The University of Texas at Dallas

CS 6322

Information Retrieval

Spring 2023

Class Project Report

Project TITLE: Search Engine for Mountains

Group: No. 11

Students

Arpit Singh, axs210204@utdallas.edu
Tanmay Singhal, txs210014@utdallas.edu
Prem Sharma, pxs210046@utdallas.edu
Karan Jariwala, khj200000@utdallas.edu
Anirudh Kiran, axk200227@utdallas.edu

Introduction

We designed and developed a search engine that focuses on collecting information about mountains. The following section shows the main tasks and high-level architecture for building the search engine and details about the person who worked on those tasks:

- **Crawling**: Karan Jariwala
- **Indexing and Relevance:** Tanmay Singhal
- **User Interface and Comparisons with Google and Bing:** Anirudh Kiran
- **Clustering:** Prem Sharma
- **Query Expansion and Relevance Feedback:** Arpit Singh

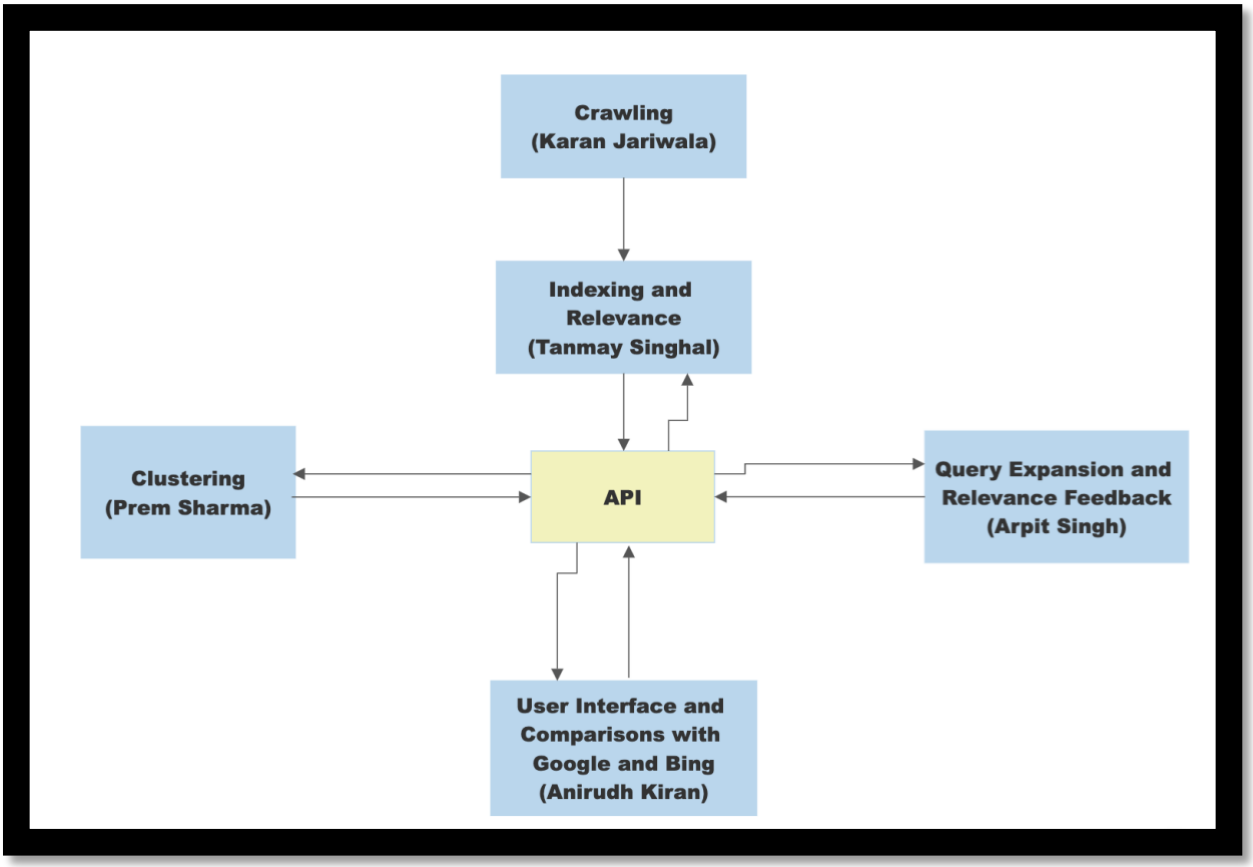


Figure 1 Search Engine Architecture

Learnings from the project: -

1. Working with substantial amounts of data.
2. Insights into design of a system.
3. The necessity of considering scalability, run-time, and space complexity constraints.
4. Usage of open-source software.
5. Usage of multi-threading techniques.
6. Importance of breaking data in small batches and batch-processing.
7. Writing robust code and using checkpoints.

Difficulties faced during the project: -

1. Working with large amount of data and running out of memory.
2. Choosing the right tool and technology for a particular task.
3. Inefficient coding practices.
4. Learning a new technology and applying it on a large scale quickly
5. No access to powerful machines that can handle massive amounts of data.
6. Waiting too long to know the output.

Solutions to the problems faced: -

1. Dividing the data into batches and processing it in batches.
2. Researching in-depth before choosing a technology/tool.
3. Writing efficient code once the functionality is verified.
4. Taking time to learn a new technology.
5. Utilizing the concept of horizontal scaling and batch processing
6. Using batch processing and efficient coding practices.

Karan Jariwala KHJ200000

Crawling

Several different approaches were tried for crawling the web. Some python libraries such as BeautifulSoup, Scrapy and Selenium were experimented with. But these were not particularly useful as these tools did not perform well on the initial seed URLs chosen. An out-of-the-box crawler like Octoparse was also used, but it returned many duplicate pages. I also tried using Wikipedia's API to extract data from Wikipedia. I was able to collect approximately 40,000 pages too. But, after discussion with the team, I realized that we had to have pages from different domains and websites, not just one website. Hence, this approach was also not used. Finally, after research, I decided to use Apache Nutch. I used Apache Nutch 1.19 for crawling the web and collecting data.

I started off with performing search on Google and Bing to observe the top results returned by these search engines. Based on that, a list of seed URLs was generated. The results of web crawling were gathered in a binary format by the Nutch crawler in 3 directories viz. **crawldb**, **linkdb** and **segments**. These binary files had information about the page crawled, hyperlinks and segments of data. These files were shared with other team members by providing them with read access to the data collected.

I was able to crawl **117,685** pages. The following list of 31 websites was used as the seed URLs for the crawling process.

1. http://earmountain.com
2. https://mmbhof.org
3. https://mbaa.net
4. https://www.mountain-heritage.org
5. https://mountain.org
6. https://www.americanavalancheassociation.org
7. https://www.alpine-rescue.org
8. https://mra.org
9. https://www.mountainresearchinitiative.org
10. https://www.mwis.org.uk
11. https://www.mck.or.ke
12. https://mcsa.org.za
13. https://www.banffcentre.ca/banffmountainfestival
14. https://www.himalayanclub.org
15. https://www.alpineclubofcanada.ca/web/
16. https://americanalpineclub.org
17. https://theuiaa.org
18. https://www.ims.bz/en/
19. https://kimff.org
20. https://www.nepalmountaineering.org
21. https://alpineclub.org.nz
22. https://www.smc.org.uk
23. https://alpineclubofhimalaya.com/destinations/india/
24. https://www.mountainproject.com

25. https://www.tripsavvy.com/major-mountain-ranges-in-india-4687498
26. https://travel2next.com/mountains-in-india/
27. https://www.worldatlas.com/articles/the-major-mountain-ranges-in-europe.html#:~:text=The%20Major%20Mountain%20Ranges%20In%20Europe%201%20Pyrenees,Mountains%20...%208%20Black%20Forest%20...%20More%20items
28. https://www.mountainiq.com/
29. https://www.alltrails.com/
30. https://wildlandtrekking.com
31. https://www.blueridgeoutdoors.com

All these websites were browsed to ensure they had relevant information about mountains.

To ensure that the crawled data did not have duplicates, Nutch's built-in **crawldb** was utilized which marked the pages with duplicate content. Using the "***dedup***" job of Nutch, the duplicate URLs were eliminated. During crawling, at least 4452 duplicate pages were eliminated.

