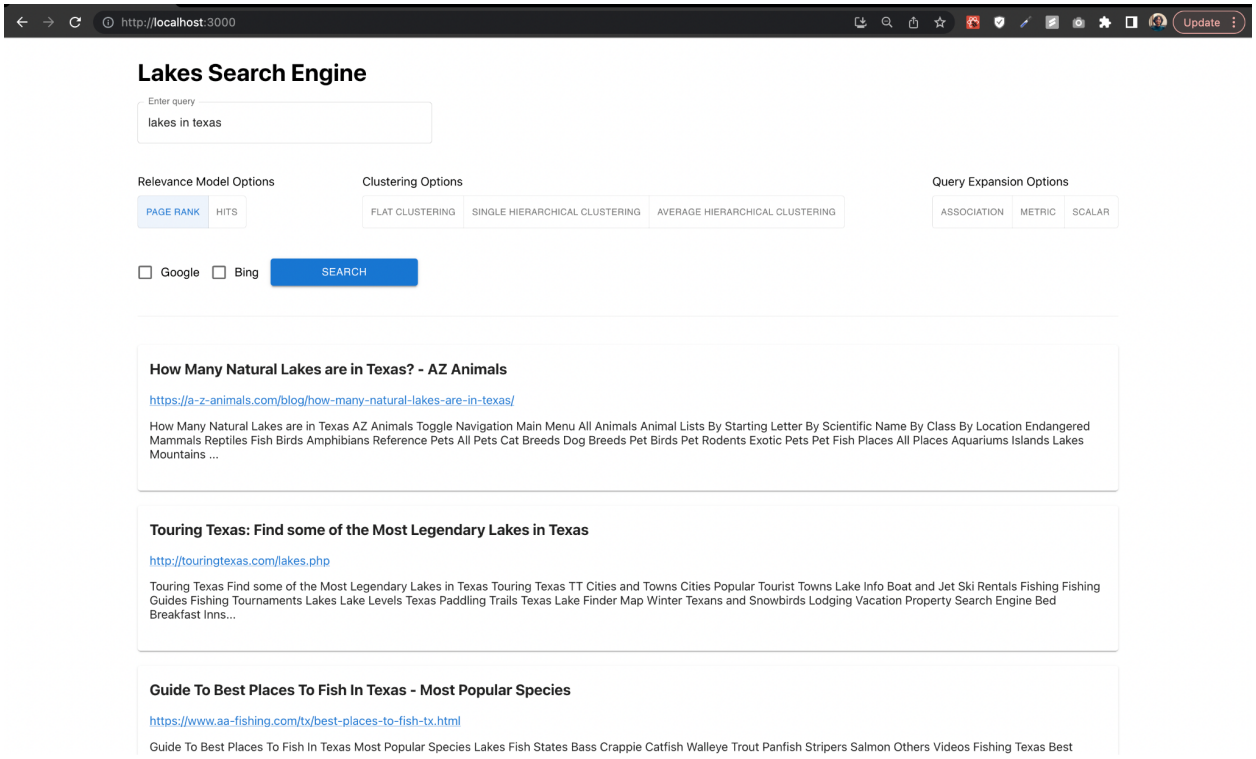
Together, Soha and I tried a few queries to test the communication between the User Interface and API, I also tested more than 20 queries to make sure that the API requests could handle a lot of requests and to make sure that all of the various parameters were being sent to the API correctly, by myself I tested 20 queries. The clustering part is done by Nausheen and for every query entered Soha's relevance model will give top 50 results to Nausheen, then her clustering program rearranges the results according to which clustering type is selected and such that the results would begin with the same results as before and subsequent results would be ordered based on the clusters they belong to.

When compared to Google and Bing search engines, for certain queries our search engine can do same or even better, but in general it is not as good as Google or Bing, as we don't have the access to a huge amount of web pages and real time updating of results based on changing traffic to websites based on query.

Simran Bhake, sxb190165

For the demo presentation, we have selected the queries that show accurate and relevant results and also improve after clustering and query expansion is added.

Search results for query, “lakes in texas”:



Simran Bhake, sxb190165

Google search results:

Lakes Search Engine

Enter query

lakes in texas

Relevance Model Options

PAGE RANK | HITS

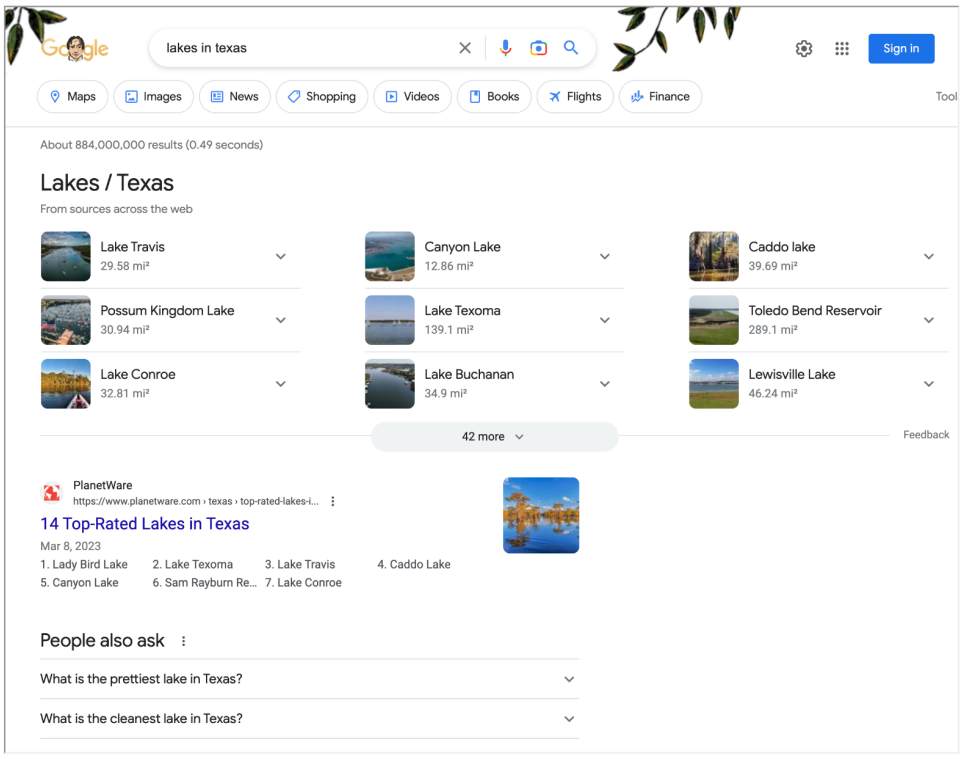
Clustering Options

FLAT CLUSTERING | SINGLE HIERARCHICAL CLUSTERING | AVERAGE HIERARCHICAL CLUSTERING

Query Expansion Options

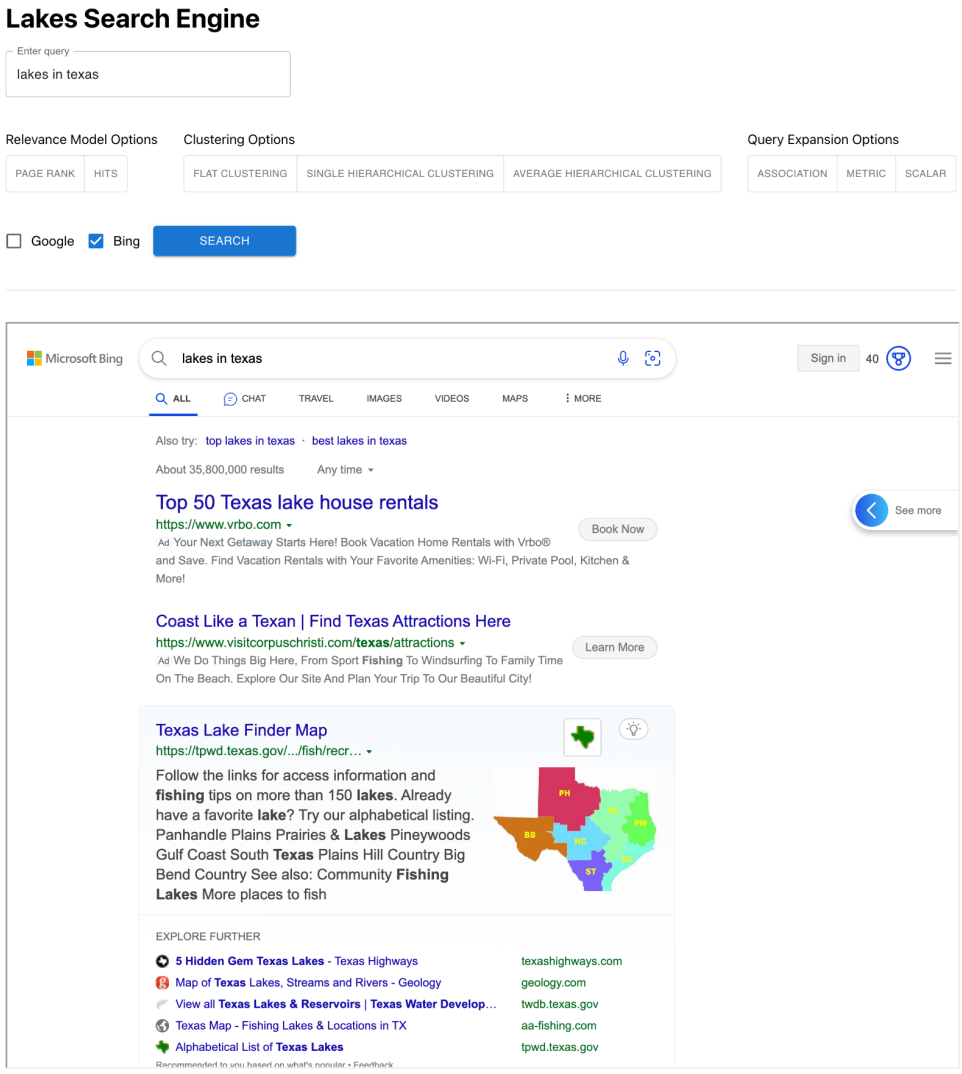
ASSOCIATION | METRIC | SCALAR

Google Bing SEARCH



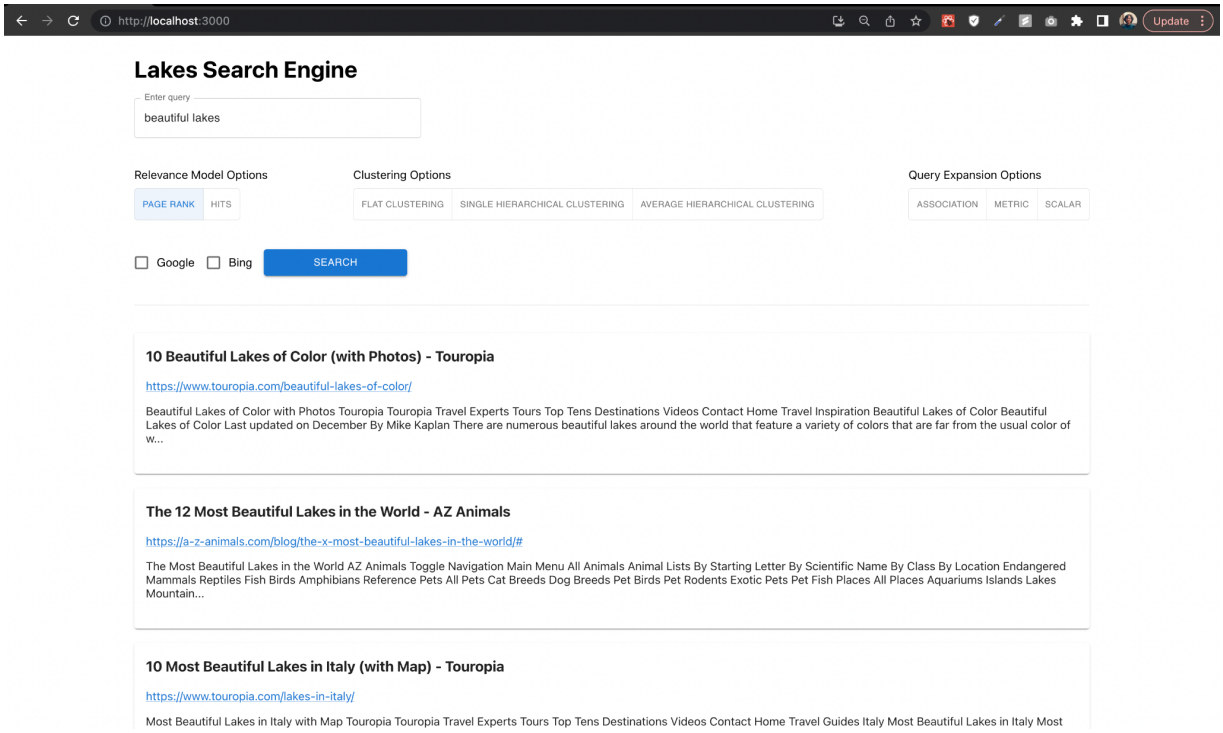
Simran Bhake, sxb190165

Bing search results:



Simran Bhake, sxb190165

Search results for query, “beautiful lakes”:



Simran Bhake, sxb190165

Google search results:

Lakes Search Engine

Enter query

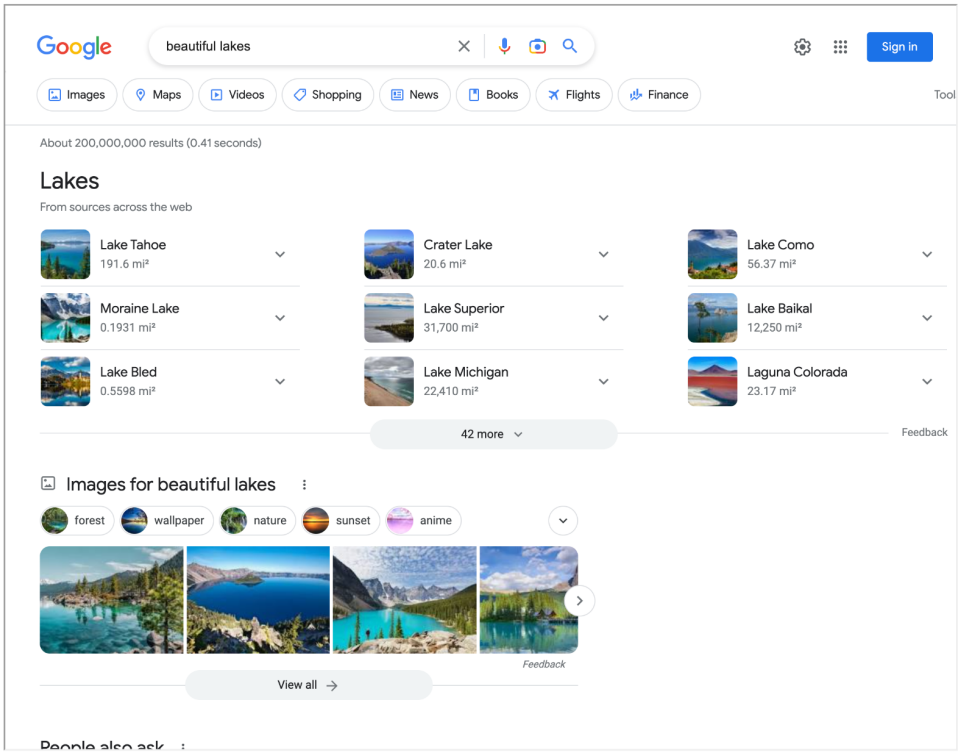
beautiful lakes

Relevance Model Options: PAGE RANK | HITS

Clustering Options: FLAT CLUSTERING | SINGLE HIERARCHICAL CLUSTERING | AVERAGE HIERARCHICAL CLUSTERING

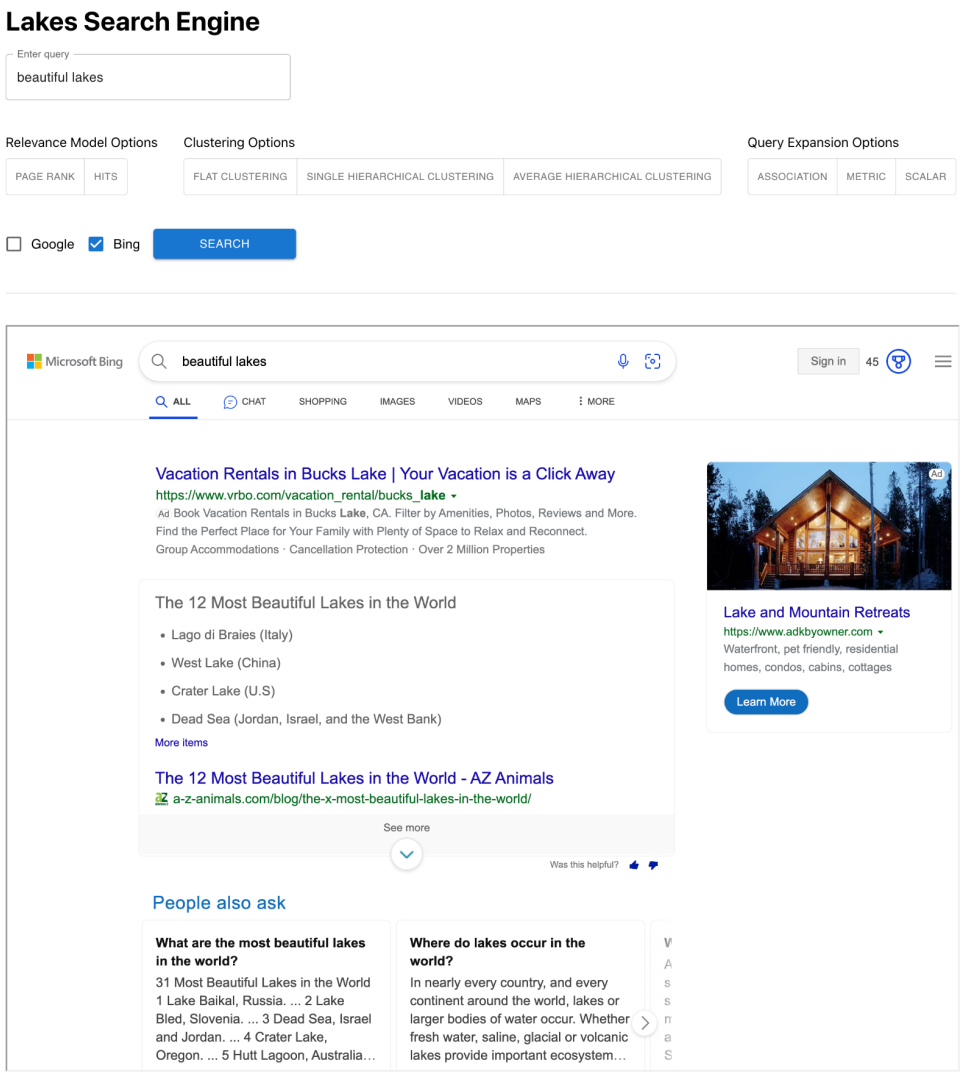
Query Expansion Options: ASSOCIATION | METRIC | SCALAR

Google Bing SEARCH



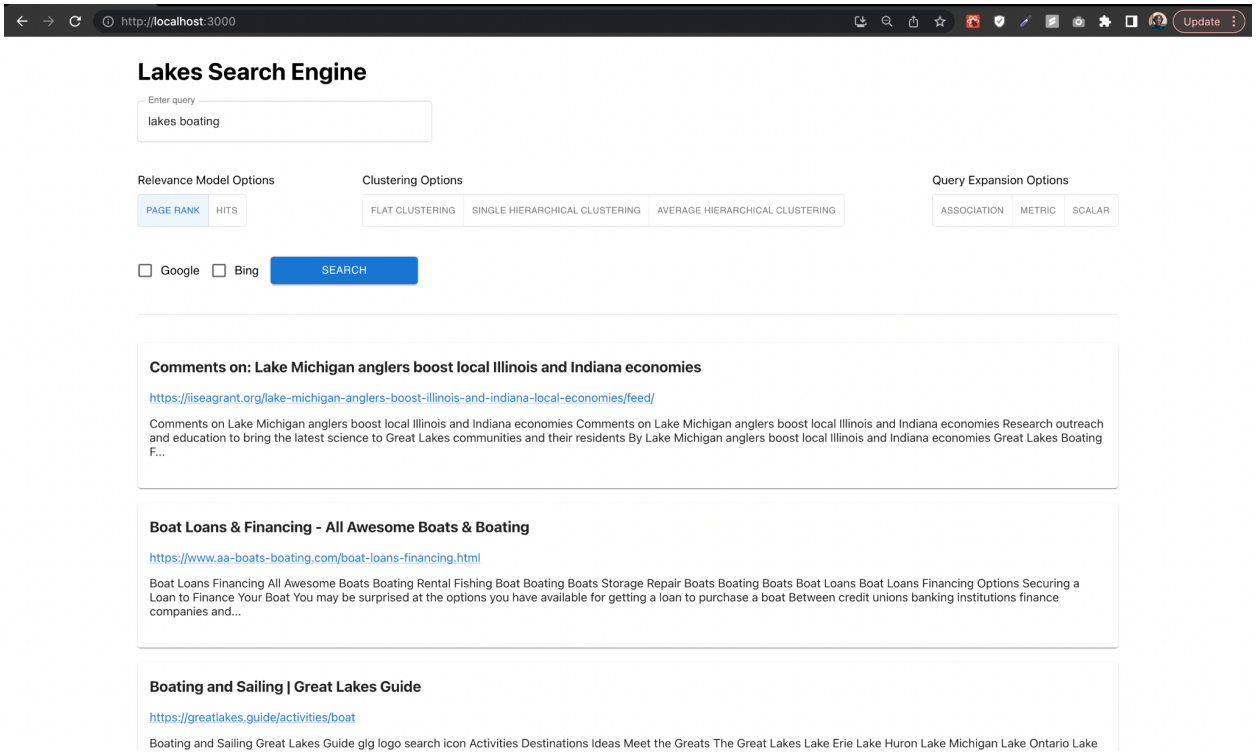
Simran Bhake, sxb190165

Bing search results:



Simran Bhake, sxb190165

Search results for query, “lakes boating”:



Simran Bhake, sxb190165

Google search results:

Lakes Search Engine

Enter query

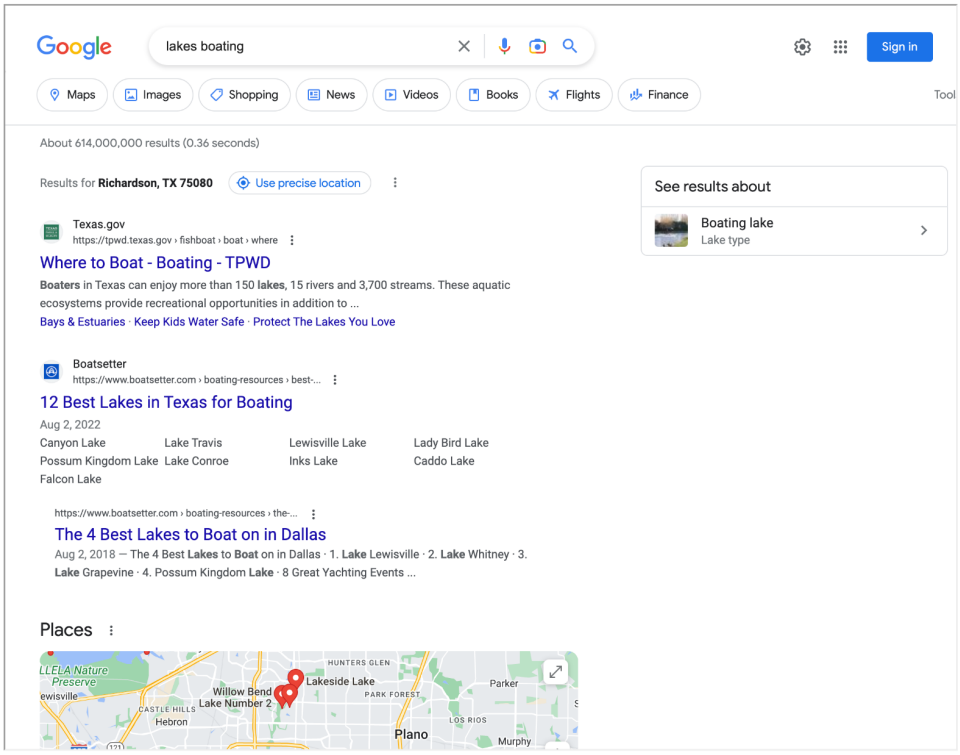
lakes boating

Relevance Model Options: PAGE RANK | HITS

Clustering Options: FLAT CLUSTERING | SINGLE HIERARCHICAL CLUSTERING | AVERAGE HIERARCHICAL CLUSTERING

Query Expansion Options: ASSOCIATION | METRIC | SCALAR

Google Bing SEARCH



Simran Bhake, sxb190165

Bing search results:

Lakes Search Engine

Enter query

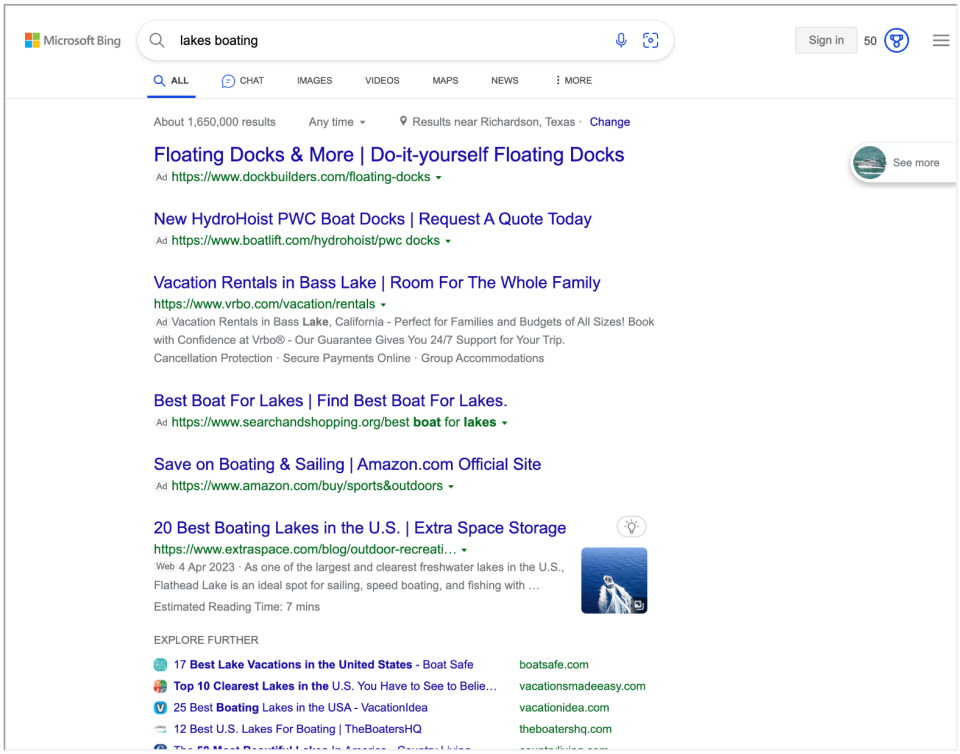
lakes boating

Relevance Model Options: PAGE RANK | HITS

Clustering Options: FLAT CLUSTERING | SINGLE HIERARCHICAL CLUSTERING | AVERAGE HIERARCHICAL CLUSTERING

Query Expansion Options: ASSOCIATION | METRIC | SCALAR

☐ Google ☑ Bing SEARCH



Nausheen Fathima, nxf200000

5. Clustering (45 points): Nausheen Fathima,NXF200000

Flat Clustering

Flat clustering was performed on the indexed data using K-means. k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the 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) (i.e. variance).

where μi is the mean (also called centroid) of points in Si

Predefined cluster selection – K value
Values of K, ranging from 5 to 25 were tested on sample dataset and relevance of documents was compared. Later we used silhouette method to determine optimal K value. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually. This measure has a range of [-1, 1]. I chose silhouette over elbow method because the silhouette score provides a more comprehensive measure of the quality of clustering results because it takes into account both the cohesion within clusters and the separation between clusters. As a result all the documents are clustered in 21 clusters

Flat Clustering Design:
All the documents crawled by Meghana Sai were indexed by Soha into Apache Nutch. I took the data in json file format and used for clustering

1. I extracted the data required from content attribute from json file
2. URL was also extracted and stored
3. I used TFIDFVectorizer for vectorization
4. K-means algorithm from scikit learn was used to apply K-means clustering algorithm with K=21

5. K-means algorithm gave the clusters of documents from which result was stored in text file in below format: URL, Cluster

Clustering results integrated with UI:

1. UI is having three buttons under clustering : Flat Clustering, Hierarchical Single Link and Hierarchical Average Link
2. When one of the button is pressed a get request is fired along with what kind of clustering is requested
3. Solr fetches the top 50 relevant results based on page ranking and HITS score and forwards the data to clustering function
4. In clustering function results are ranked in such a way that cluster wuth top ranked were clustered in same and ordered in decaresing page rank and HITS scores
5. The process repeats for the next highest ranked results
6. This process repeats till we reach threshold of 50
7. The results are then sent as response to UI

Hierarchical clustering

Agglomerative clustering algorithm with single and average link was implemented for hierarchical clustering of data that was indexed.

Hierarchical clustering (also called hierarchical cluster analysis or HCA) is a method of cluster analysis that seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two categories: Agglomerative and Divisive

Agglomerative clustering algorithm

1. Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. Agglomerative is a "bottom-up" approach: Each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
2. Results from hierarchical clustering are represented as dendrograms.
3. We have the option to use distance metric either Euclidean, Manhattan etc
4. I have used Euclidean distance metric
5. We can choose different linkages such as single, complete, average, war, centroid link, etc.
6. In this project I have just two linkages : Single and Complete

Nausheen Fathima, nxf200000

Hierarchical design:
All the documents crawled by Meghana Sai were indexed by Soha into Apache Nutch. I took the data in json file format and used for clustering

1. I extracted the data required from content attribute from json file
2. URL was also extracted and stored
3. I used TFIDFVectorizer for vectorization
4. SVD was applied on vectorized data to get reduced representation of the input matrix which enable us to run clustering on huge amount of data
5. Fastcluster library was used to perform hierarchial clustering for single linkage and average on LSA matrix using eculidean distance
6. Generated dendograms to visualize the clusters
7. The final result was stored in format : URL, Clusternumber

Clusters obtained
Single linkage and Average linkage clustered the data into 10 clusters. All the data was clustered into these 10 clusters.

Clustering results presentation on UI
Once the user selects one of the clustering buttons from Flat Clustering, Single Linkage, Average Linkage. First, the user selects one of the vector relevance models and then on top of it we choose one of the clustering techniques.

Queries experimented to improve result
I have tested clustering method using 50 queries to improve the results generated. Results of the queries ran are listed below and observations on the queries are also listed below each query result

Query testing generation and impact on results and relevancy
For both clustering methods flat and hierarchical I ran 50 queries. Queries were given manually and depending on initial results obtained from the original relevance model(TF-IDF) using page rank and HITS. Results after applying clustering seem more relevant as related data was clustered and it was better than non-relevant results

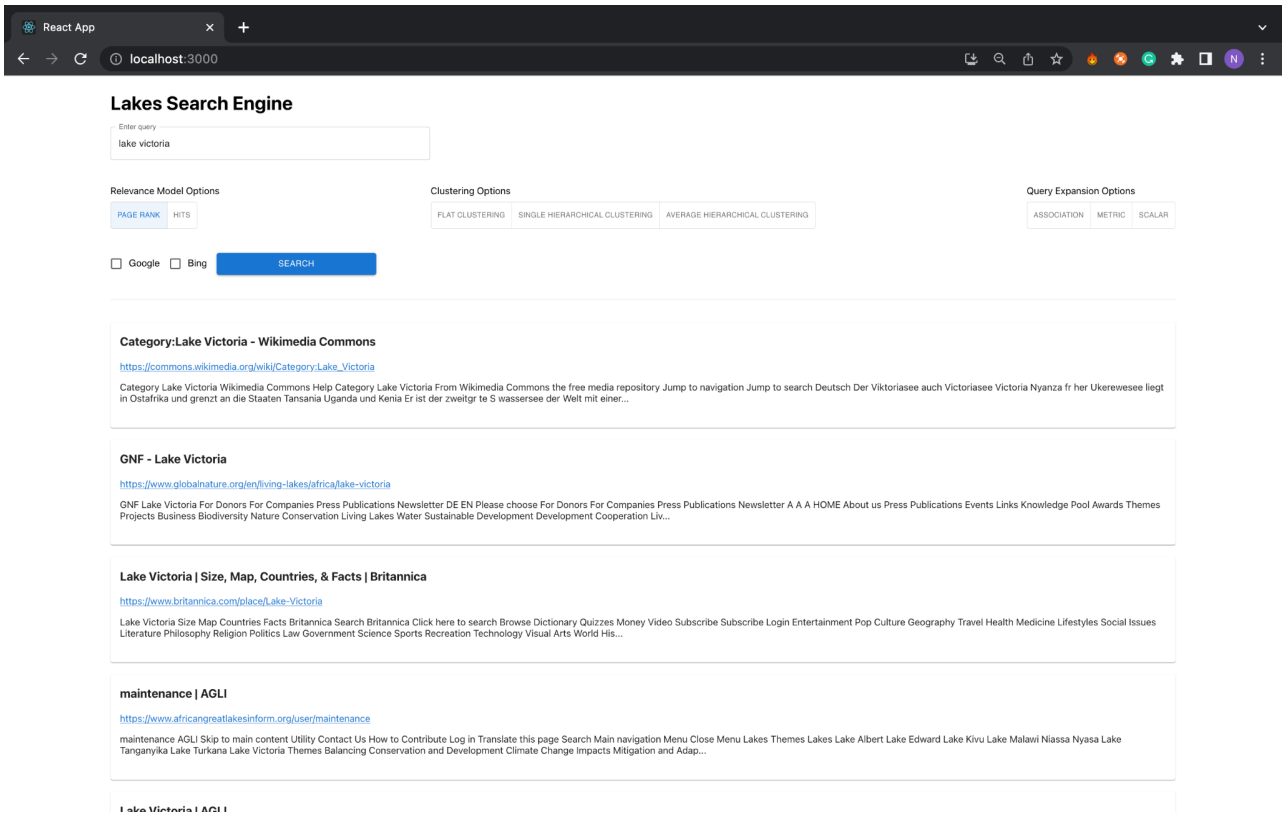
Query selection:
I selected the queries depending on how the data is classified into clusters for the Hierarchial Agglomerative clustering techniques. From the above screenshots showcasing the results, we can see that popular lakes around is a criterion to choose queries and the corresponding results show that they are classified into a cluster. One

more criteria that we used is based on the topic associated like type of lakes or based on country. For instances, all artificial lakes will be classified into a cluster.

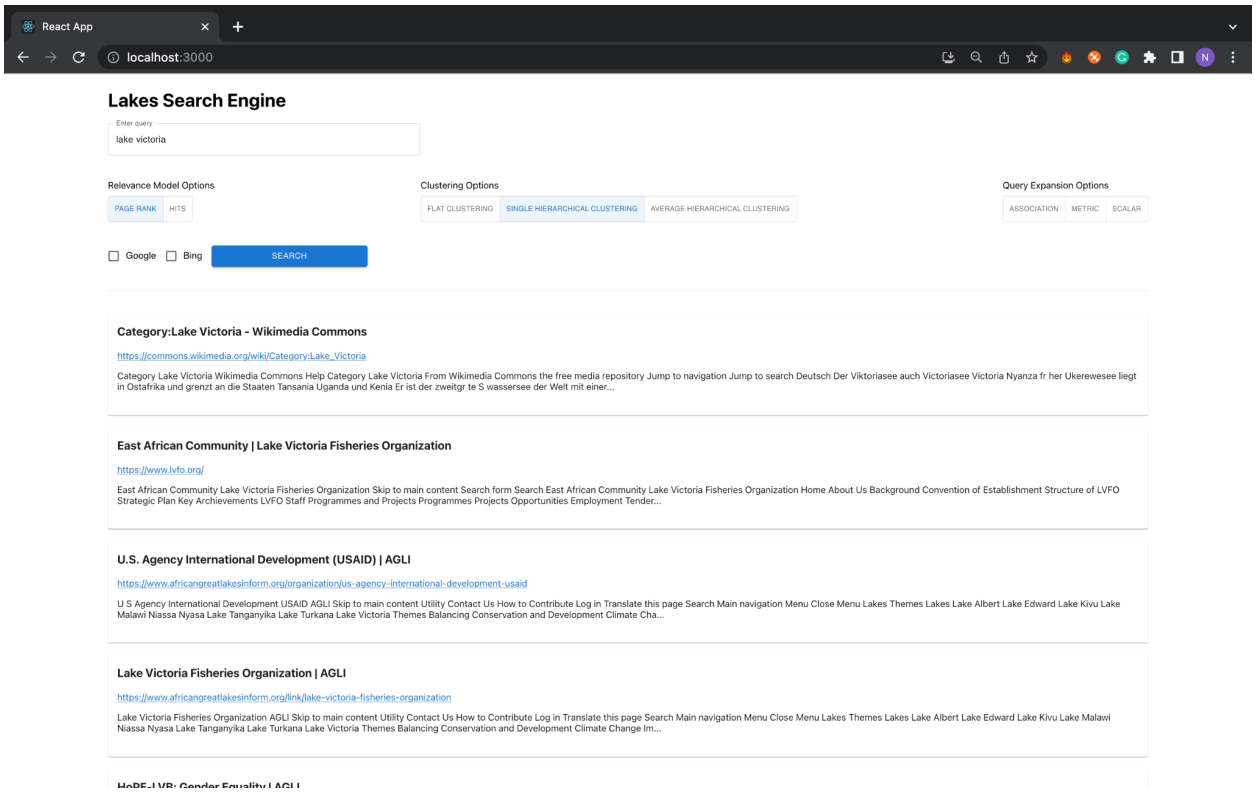
Query examples with clustering

I have attached 5 queries. First image is about page ranking, second image is flat clustering, third image is single linkage hierarchial and fourth image is average linkage heirarchial