As discussed earlier, the crawled output containing the hyperlink information, text and other useful information was shared with other team members by sharing the compressed output file of almost 5 GBs.

Indexing and Relevance

Indexing

I setup the Apache Solr with version 8.11.2 which is built against the 1.19 version of Apache Nutch. I referred the following documentation-
https://cwiki.apache.org/confluence/display/NUTCH/NutchTutorial#NutchTutorial-SetupSolrforsearch

The "bin/nutch" binary file corresponds to the Apache Nutch software. By running the "index" command in Nutch, the content from one or more segments is retrieved and then passed to all enabled IndexWriter plugins, which ultimately send the documents to Solr for indexing. Once the solr was setup, using Nutch, the crawled pages were indexed in Apache Solr by utilizing three output folders: "crawldb," "linkdb," and "segments." given by the teammate (Karan) working on crawling task. All these databases were included as input during the index construction process in Solr.

The following command was executed to generate the index for the crawled pages in Solr with the help of Nutch:

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

Once the command is executed, I was able to see the vector space relevance model results which is nothing, but the output given by Apache Solr in JSON format.

I ran the following the command to get the whole data in my local system-
curl "http://localhost:8983/solr/nutch/select?q=":"&wt=json&indent=true&rows=118780" > solr_data.json

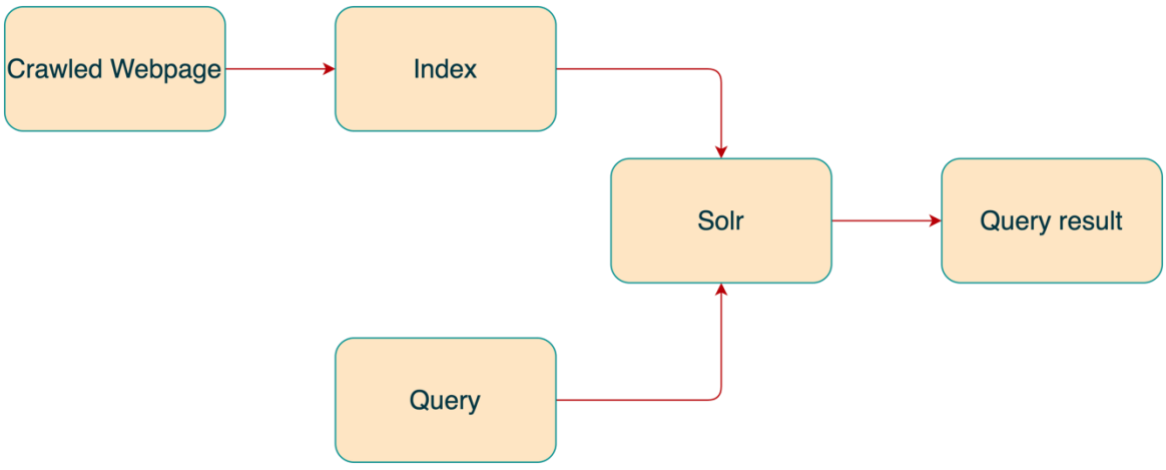


Figure 2: Index creation flow diagram

WebGraph creation and statistics:

WebGraph is an alias for org.apache.nutch.scoring.webgraph

From the crawled URLs, I have created a web graph using nutch command. This WebGraph class creates three databases, one for inlinks, one for outlinks, and a node database that holds the number of inlinks and outlinks to a url and the current score for the url. The WebGraph is an update-able database. Outlinks are stored by their fetch time or by the current system time if no fetch time is available. Only the most recent version of outlinks for a given url is stored. As more crawls are executed and the WebGraph updated, newer Outlinks will replace older Outlinks. This enables WebGraph to adapt to changes in the link structure of the web. The Inlink database is created from the Outlink database and is regenerated when the WebGraph is updated. The Node database is created from both the Inlink and Outlink databases. Because the Node database is overwritten when the WebGraph is updated and holds current scores for urls, a crawl-cycle (one or more full crawls) should be fully complete before the WebGraph is updated.

Statistics

Total Number of links = 123862
Total Number of nodes = 31875
The largest number of ingoing links = 3746
The largest number of outgoing links = 22

For showing webgraph information connected to index, the Webgraph is intended to be a step in the **score calculation** based on the link structure (i.e., webgraph). During the process of creating an index in Solr, the LinkRank program in Nutch was executed to perform a link analysis of the webgraph in an iterative manner. This program is similar to PageRank, and it aims to calculate stable global scores for each URL in the webgraph.

At the start of the LinkRank process, all URLs are assigned a common score. Then, the program calculates a global score for each URL based on two factors: the number of incoming links and their respective scores, and the number of outgoing links from the URL. This means that URLs with a higher number of incoming links from other high-scoring URLs will receive a higher global score. Conversely, URLs with a lower number of incoming links or with incoming links from low-scoring URLs will receive a lower global score. The LinkRank process continues to iterate until stable global scores are obtained for each URL in the webgraph.

Relevance Models

1. Vector Space Relevance model

The vector space relevance model is a commonly used approach for ranking search results in information retrieval systems and is also used in Apache Solr to index crawled pages which is nothing but tf-idf based relevance model.

Here's a detailed explanation of how the vector space relevance model works in Solr:

1. Crawling and Indexing: As with other search engines, Apache Nutch is used to crawl web pages and index them in Solr. During indexing, each document is represented as a bag of words, where the words are the terms that appear in the document.

2. Building the Inverted Index: Solr builds an inverted index of the documents based on the terms that appear in them. The inverted index maps each term to a list of documents that contain that term, along with the frequency of the term in each document.

3. Querying: When a user enters a search query, Solr parses the query and converts it into a vector of query terms, which is used to represent the query in the same way as documents are represented.

4. Calculating the Score: Solr uses the vector space relevance model to calculate a score for each document in the index based on its similarity to the query. The similarity is measured using the cosine similarity between the query vector and the document vector.

5. Ranking: Once the scores have been calculated, Solr ranks the documents in descending order of their scores and returns the top documents as search results.

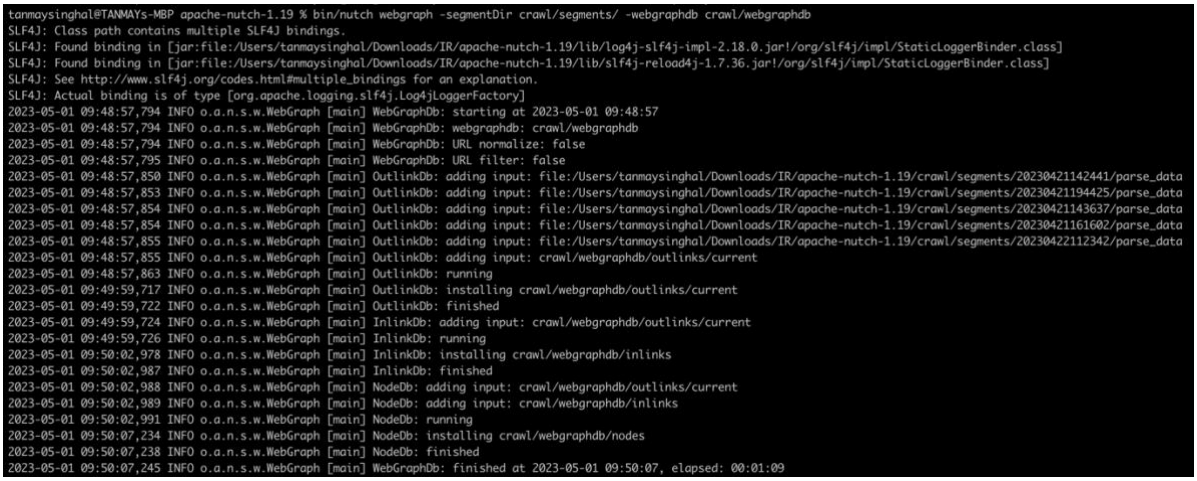
In more detail, the vector space model represents each document and query as a vector of term frequencies. The vector is high-dimensional, with each dimension corresponding to a different term in the vocabulary. The entries in the vector are the frequencies of the corresponding terms in the document or query. The vectors are then normalized to unit length to remove the effect of document length. The similarity between the query vector and the document vector is measured using the cosine similarity measure. This measures the cosine of the angle between the two vectors, which is a measure of their similarity. The higher the cosine similarity, the more similar the document is to the query.

2. PageRank

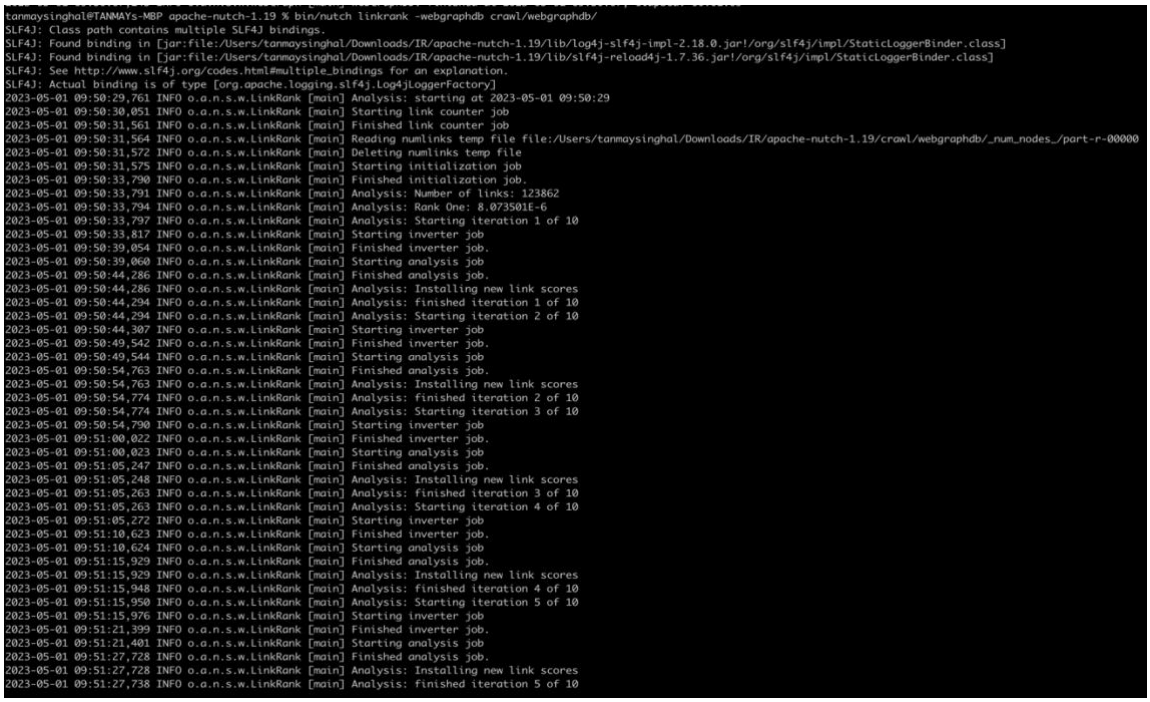
Apache Nutch is an open-source search engine software that is capable of implementing the PageRank algorithm for ranking web pages in search results. Nutch's implementation of the PageRank algorithm starts with a graph-based representation of the web, where each web page is a node and each hyperlink between pages is an edge. The algorithm then uses link analysis techniques to determine the importance of each node in the graph. Nutch's implementation of the PageRank algorithm calculates the PageRank score for each node based on the number and quality of incoming links to the node. The algorithm assigns an initial score of 1 to each node and then iteratively recalculates the scores until they converge to stable values. Once the PageRank scores have been calculated, Nutch uses them to rank web pages in search results. Pages with higher PageRank scores are considered more important and are ranked higher in search results.

Below are the commands used to execute the pagerank algorithm in apache nutch-

a. bin/nutch webgraph -segmentDir crawl/segments/ -webgraphdb crawl/webgraphdb
Created a webgraph from the set of URLs fetched by the crawler

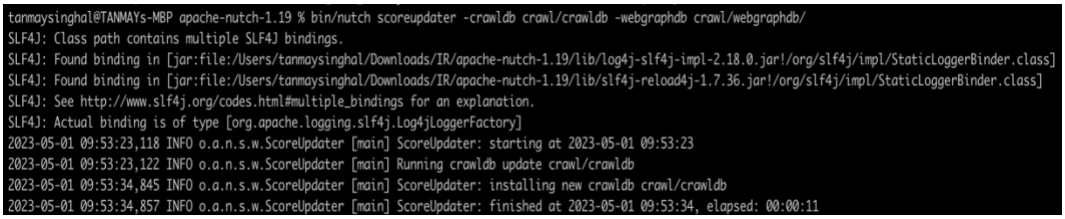


b. bin/nutch linkrank -webgraphdb crawl/webgraphdb/



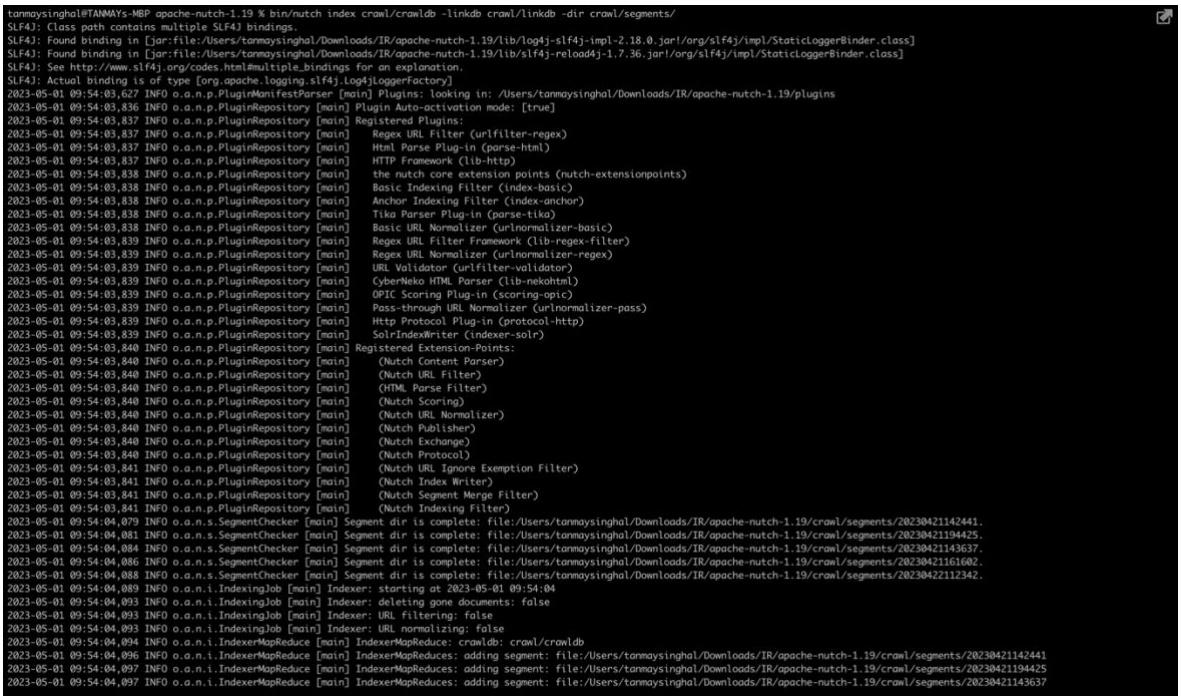
This command linkrank will calculate the score based on the previous structures until the score converges.

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



This command **scoreupdater** will update the score from the webgraph back into the crawldb

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



This command index the updated crawled database.