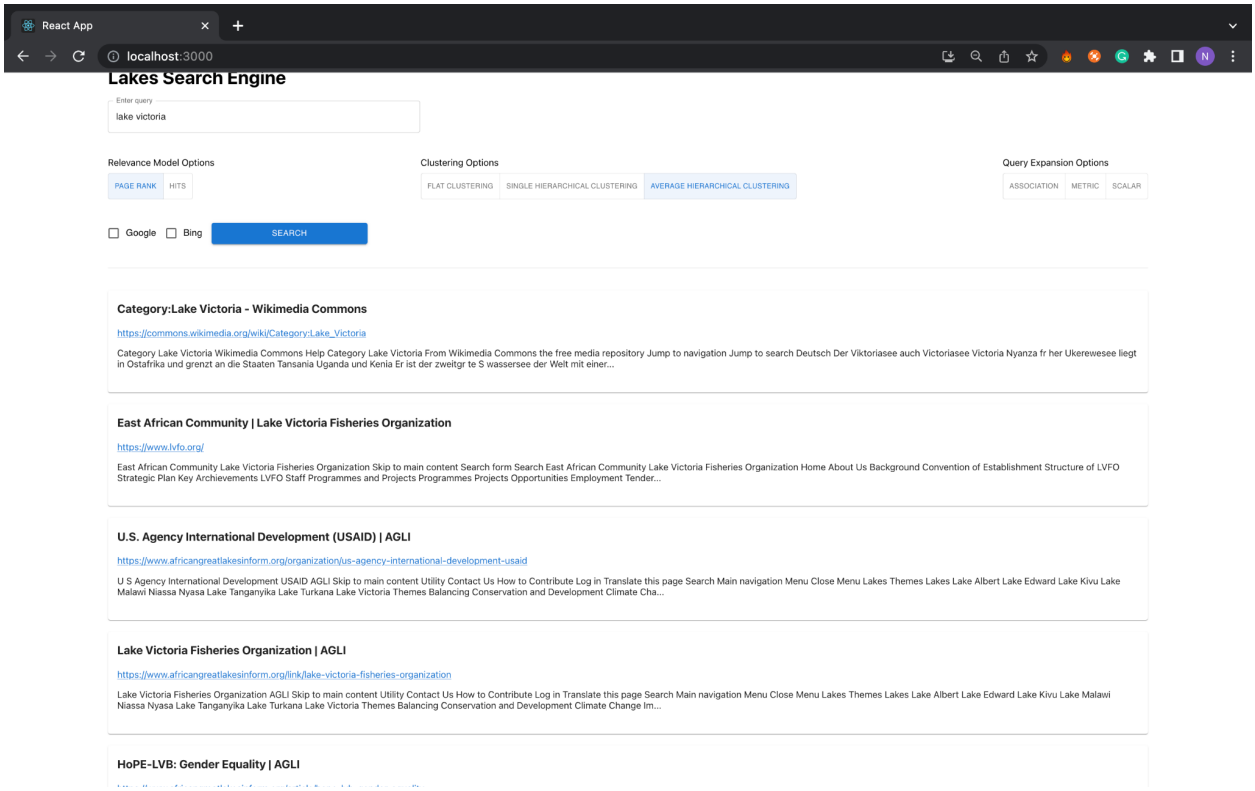
1. Lake Victoria



Nausheen Fathima, nxf200000



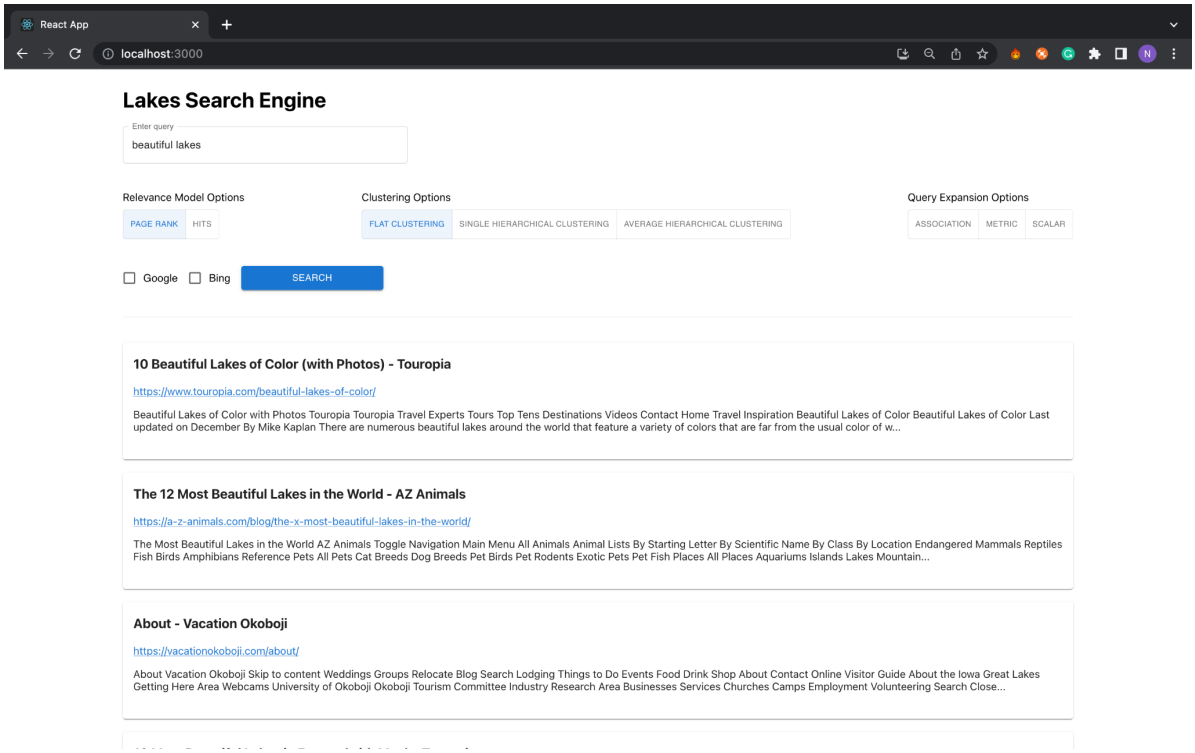
Nausheen Fathima, nxf200000

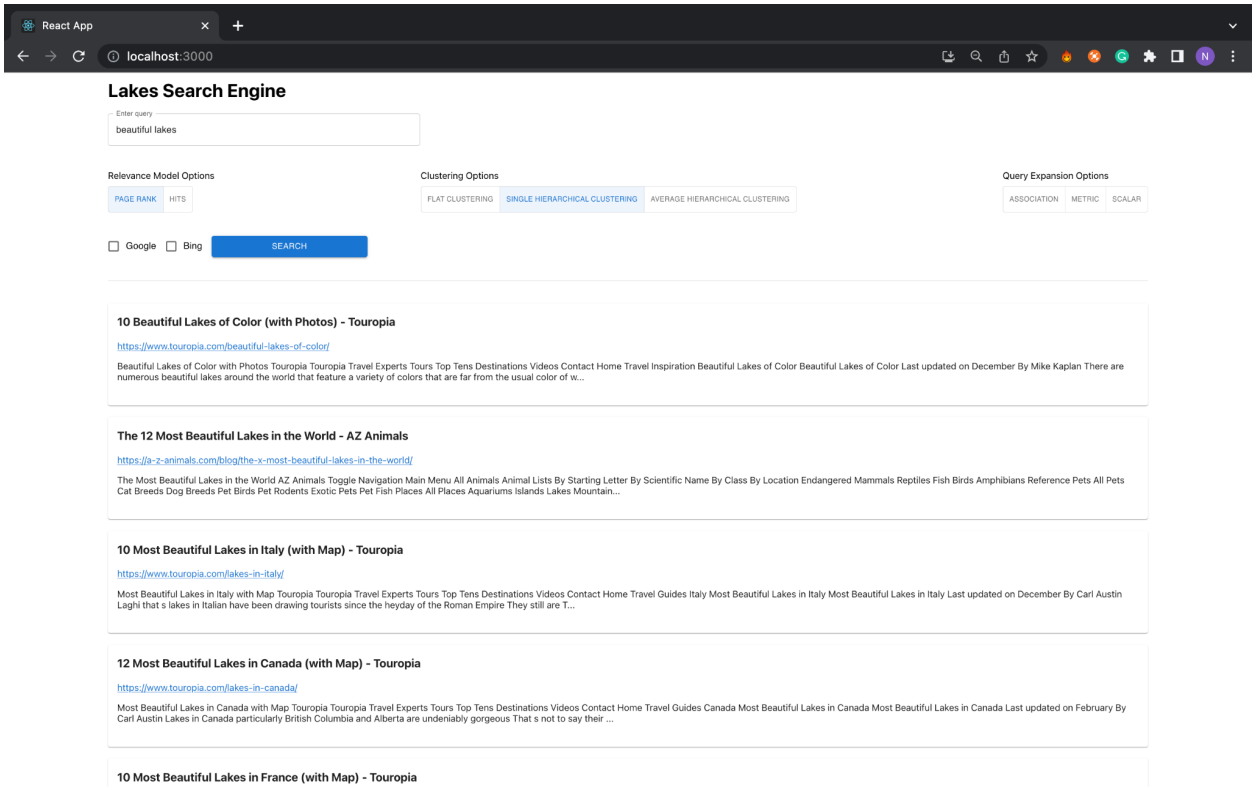


Observations: The above query of Lake Victoria with just page rank results in websites that are even remotely related to lake victoria like the maintenance which is just a african great lakes website which mentions lake victoria links in their maintenance site. However, clustering techniques removes that and gives more relevant pages to the query.