I used the nodedumper command to get useful data out of the webgraph data, including the actual score of a node and the highest scored inlinks/outlinks. Below is the command-

e. bin/nutch nodedumper -output /Users/tanmaysinghal/Downloads/IR/apache-nutch-1.19/crawl/pagerank_scores -webgraphdb /Users/tanmaysinghal/Downloads/IR/apache-nutch-1.19/crawl/webgraphdb

Example of topic-based page ranking results

1. **Mountains by countries**

Url- https://mountains-connect.org/mountain-range-andes/
Title- ANDES – Mountains Connect
Pagerank score- 0.15000685

Url- https://mountain.org/where-we-work/global-initiatives/
Title- Global Initiatives - Instituto de Montaña
Pagerank score - 0.15000685

2. **Mountains by height**

Url- https://wildlandtrekking.com/destination/north-cascades-hiking-backpacking-trips/
Title- North Cascades Hiking & Backpacking Tours | Wildland Trekking
Pagerank score- 0.15000685

Url- https://www.mwis.org.uk/blog/post/mountains-make-own-weather
Title - Why are the Cairngorms colder than other UK mountains at the moment?
Pagerank score- 0.1500685

3. HITS (Hyperlink Induced Topic Search)

The HITS algorithm, or Hyperlink-Induced Topic Search, is another link analysis algorithm that can be used for ranking web pages in search engine results. The HITS algorithm starts by constructing a web graph, where each web page is a node and each hyperlink between pages is an edge. The graph is created by using the library Network .The graph is constructed using the information from the search engine's index. The HITS algorithm calculates two scores for each node in the web graph: an authority score and a hub score. The authority score measures the importance of a node based on the number and quality of incoming links to it, while the hub score measures the importance of a node based on the number and quality of outgoing links from it. The HITS algorithm iteratively recalculates the authority and hub scores for each node in the graph. The algorithm starts by assigning an initial score of 1 to each node and then iteratively recalculates the scores until they converge to stable values. Once the authority and hub scores have been calculated, the HITS algorithm uses them to rank web pages in search results. Pages with high authority scores are considered more important and are ranked higher in search results, while pages with high hub scores are good sources of information and may be useful as additional search results.

Overall, the HITS algorithm provides an alternative way of ranking web pages in search results based on the importance of nodes in the web graph. While the PageRank algorithm considers only the importance of incoming links, the HITS algorithm considers both incoming and outgoing links to determine the relevance and quality of web pages.

We used the bin/nutch **readlinkdb** link to receive the dump of linkdb into a text file which also contains specific information related to specific URL.

Among the web pages, we crawled,

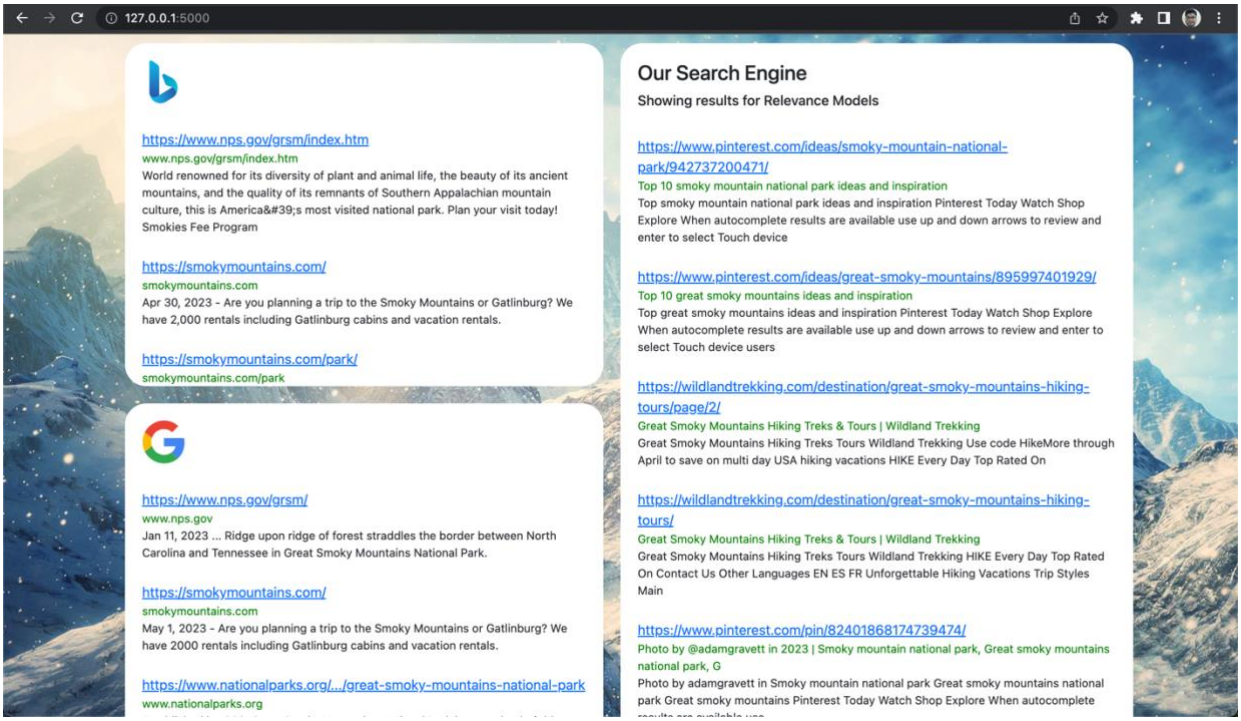
1. Highest hub score was for: https://www.mountainiq.com/africa/
   Score - 1.1796536393492583e-06
2. Highest authority score was for: https://www.mountainiq.com/africa
   Score - 1.1796536393492583e-06

Collaboration

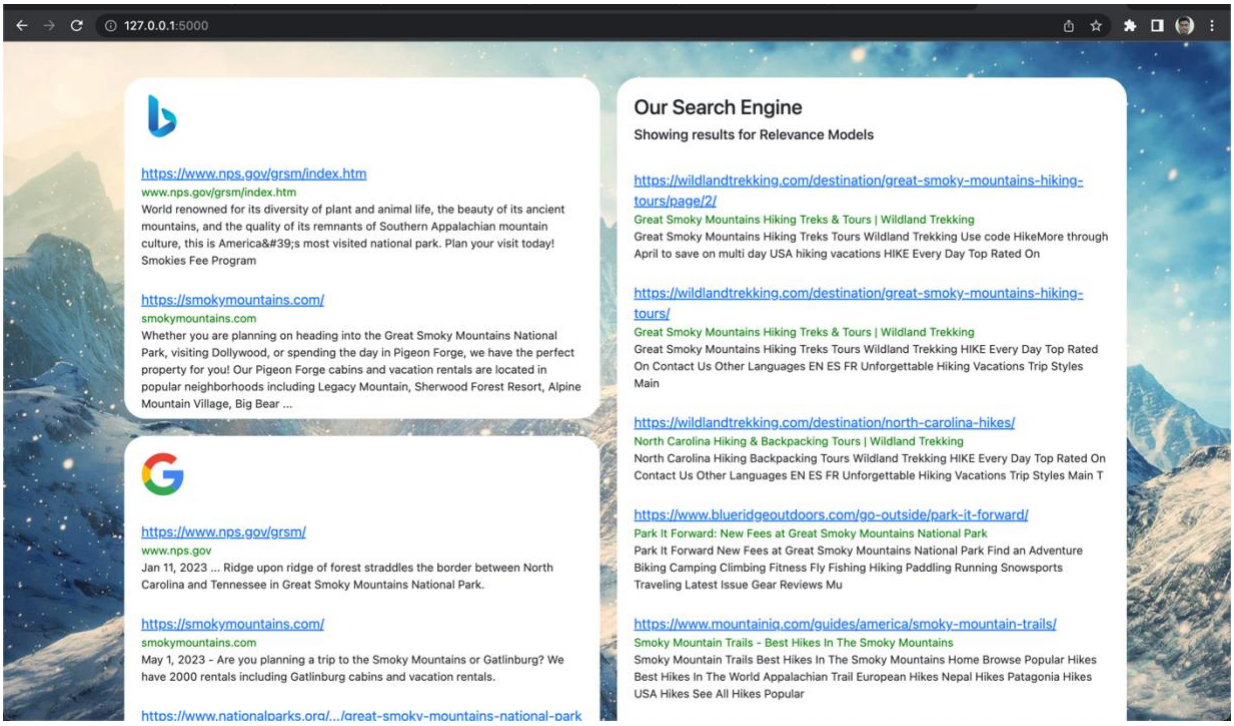
I collaborated with Anirudh (the person responsible for task 3 – UI) to test the several queries generated for evaluating the effectiveness of the relevance models implemented in our search engine. I generated 2 queries and tested on UI by selecting HITS and PageRank options. I have judged the results in ways that when selecting HITS, the results were more relevant as compared to when selecting PageRank. Here is the example-

Query1 - smoky mountains

Below is the screenshot when we selected pagerank relevance model.



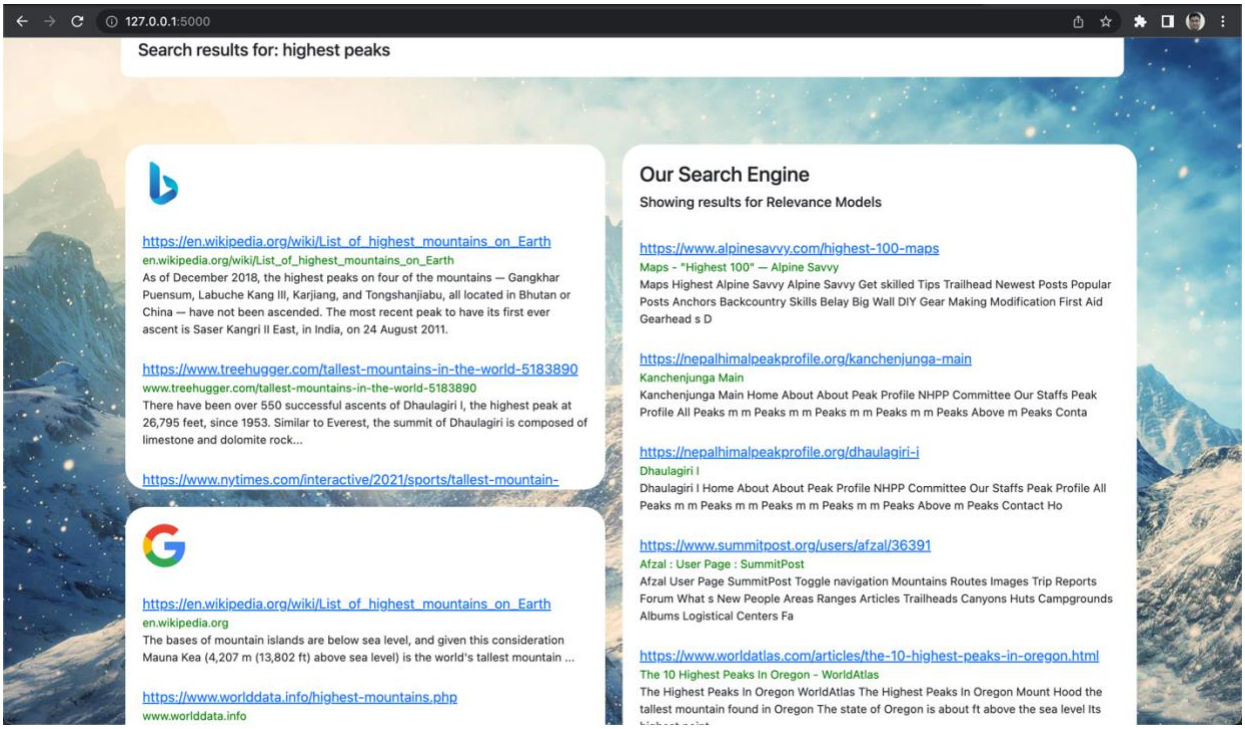
Below is the screenshot when we selected HITS relevance model.



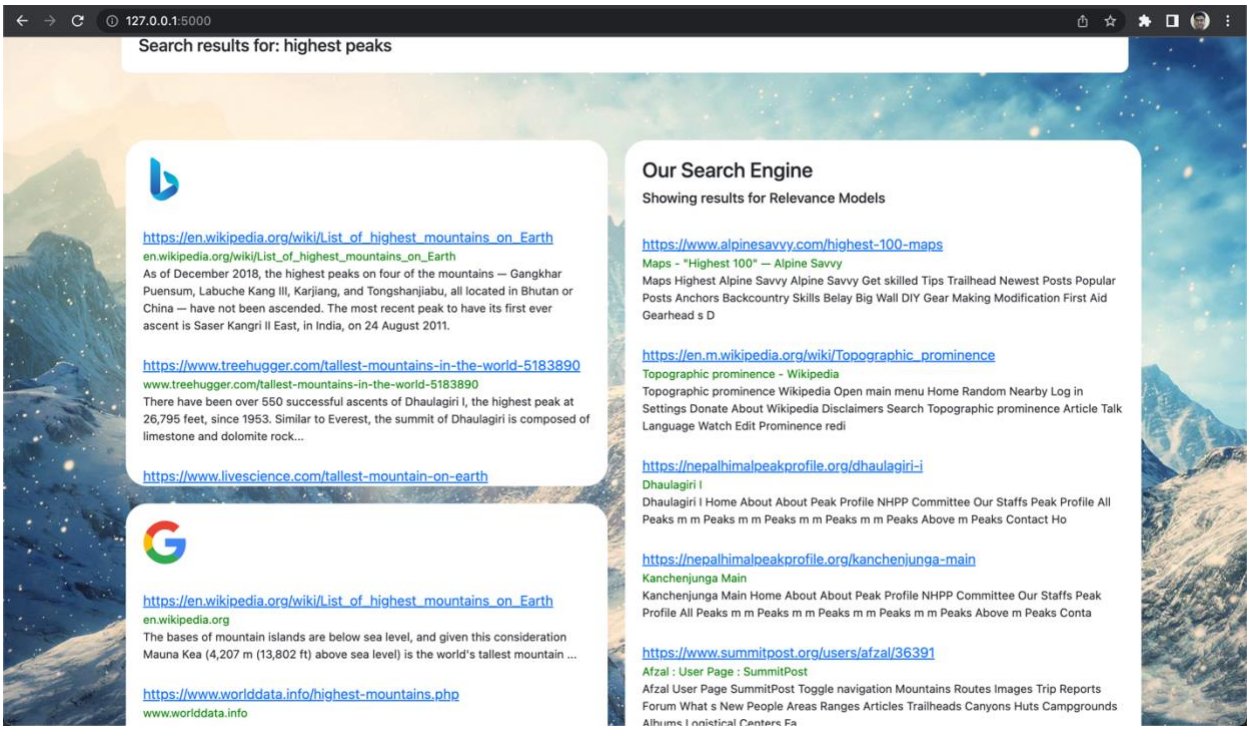
Observation - We can see from the above results HITS is giving more relevant results as compared to PageRank model. PageRank is giving some Pinterest URLs while HITS is giving about trekking URLs to smoky mountains.

Query2 – highest peaks

Below is the screenshot when selected PageRank option.



Below is the screenshot when selected HITS option.



Observation – We can see from the above screenshots that HITS is giving Wikipedia results on the top while PageRank doesn't even show it.

I collaborated with Prem (the person responsible for task 4- clustering) to enhance the effectiveness of the relevance models and has found that clustering has improved the search results significantly compared to both the PageRank and HITS relevance models for the above queries.

Anirudh Kiran AXK200227

User Interface and Comparisons with Google and Bing

The user interface for our search engine is developed using HTML, CSS, Python and Flask. I had a simple goal in mind while developing the webpage – to provide the users a simple, intuitive way of looking up information about mountains.

I used HTML to design the general layout of the webpage and CSS to make it more appealing. A search bar at the center of the page makes it easy for anyone to enter a search query. A collection of dropdowns enables you to choose one of several options based on which query results are displayed.

The query results are divided into three sections on the same page – one section each for Bing and Google results and one section for the results we have generated.

As mentioned by Tanmay, Solr returns the indexed data along with the results of the vector relevance model in JSON format. All I had to do was to develop an API to retrieve this data from Solr. I then had to write a function to parse the JSON to get only relevant fields such as the URL, title and content. In some cases, the function had to be modified to include the PageRank as well as the HITS scores. To check the correctness of this function and to ensure proper communication between Tanmay's relevance model results and the user interface, we tested around 15 queries. In addition to this, I tested about 20 queries to ensure that the user interface can handle inputs of different option types and display the appropriate results on the webpage.

Working of the user interface:
When a user enters a query and chooses any one of the options, the query along with the option is sent to the Flask backend. Based on the option chosen, I call the appropriate functions to generate results. These results are then sent to the front-end using the render_template() function of Flask. I used the *Jinja* markup offered by Flask to easily display the results on the webpage.

The different options that a user can choose are as follows:

1. Relevance models:
    a. Vector Space
    b. PageRanking
    c. HITS

    For these options, I wrote code that interacts directly with Solr to fetch relevance.
    results, page rank and the HITS score

2. Clustering models:
    a. Flat Clustering
    b. Single-link Agglomerative Clustering
    c. Complete-link Agglomerative Clustering

Similar to the relevance model option, I had to call functions to generate results and clusters. Additional code had to be written to mention the cluster on the webpage.