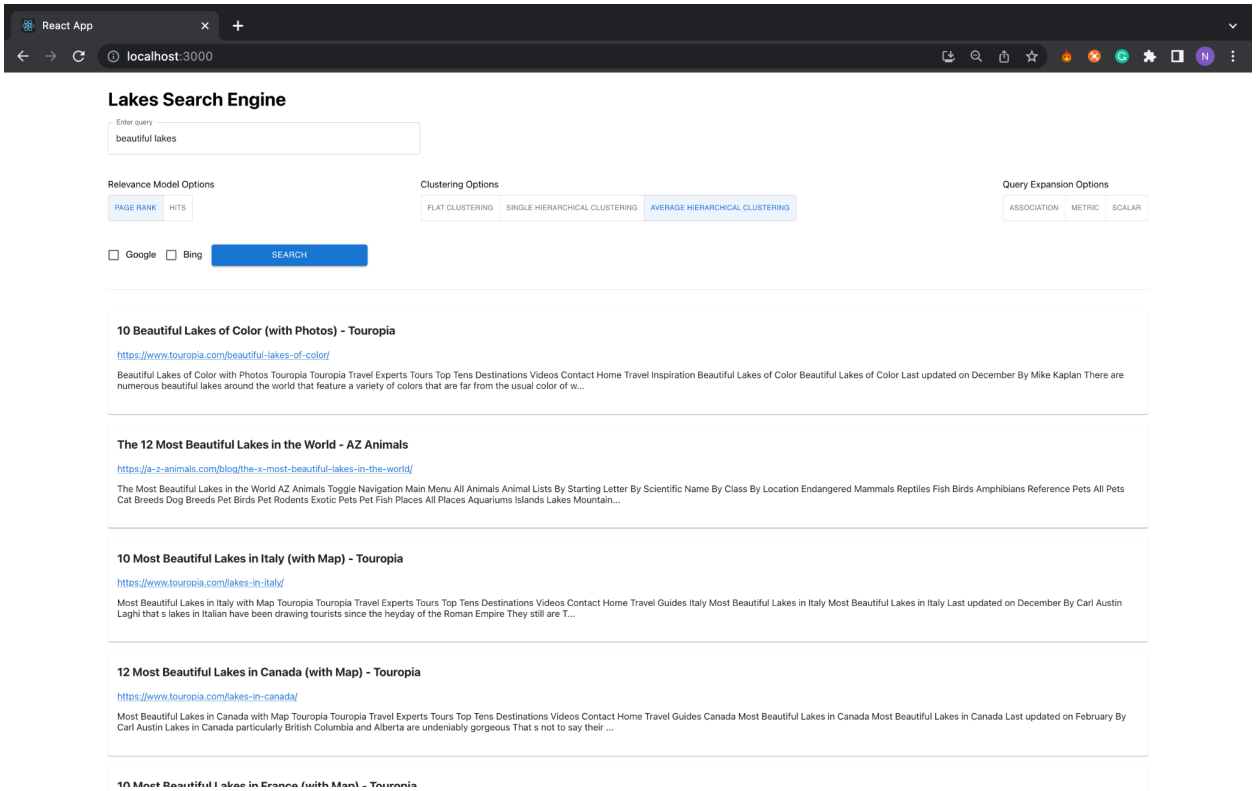
Nausheen Fathima, nxf200000

2. Beautiful lakes





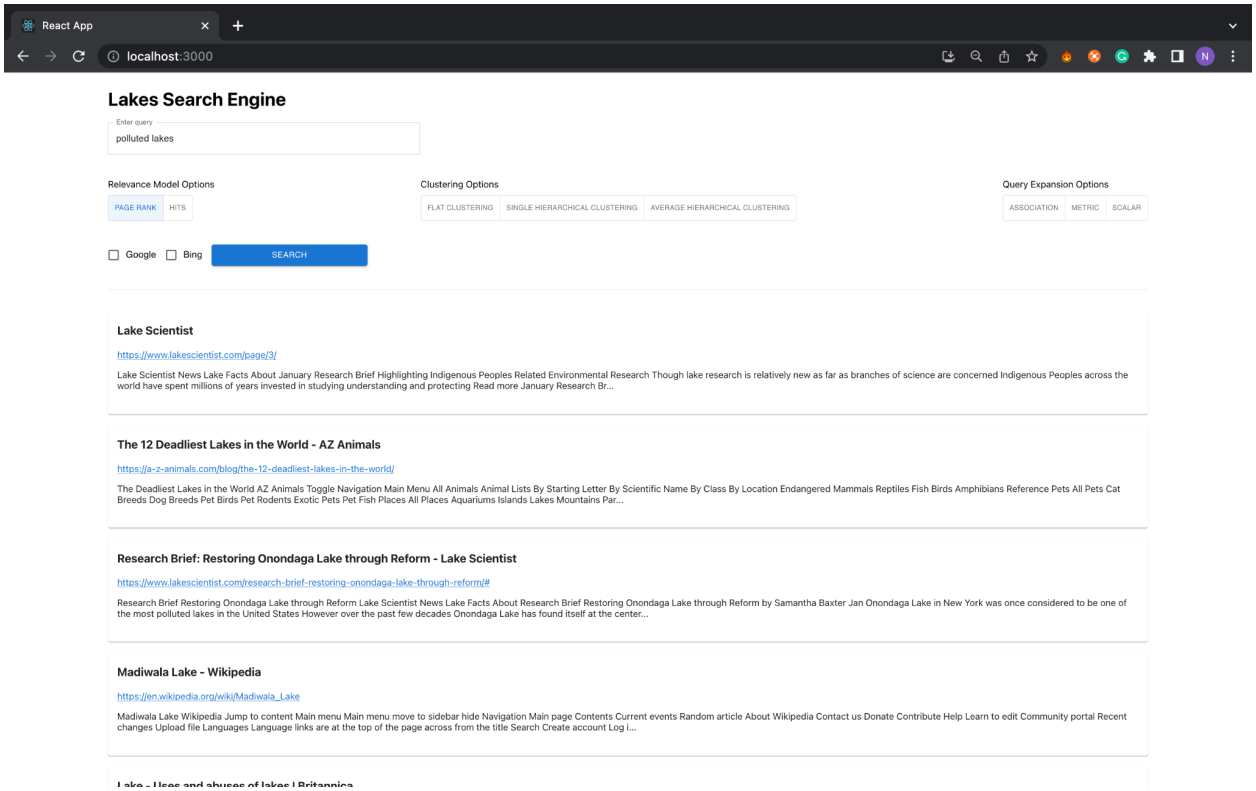
Nausheen Fathima, nxf200000



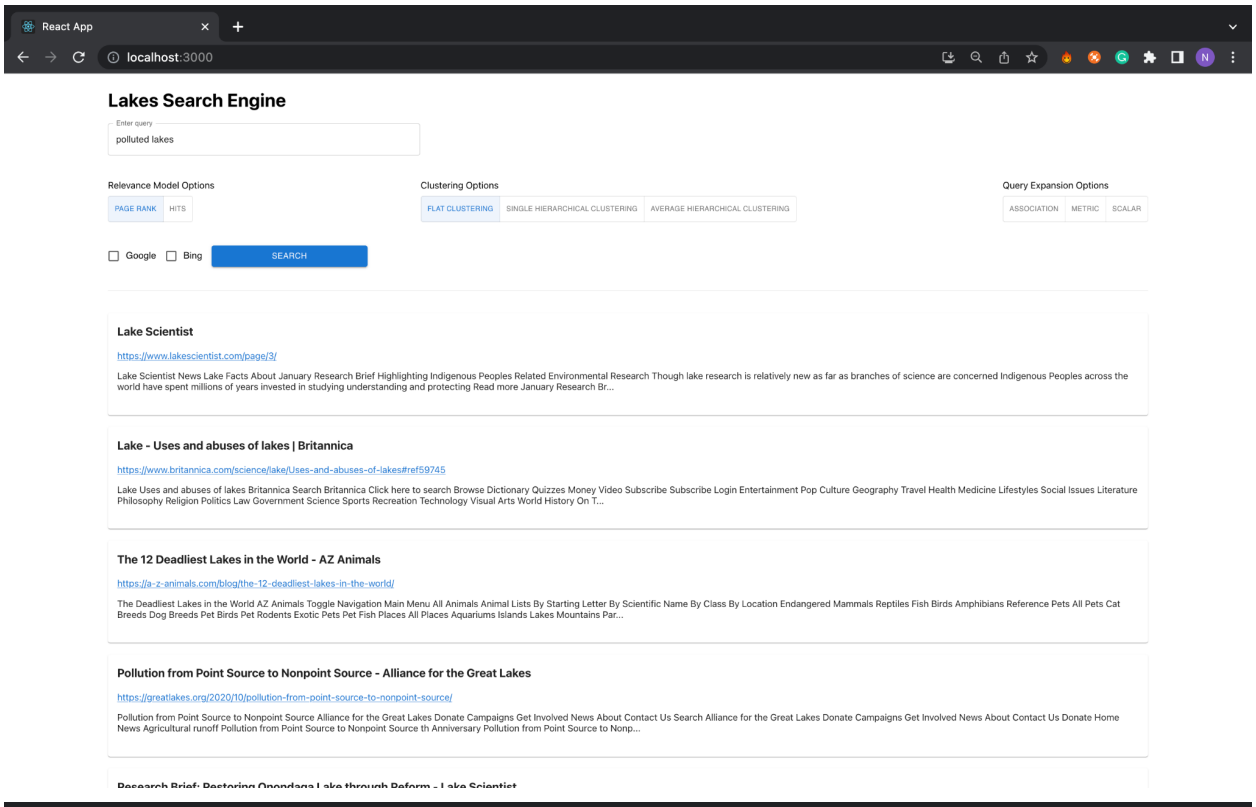
Observations: Clustering gave more relevant such as results of top 10 beautiful lakes, 12 most beautiful lakes, 10 most beautiful lakes in italy where as compared to HITS which gave website about lake lodge on lake seymour.

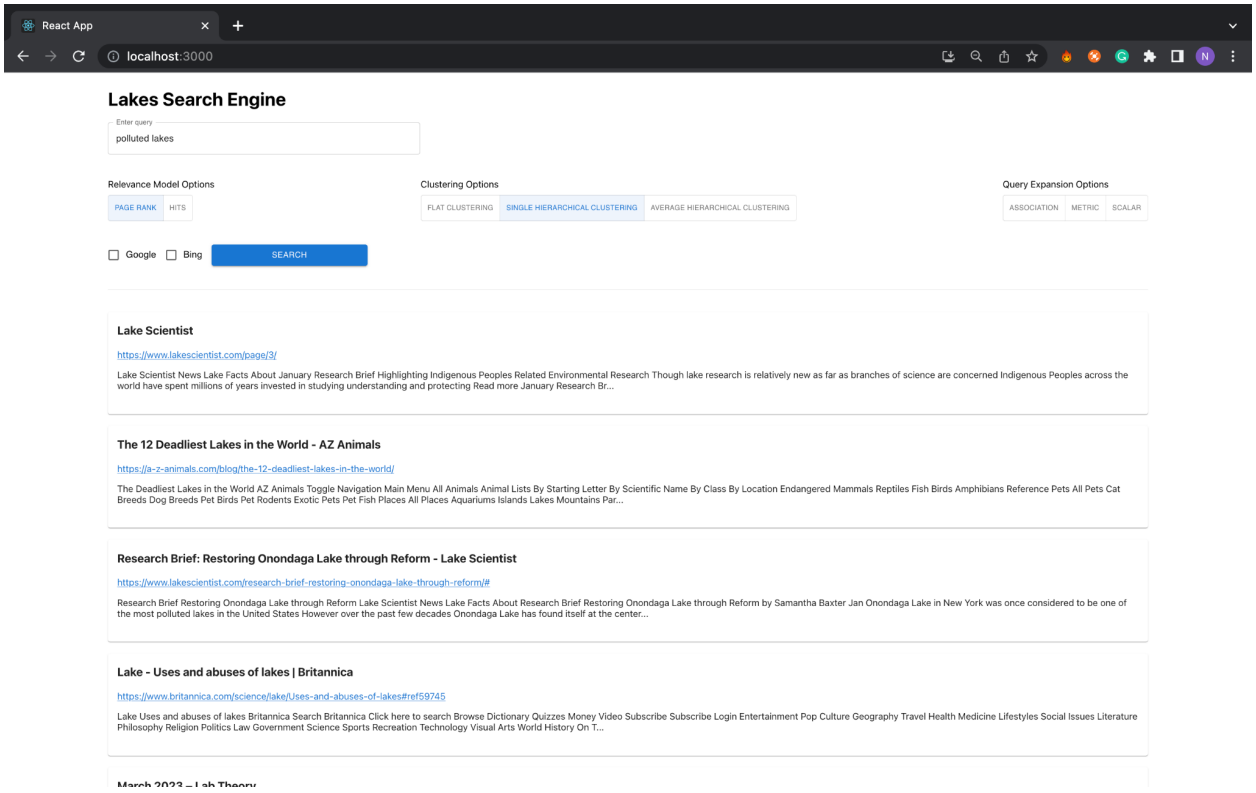
Nausheen Fathima, nxf200000

3. Polluted Lakes

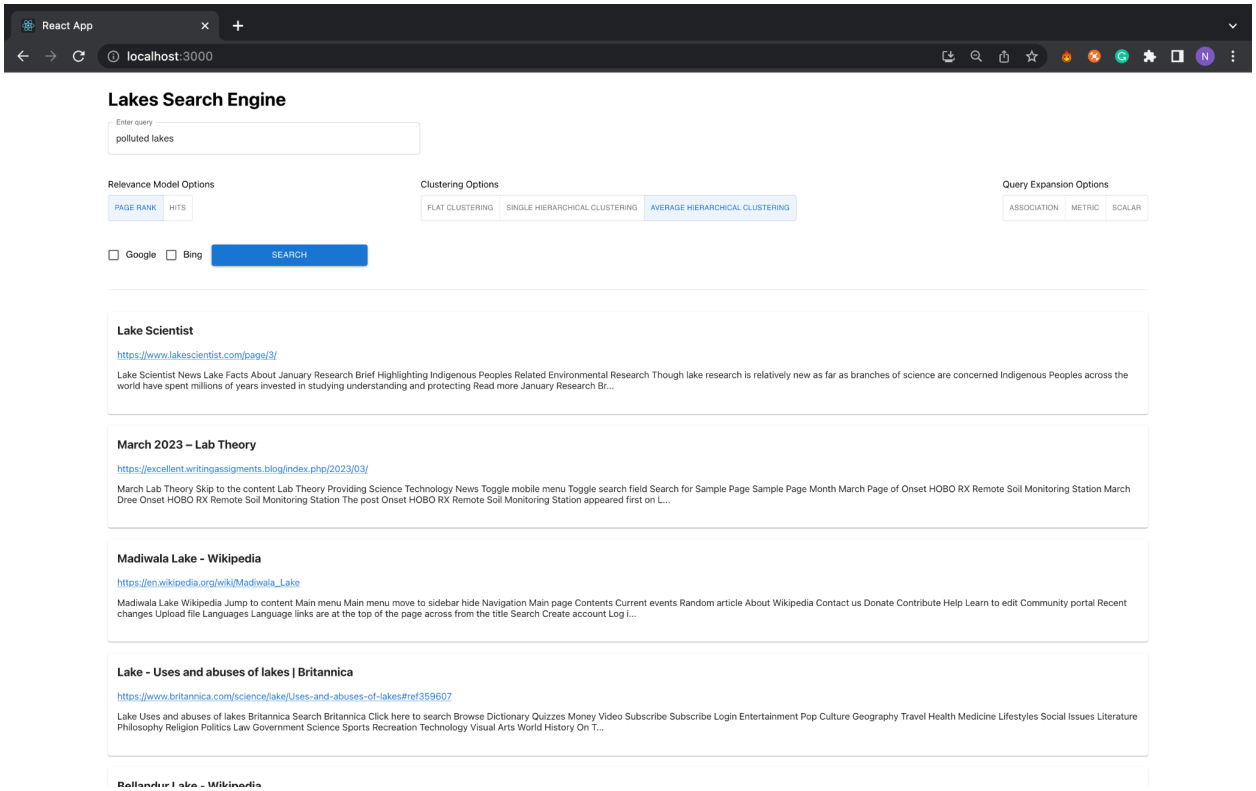


Nausheen Fathima, nxf200000





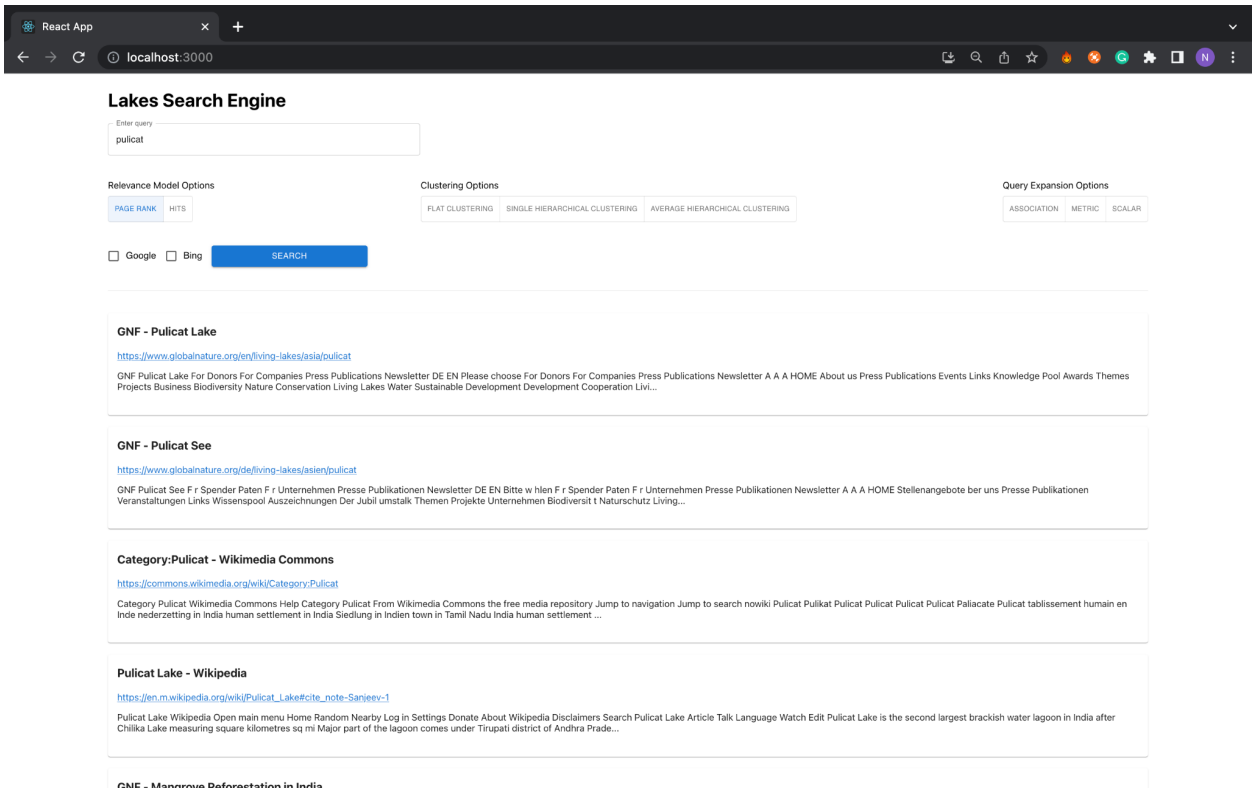
Nausheen Fathima, nxf200000



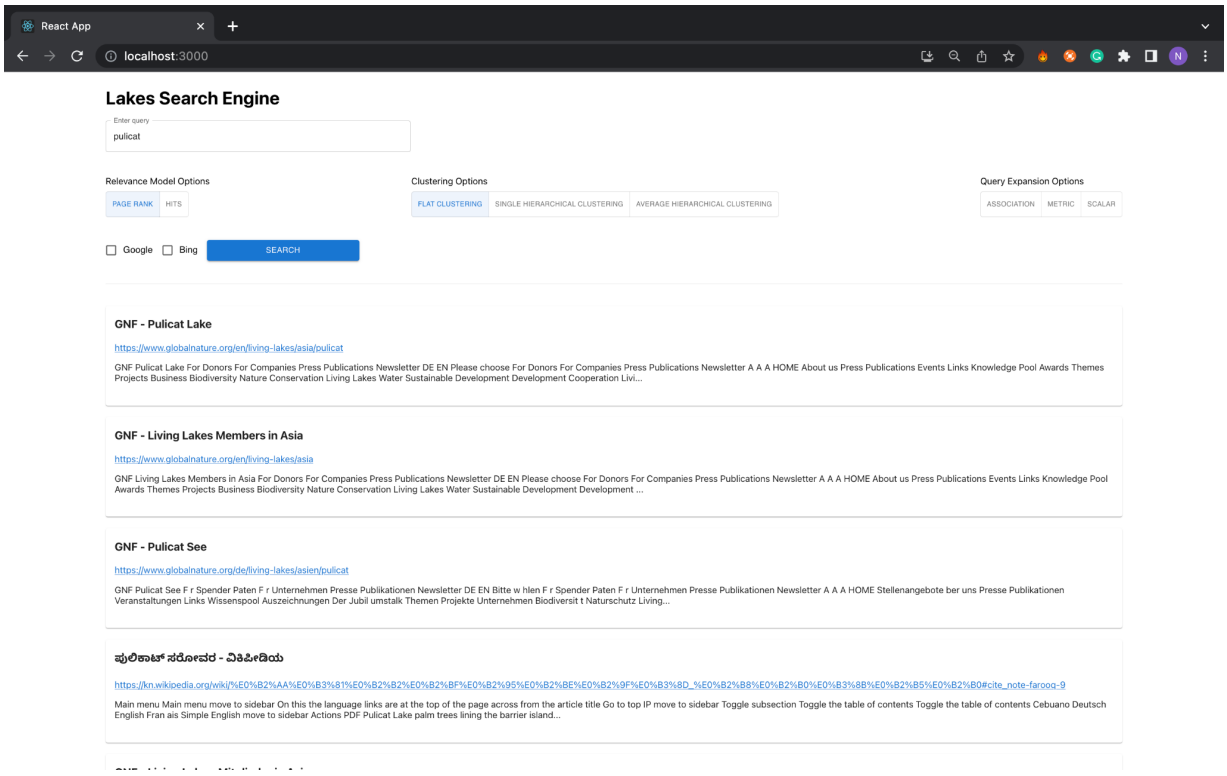
Observations: Clustering gave results such as pollution from point source, uses and abuses of lakes, Madiwala lakes when compared to pagerank which gave about research

Nausheen Fathima, nxf200000

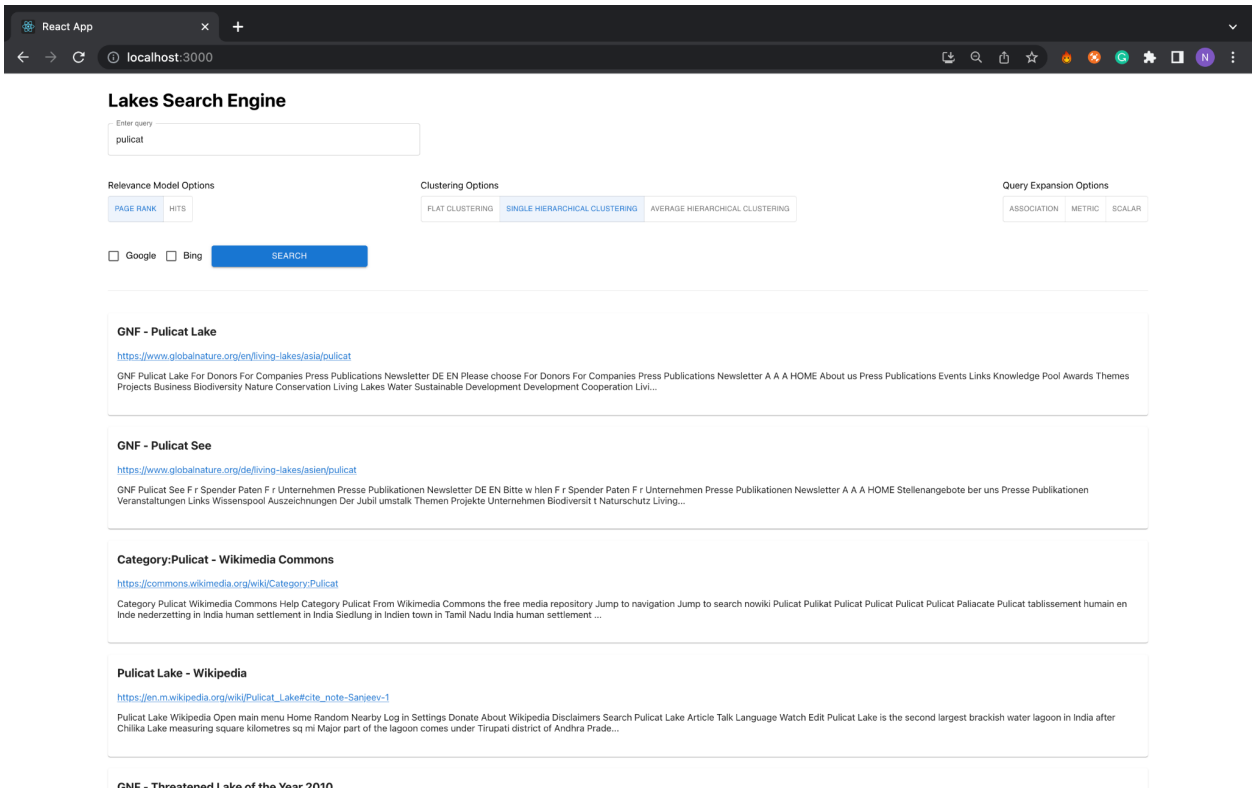
4. Pulicat

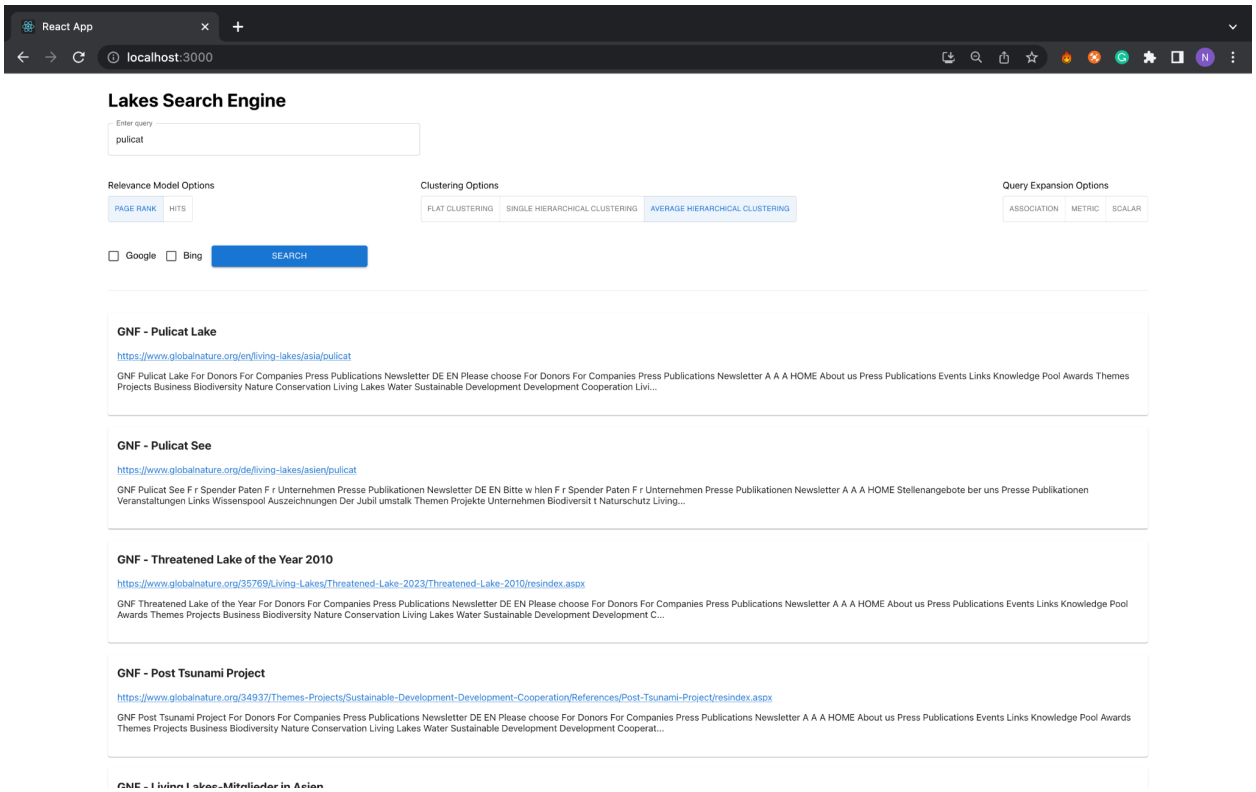


Nausheen Fathima, nxf200000



Nausheen Fathima, nxf200000

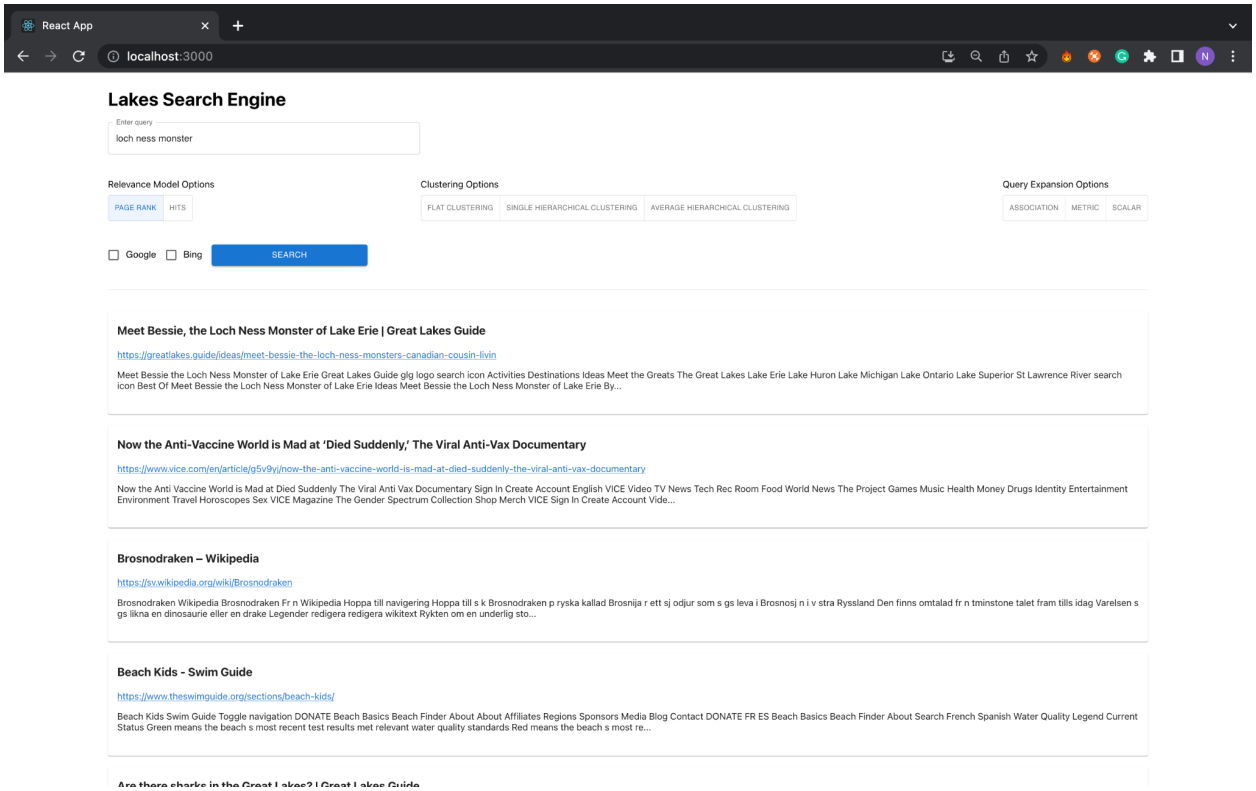




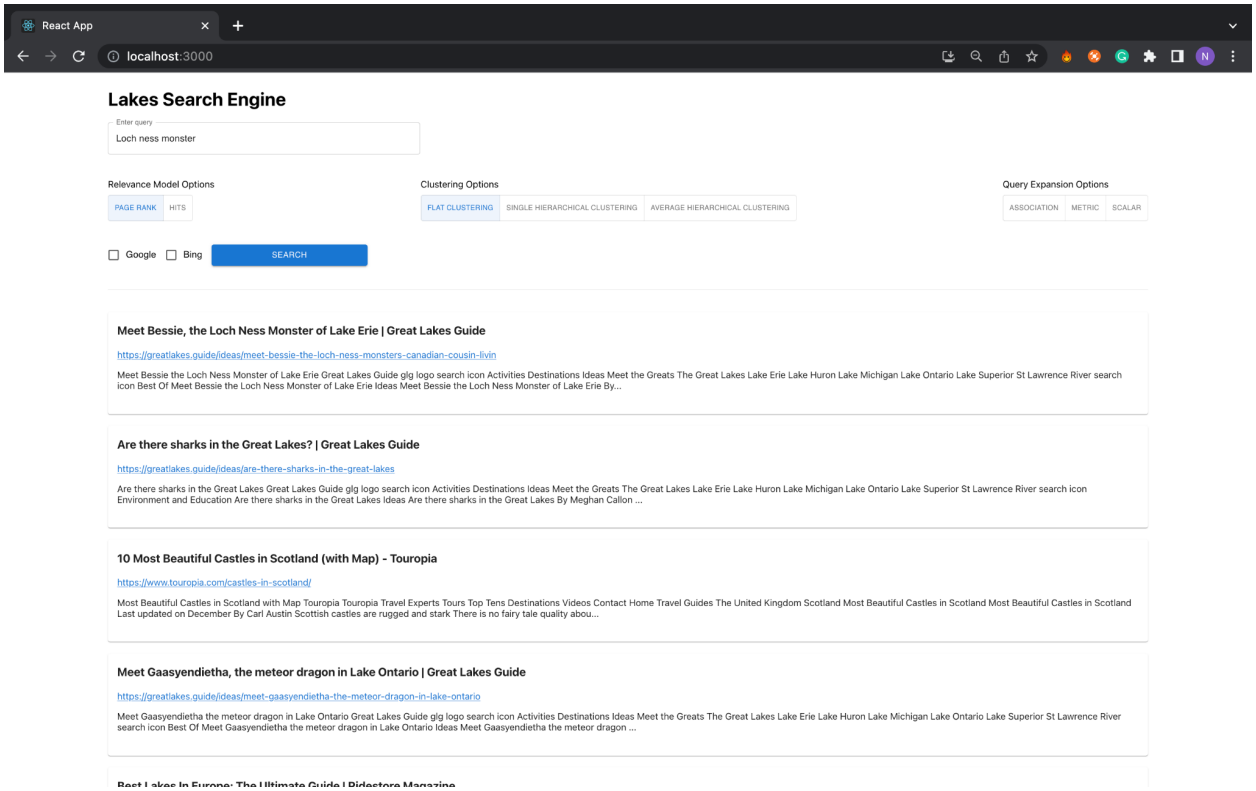
Nausheen Fathima, nxf200000

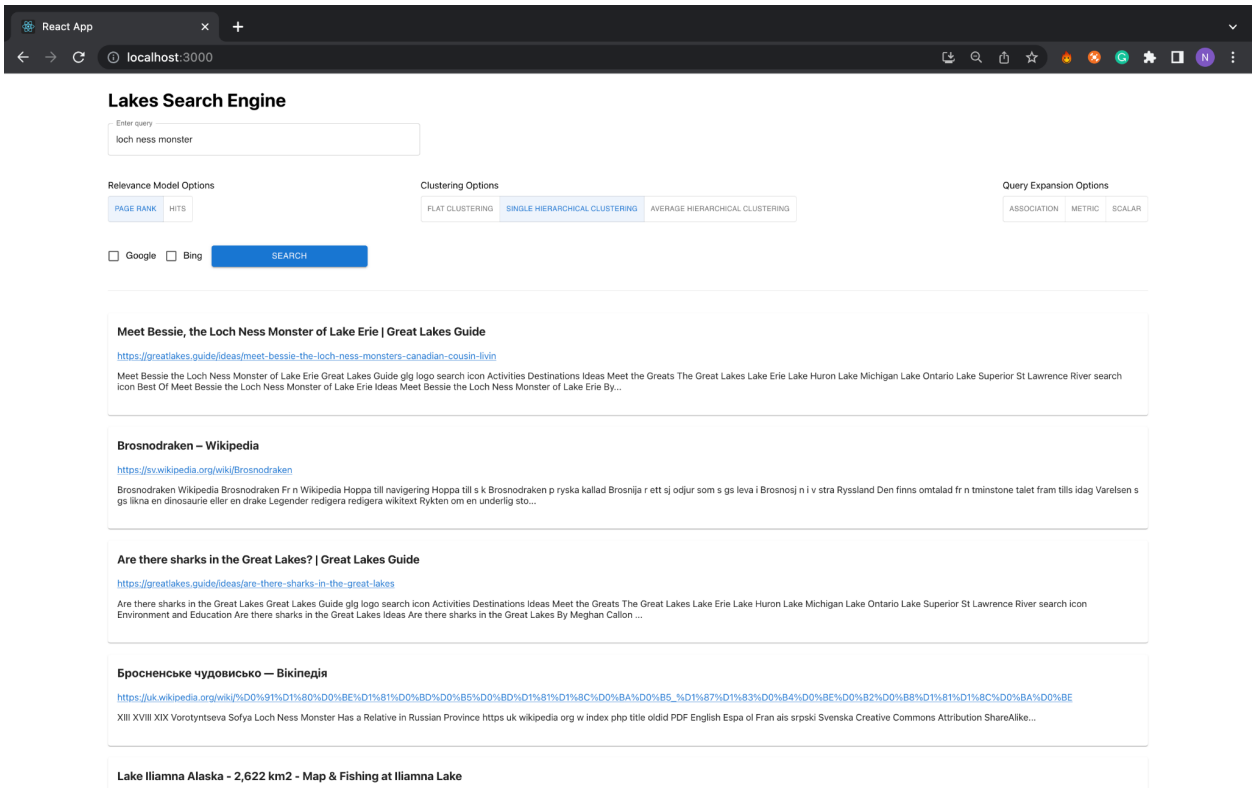
Observations: Clustering gave pulicat lake wiki websites talking about threatened lake and post tsumami project whereas page rank gave websites related to mangrove restoration

5. Loch ness monster

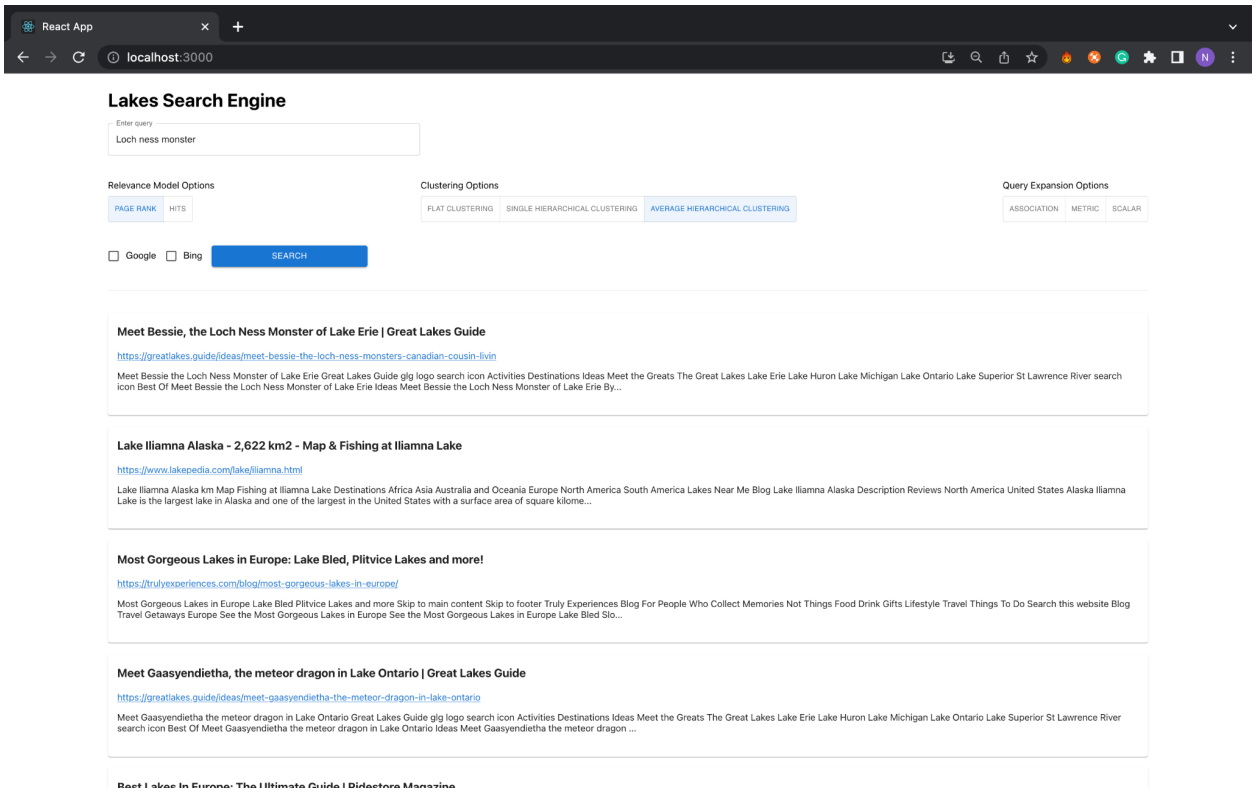


Nausheen Fathima, nxf200000





Nausheen Fathima, nxf200000



Observation: The above query using page rank results in websites like swim guide and anti-vax documentary which contains few paragraphs about the Loch Ness Monster; however, using clustering techniques, it results in websites which talks about the sightings or the myth originated from scotland itself.

Varasiddhi Jayasuryaa Govindraj, VXG190049

6. Query expansion and relevance feedback (45 points): Varasiddhi Jayasuryaa Govindraj, VXG190049

By using the local strategies listed below, query expansion is employed to enhance search results:

1) Relevance Feedback (Rocchio Algorithm)
2) Pseudo Relevance Feedback
   - Association cluster
   - Metric cluster
   - Scalar cluster

The Roccio algorithm is implemented using the following formula:

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

q_m = modified query vector; q_0 = original query vector;
α, β, γ: weights
D_r = set of known relevant doc vectors;
D_{nr}= set of known irrelevant doc vectors

I gave 20 queries as q0 as inputs. The algorithm was tested by setting certain values for the parameters, alpha=1.0, gamma=0.1, beta=0.9.

The 20 queries were selected based on no. of query terms, spelling mistakes and queries containing words that are outside of lakes domain.

1. Five queries with one search term regarding the data that was crawled
   a. texas
   b. boating
   c. lakes
   d. fishing
   e. michigan

2. Five queries with two search terms regarding the data that was crawled
   a. lakes fishing
   b. lakes boating

c. beautiful lakes
d. polluted lakes
e. deepest lakes

3. Five queries with spelling mistakes
   a. laks
   b. laks txas
   c. boting
   d. fising
   e. bautiful laks

4. Five queries outside the lakes domain
   a. monster
   b. waste
   c. table
   d. germs
   e. Electronics

Examples of web pages that are relevant:

Query	**Relevant web pages**
michigan lakes	https://lakelubbers.com/lake-location/usa/midwest/michigan/northeast-michigan/
	https://www.aa-fishing.com/mi/michigan-ice-fishing.html
	https://www.aa-fishing.com/mi/michigan-crappie-fishing.html
	http://Michigan.USLakes.info/Events/
beautiful lakes	https://www.touropia.com/beautiful-lakes-of-color/
	https://a-z-animals.com/blog/the-x-most-beautiful-lakes-in-the-world/#
	https://www.touropia.com/lakes-in-italy/
	https://www.touropia.com/lakes-in-canada/
polluted lakes	https://a-z-animals.com/blog/the-12-deadliest-lakes-in-the-world/
	https://en.wikipedia.org/wiki/Madiwala_Lake
	https://www.britannica.com/science/lake/Uses-and-abuses-of-lakes#ref59745
	https://greatlakes.org/2020/10/pollution-from-point-source-to-nonpoint-source/

Varasiddhi Jayasuryaa Govindraj, VXG190049

Examples of web pages that are irrelevant:

Query	Irrelevant web pages
michigan lakes	https://www.aa-fishing.com/mi/michigan-fishing.html
	http://Michigan.USLakes.info/Classifieds/Boats/
beautiful lakes	https://www.touropia.com/best-things-to-do-in-quebec-city
	https://www.touropia.com/how-to-spend-2-weeks-in-germany/
polluted lakes	https://www.lakescientist.com/page/3/
	https://excellent.writingassigments.blog/index.php/2023/03/

Varasiddhi Jayasuryaa Govindraj, VXG190049

Modified queries after applying Roccio algorithm to above queries:

Original Query	**Query after Rocchio algorithm**
lakes	lakes great
reef	reef map
swim	swim kids
waste	waste lake
california	california beach
famous	famous water
fish	fish eat
crater	crater lake
pond	pond fishing
loch ness	loch ness monster
underwater	underwater cave
falls	falls water
preservation	preservation national
hyderabad	hyderabad fish
dead	dead sea
algae	algae plant
gills	gills breathe
stratification	stratification lake
sedimentary	sedimentary caves
shore	shore beach

Varasiddhi Jayasuryaa Govindraj, VXG190049

The approach proposed was utilized for the 20 queries shown above. These were our findings:

1. Long queries (such as “best lakes in US”) did not fetch any documents.
2. Relevant results come up after rank 20. The user would rarely reach these documents.
3. Performance issues were high as it took time to get stabilized weights. Thus, on the spot calculations could not be performed.
4. Query expansion did result in few queries that gave appropriate and relevant results however, there were a lot of queries which did not give results which are relevant. Additionally, there were few queries which did not generate any results as we could crawl only a limited number of webpages and some expanded queries did not give results

We have not incorporated the Rocchio algorithm into our User Interface because it is challenging to implement in the framework we use.

Pseudo Relevance Feedback:

1. **Association Cluster:** The stems that co-occur frequently in documents have a synonymy connection.
2. **Metric Cluster:** The theory holds that stems that are spread out across the document are less correlated with terms that are close together, such as in a single sentence.
3. **Scalar Cluster:** The idea is that two stems are more co-related if they have similar neighborhoods.

Varasiddhi Jayasuryaa Govindraj, VXG190049

50 queries chosen for testing Pseudo-relevance

Query	**Relevant results fetched after expansion**	**Most relevant expanded query**	**Cluster method used**
great lakes	3	great lakes page	metric
fishing	17	fishing kids	association
california	19	california san	metric
boating	19	boating get	association
texas	20	texas page	association
stratification	7	stratification lake	association
sea	20	sea gulf	association
artificial	4	artificial as	scalar
lakes	20	lakes policy	metric

Varasiddhi Jayasuryaa Govindraj, VXG190049

waste	8	waste also	metric
polluted	1	polluted one	association
swim	12	swim every	scalar
deepest	1	deepest on	scalar
frozen	20	frozen trivia	association
clear	20	clear lake	association
blue	1	blue just	scalar
salt	6	salt sea	scalar
famous	20	famous one	association
danger	6	danger you	scalar
biggest	9	biggest historical	association

Varasiddhi Jayasuryaa Govindraj, VXG190049

Query: **boating**

Local Document Sets:

https://www.boatus.org/free/
https://www.boatsetter.com/regulations
https://www.boat-ed.com/boating_license/
https://www.boatus.org/alcohol-and-boating/effects/
https://www.boatus.org/alcohol-and-boating/
https://www.boat-ed.com/
https://www.boat-ed.com
https://www.boat-ed.com/#price-and-payment
https://www.boat-ed.com/#states
https://www.boat-ed.com/#main
https://www.boat-ed.com/#select-your-course
https://www.boat-ed.com/#top
https://www.boat-ed.com#select-your-course
https://www.boat-ed.com#main
https://www.boat-ed.com#top
https://www.boatus.org/cold-water-boating/infographic/
https://www.boatus.org/alcohol-and-boating/myths/
https://www.boatus.org/life-jackets/infographic/
https://www.boatus.org/cold-water-boating/
https://www.boatus.org/about/mission/

Local Vocabulary Set:

Vocabulary Size: 966

boating: 1521
course: 883
get: 728
safety: 700
license: 671
certified: 640
official: 603
state: 391

Varasiddhi Jayasuryaa Govindraj, VXG190049

topics: 344
boater: 303
department: 293
resources: 265
boat: 257
clean: 193
courses: 182

Local Stem Set:

Stem set size: 804

boat: 1783
cours: 1065
get: 740
safeti: 700
licens: 671
certifi: 640
offici: 604
state: 505
boater: 380
topic: 349
depart: 293
resourc: 265
educ: 213
clean: 201
natur: 150

Query: fishing

Local Document Sets:

https://outdoorsman.guide/how-to-fishing-guides-and-reports/#content
https://outdoorsman.guide/how-to-fishing-guides-and-reports/
https://outdoorsman.guide/arizona-fishing-lakes/#
https://outdoorsman.guide/arizona-fishing-lakes/#content
https://outdoorsman.guide/arizona-fishing-lakes/

https://outdoorsman.guide#

Varasiddhi Jayasuryaa Govindraj, VXG190049

https://outdoorsman.guide#content
https://outdoorsman.guide
https://outdoorsman.guide/
https://outdoorsman.guide/colorado-fishing-lakes/#
https://outdoorsman.guide/colorado-fishing-lakes/#content
https://outdoorsman.guide/colorado-fishing-lakes/
https://outdoorsman.guide/washington-fishing-lakes-rivers/
https://outdoorsman.guide/oregon-fishing-lakes/#content
https://outdoorsman.guide/new-mexico-fishing-lakes/
https://outdoorsman.guide/oregon-fishing-lakes/
https://outdoorsman.guide/tennessee-fishing-lakes-rivers/#
https://outdoorsman.guide/nevada-fishing-lakes/#
https://outdoorsman.guide/nevada-fishing-lakes/#content
https://outdoorsman.guide/tennessee-fishing-lakes-rivers/#content

Local Vocabulary Set:

Vocabulary Size: 397

fishing: 5603
guide: 3403
lake: 2188
lakes: 1374
best: 589
guides: 506
books: 421
rivers: 330
bass: 284
arizona: 283
white: 199
trout: 174
california: 165
read: 162
state: 160

Local Stem Set:

Varasiddhi Jayasuryaa Govindraj, VXG190049

Stem set size: 380

fish: 5708
guid: 3909
lake: 3562
best: 589
book: 421
river: 359
arizona: 286
bass: 284
white: 199
trout: 174
california: 165
read: 162
state: 160
pond: 155
menu: 140

Query: Switzerland

Local Document Sets:

https://ig.wikipedia.org/wiki/Switzerland
https://www.touropia.com/explore/switzerland/
https://en.wikipedia.org/wiki/Category:Caves_of_Switzerland
https://en.wikipedia.org/wiki/Residence_time_of_lake_water
https://en.wikipedia.org/wiki/Lake_retention_time#See_also
https://www.touropia.com/best-places-to-visit-in-switzerland/
https://en.wikipedia.org/wiki/Template_talk:Valais-lake-stub
https://en.wikipedia.org/wiki/Talk:Saint-L%C3%A9onard_underground_lake
https://en.wikipedia.org/wiki/File:Valais-coat_of_arms.svg
https://www.touropia.com/tourist-attractions-in-lucerne/
https://www.touropia.com/tourist-attractions-in-zurich/
https://en.wikipedia.org/wiki/Category:Show_caves_in_Switzerland
https://en.wikipedia.org/wiki/Saint_L%C3%A9onard,_Switzerland
https://en.wikipedia.org/wiki/Saint-L%C3%A9onard,_Switzerland
https://en.wikipedia.org/wiki/Ferden
https://www.pinterest.com/ideas/beautiful-scenery/905201467809/
https://www.ridestore.com/mag/most-popular-ski-resorts-in-the-world/

Varasiddhi Jayasuryaa Govindraj, VXG190049

https://en.wikipedia.org/wiki/List_of_international_lakes#Europe
https://www.wbcsd.org/Overview/Events
https://www.pinterest.com/ideas/visiting/944710546669/

Local Vocabulary Set:

Vocabulary Size: 4450

lake: 486
switzerland: 310
years: 286
united: 152
states: 125
ski: 116
popular: 115
swiss: 101
edit: 99
see: 87
wikipedia: 84
travel: 83
lakes: 80
resorts: 75
articles: 75

Local Stem Set:

Stem set size: 3921

lake: 566
switzerland: 315
year: 305
unit: 156
ski: 137
state: 126
popular: 123
page: 122
edit: 111
articl: 107

Varasiddhi Jayasuryaa Govindraj, VXG190049

swiss: 101
travel: 99
resort: 97
use: 95
time: 95

Collaboration with UI and relevance model:

1. The query to be searched and the clustering method to be used is received from the UI
2. Top 20 documents are taken from the original result and then given to the query expansion program along with the original query passed.
3. The returned documents are used to build vocabulary and stem sets.
4. Based on the query expansion type selected, the processed documents are sent to the corresponding method to expand the query.
5. The expanded query is returned by the assigned method and sent to the relevant model to fetch new relevant documents.
6. The final results are returned to the UI through the API and displayed along with the expanded query.

Query selection for demo:
For the demo presentation, the query “lakes” has been chosen since it is expanding the query into more relevant terms and that improves the query results significantly. It is tested for association cluster method.

7. Discussion (2.5 points): All Team

Meghana Sai Bijivemula, mxb210011
Soha Anant Parasnis, sxp200044
Simran Bhake, sxb190165
Nausheen Fathima, NXF200000
Varasiddhi Jayasuryaa Govindraj, VXG190049

The entire team first decided on a tentative timeline for the project and distributed tasks. We discussed every part of the project at appropriate intervals by meeting in-person and on Microsoft Teams. This made sure everyone was on the same page regarding the various tasks as well as technologies to finalize. For collaborations between specific tasks (like the student who did crawling collaborating with the student who did indexing), we scheduled special meetings or calls to make sure such integrations happened smoothly.

For memory intensive tasks like clustering, we used Google Colaboratory Pro. We used GitHub for version control and efficient collaboration between the team members.

After completing this project, we learned about the many different aspects required to make a simple and working search engine. We created a search engine that crawled web data related to lakes, indexed it, and then returned somewhat relevant results based on various queries. While this project provided with a great deal of practical knowledge, we also know that this is just the tip of the iceberg that is Information Retrieval.

8. Conclusion (2.5 points): All Team

Meghana Sai Bijivemula, mxb210011
Soha Anant Parasnis, sxp200044
Simran Bhake, sxb190165
Nausheen Fathima, NXF200000
Varasiddhi Jayasuryaa Govindraj, VXG190049

We created a search engine that focuses on Lakes. We were able to put our theoretical study gained throughout this class to practice to develop this project. We faced plenty of challenges along the way, however, we tried our best to troubleshoot them and work through those challenges to create a functional search engine that satisfied all of the requirements of the project. This whole process also helped us strengthen our understanding of the theoretical concepts we learned in class.

In conclusion, it was a challenging but fun project.