3. Query Expansion methods:
    a. Association Clustering
    b. Scalar Clustering
    c. Metric Clustering
    d. Query Expansion Rocchio

e. Rocchio Algorithm

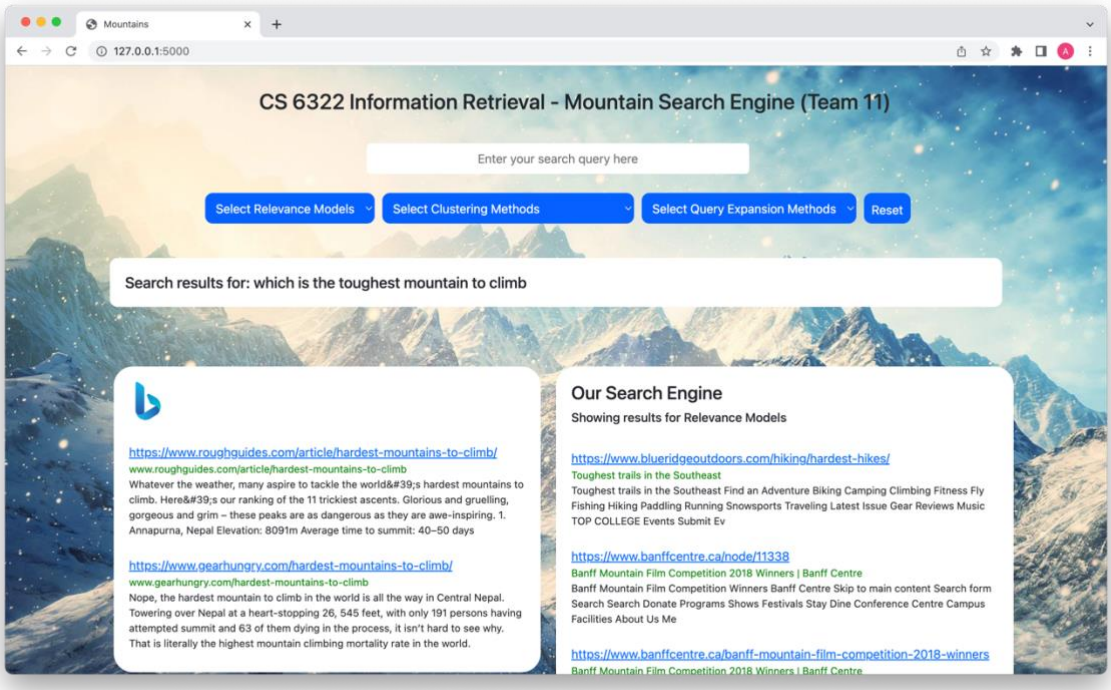
Results for each of these options were displayed on the webpage along with the expanded query.

Comparing our results with Google and Bing:

While comparing results, we need to note that Google and Bing collectively index about 15 to 30 billion websites. We, however, indexed around 100,000 websites. Given these statistics, our results are not always the most accurate. This is because we do not have the processing power of large search engines, nor do we have a way of dynamically updating our results based on the results the people usually lean towards.

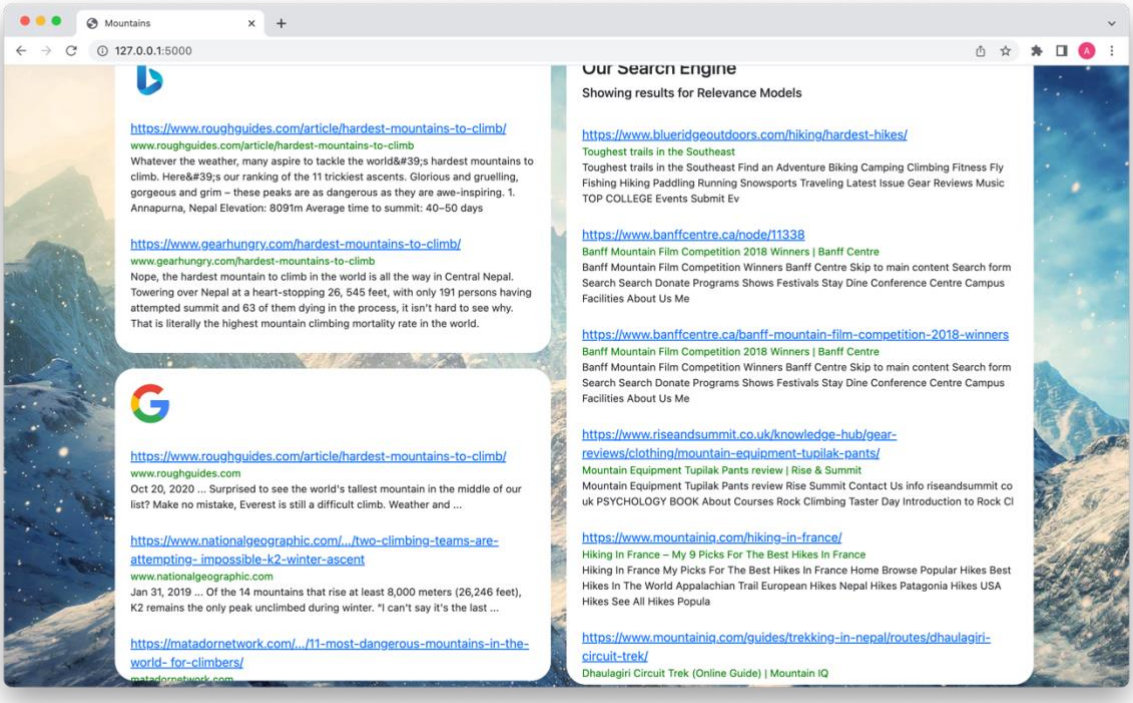
For the demonstration, we have chosen those queries that were showing the most accurate results after running clustering and query expansion.

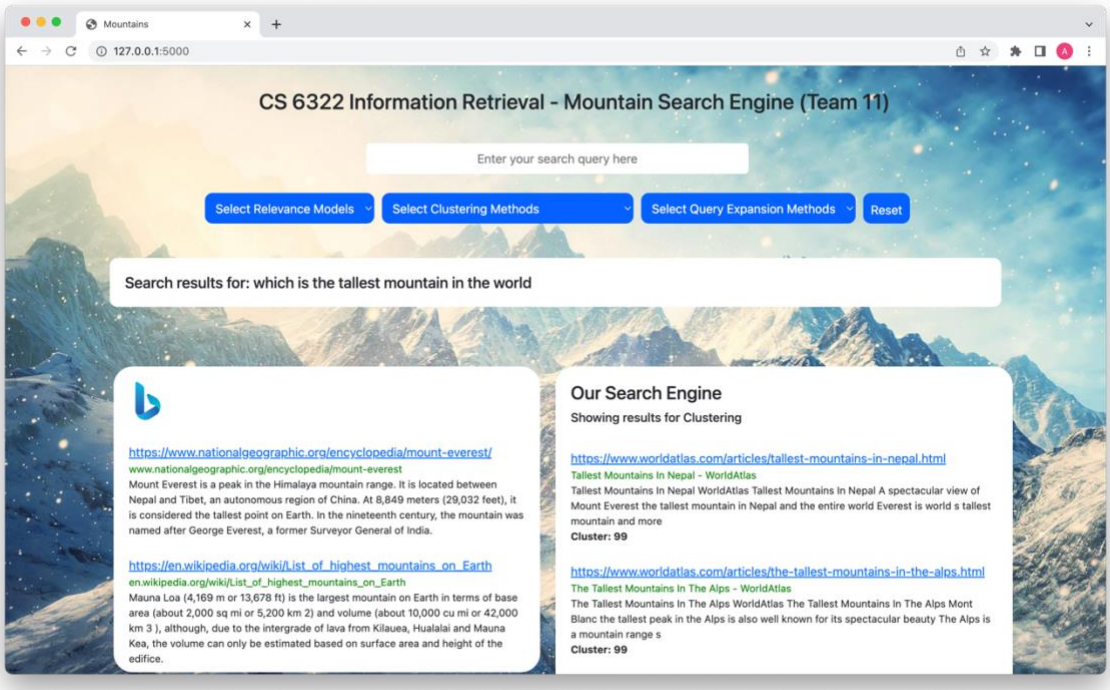
Our results compared to Google and Bing when we choose Relevance models:



Anirudh Kiran AXK200227

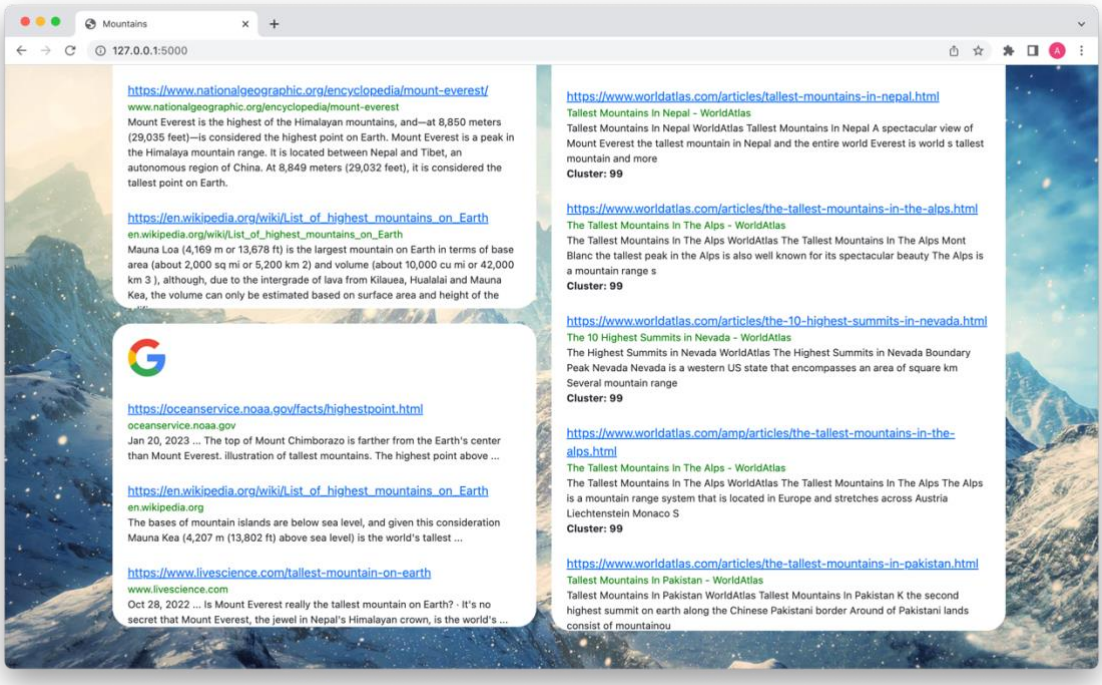
Our results compared to Google and Bing when we choose Clustering models.

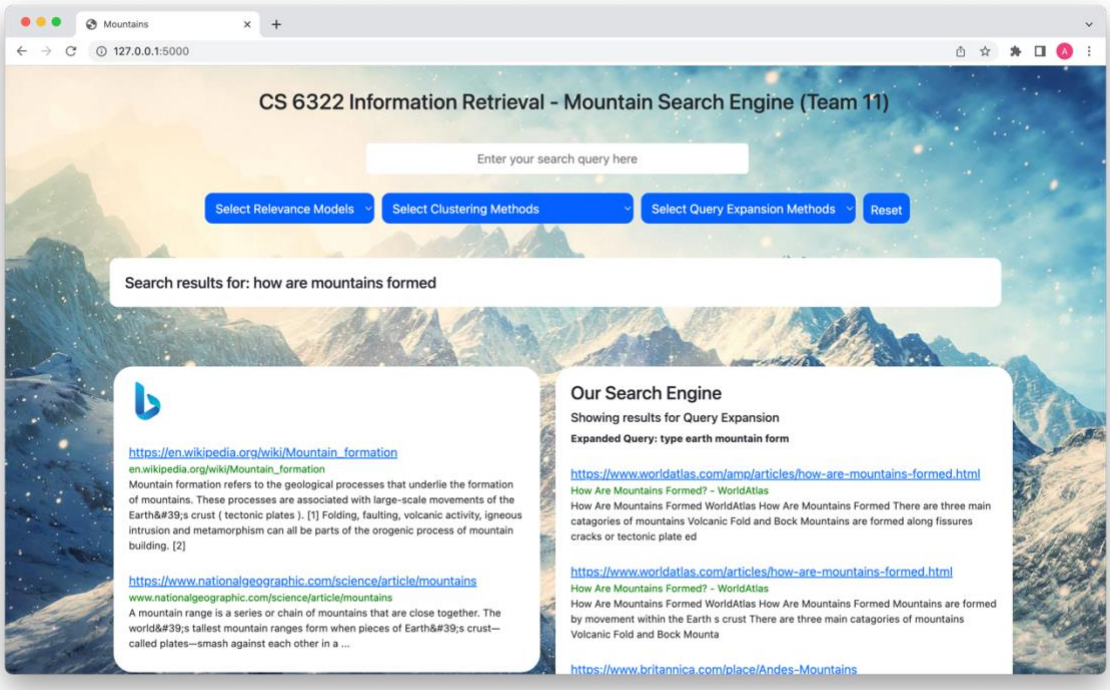


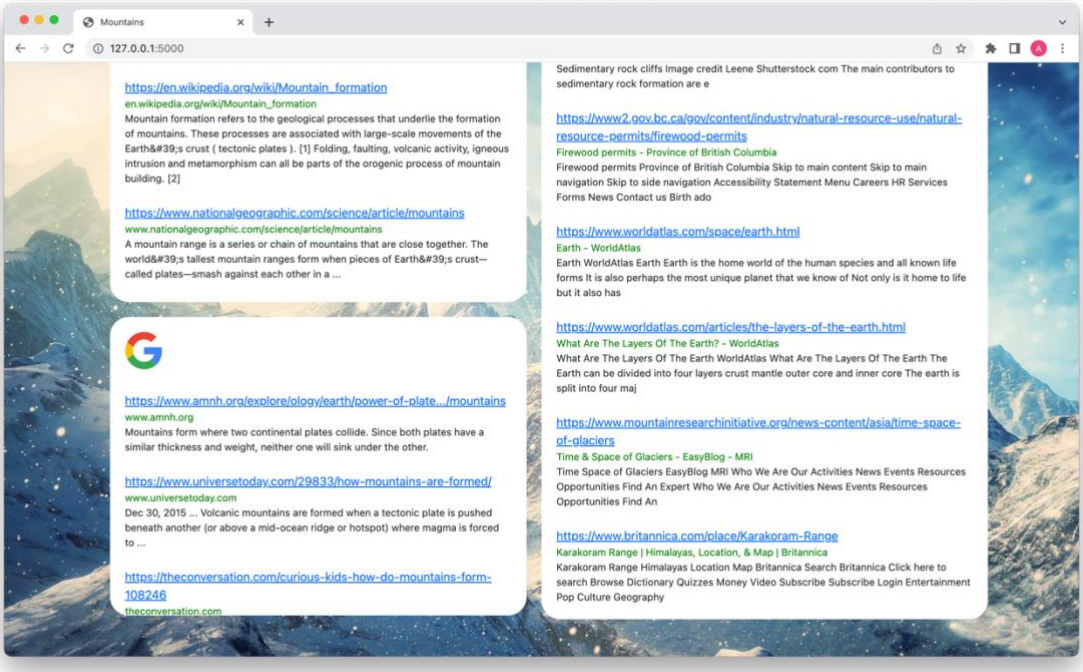


Anirudh Kiran AXK200227

Our results compared to Google and Bing when we choose Query Expansion:






Mountains
127.0.0.1:5000
https://en.wikipedia.org/wiki/Mountain_formation
en.wikipedia.org/wiki/Mountain_formation
Mountain formation refers to the geological processes that underlie the formation of mountains. These processes are associated with large-scale movements of the Earth's crust (tectonic plates). [1] Folding, faulting, volcanic activity, igneous intrusion and metamorphism can all be parts of the orogenic process of mountain building. [2]
https://www.nationalgeographic.com/science/article/mountains
www.nationalgeographic.com/science/article/mountains
A mountain range is a series or chain of mountains that are close together. The world's tallest mountain ranges form when pieces of Earth's crust—called plates—smash against each other in a ...
https://www.amnh.org/explore/ology/earth/power-of-plate.../mountains
www.amnh.org
Mountains form where two continental plates collide. Since both plates have a similar thickness and weight, neither one will sink under the other.
https://www.universetoday.com/29833/how-mountains-are-formed/
www.universetoday.com
Dec 30, 2015 ... Volcanic mountains are formed when a tectonic plate is pushed beneath another (or above a mid-ocean ridge or hotspot) where magma is forced to ...
https://theconversation.com/curious-kids-how-do-mountains-form-108246
theconversation.com
Sedimentary rock cliffs Image credit Leene Shutterstock com The main contributors to sedimentary rock formation are e
https://www2.gov.bc.ca/gov/content/industry/natural-resource-use/natural-resource-permits/firewood-permits
Firewood permits - Province of British Columbia
Firewood permits Province of British Columbia Skip to main content Skip to main navigation Skip to side navigation Accessibility Statement Menu Careers HR Services Forms News Contact us Birth ado
https://www.worldatlas.com/space/earth.html
Earth - WorldAtlas
Earth WorldAtlas Earth Earth is the home world of the human species and all known life forms It is also perhaps the most unique planet that we know of Not only is it home to life but it also has
https://www.worldatlas.com/articles/the-layers-of-the-earth.html
What Are The Layers Of The Earth? - WorldAtlas
What Are The Layers Of The Earth WorldAtlas What Are The Layers Of The Earth The Earth can be divided into four layers crust mantle outer core and inner core The earth is split into four maj
https://www.mountainresearchinitiative.org/news-content/asia/time-space-of-glaciers
Time & Space of Glaciers - EasyBlog - MRI
Time Space of Glaciers EasyBlog MRI Who We Are Our Activities News Events Resources Opportunities Find An Expert Who We Are Our Activities News Events Resources Opportunities Find An
https://www.britannica.com/place/Karakoram-Range
Karakoram Range | Himalayas, Location, & Map | Britannica
Karakoram Range Himalayas Location Map Britannica Search Britannica Click here to search Browse Dictionary Quizzes Money Video Subscribe Subscribe Login Entertainment Pop Culture Geography

Prem Sharma PXS210046

Clustering

Flat Clustering

Clustered the Web pages indexed by Tanmay. K-Means clustering has been used for flat clustering. Single-Link and Complete-Link clustering have been used as agglomerative clustering methods. The clusters that were obtained have been used for improving the relevance. Collaborated with Karan to obtain the Web crawl, and with Tanmay and Anirudh to generate experiments that showcase how the clustering information is used to enhance the relevance results. Considered 50 queries in these experiments.

K-means clustering algorithm:

Given a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_n$), where each observation is a d-dimensional real vector, k- means clustering aims to partition the n observations into k (≤ n) sets $\mathbf{S}$ = {S_1, S_2, ..., S_k} 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:

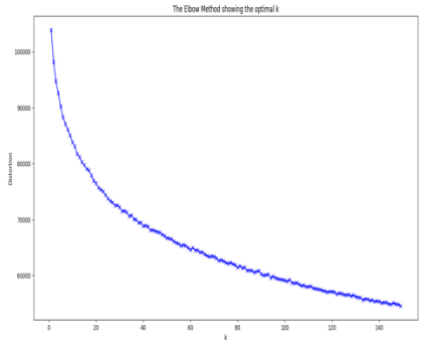
$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

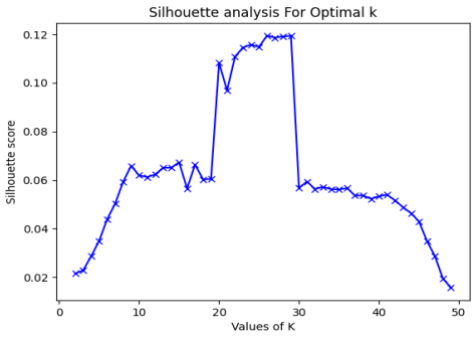
where $\boldsymbol{\mu}_i$ is the mean of points in S_i.

Selection of predefined clusters (value of k):

Internal Methods of Evaluation:

On a sample of the dataset, values of k ranging from 1 to 150 were examined, and the observed clusters were compared to the relevancy of the documents. Both the elbow approach and silhouette score methods were employed to get a suitable value for k.
The silhouette score method gave a more accurate result of 30. The peak was found to be 30. After analyzing the outcomes of various k values, it was discovered that k = 30 produced the most relevant results. As a result, all URLs retrieved are kept in one of the 30 clusters.
.





External Methods of evaluation:
Each cluster was assigned to the class which was most frequent in the cluster. For the values of K ranging from 1-50. The NMI scores and purity values were calculated. The purity value was around 0.6 for k = 30 much more than any other value of K. NMI score was also relatively higher for k at 30. Both of these methods resulted in good results for value of K as 30. Hence, 30 was selected to be the optimal value for K.

What did you do with the results of clustering – did you incorporate them in the relevance models – and did you provided to the user interface results that were obtained when clustering is used (8 points)?

Flat Clustering Algorithm Design & Implementation

Karan, the student in charge of crawling, indexed all the crawled documents, while Tanmay, the student responsible for indexing, provided the response returned by Apache Solr when queried with "*". This query returned all crawled documents in JSON format, which were saved to a file that was 1.5 GB in size.
The plain text content of each document, without the DOM and HTML tags, was extracted from the JSON, along with its corresponding URL. The content values of all crawled documents were saved in a list, which was then vectorized using TfidfVectorizer from scikit-learn.
The K-means clustering algorithm from Scikit-KMeans was applied to the vectorized data, resulting in 30 clusters, each consisting of several URLs. Later, a Pandas dataframe was created for URLs and their corresponding clusters, which were then saved to a file.
The results were stored in a text file in the following format: URL, ClusterNumber.

A sample of the file is:

https://www.summitpost.org/sugarloaf-mountain-at-sunset/261217,3.0
https://www.summitpost.org/suggested-mountains/dow-williams/19503/p7,3.0
https://www.summitpost.org/suggested-mountains/erica-n/36326,3.0
https://www.summitpost.org/suggested-mountains/lodewijk/,3.0
https://www.summitpost.org/suggested-mountains/lodewijk/26969/p3,3.0
https://www.summitpost.org/suggested-mountains/lodewijk/26969/p50,3.0
https://www.summitpost.org/suggested-mountains/lodewijk/26969/p6,3.0

Clustering results incorporation and integration with UI

- On the UI, there are 3 clustering modes: flat, single-link and complete link hierarchical clustering.
- If any of the clustering modes is selected, the GET request contains that clustering mode in the type of parameter.
- Solr fetches the top 50 relevant results based on page ranking and HITS score and sends it to cluster computation handler.
- Inside the clustering computation handler, the query is clustered along with the relevant results following the same K-Means steps.

- The URL with the highest cosine similarity is fetched and its cluster number among the clusters is identified.
- The results that belong to that cluster are shown first, all the other results belonging to the same cluster in the order of decreasing page rank scores are ranked next.
- The process repeats for the next highest ranked result and all the pages belonging to its cluster are ranked next.
- This process repeats till we reach our threshold of 50.
- The updated ranks are then sent as a response to the UI.

Agglomerative Clustering

For Agglomerative clustering of texts, single-link and complete-link agglomerative clustering algorithms are used.

Agglomerative Clustering Algorithm Design & Implementation

All the documents crawled by Karan (the student responsible for crawling) were indexed by Tanmay (the student responsible for indexing). Tanmay gave me 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 to a file. The size of this file was 1 GB.

- I extracted the "content" key's value from the JSON, which contained the content of the document in plain text format, removing the DOM and HTML tags.
- URL corresponding to the content was also extracted.
- The "content" value from all the crawled documents was stored in a list.
- The content was later vectorized using scikit-learn's TfidfVectorizer.
- The TF-IDF vectorizer creates a matrix representation of the document text.
- Truncated SVD is then applied to it for dimensionality reduction and normalization
- Vectorizer's output was converted to match agglomerative clustering algorithm's input.
- Python's fastcluster library is used to efficiently compute the results of hierarchical clustering.
- Firstly, linkage is calculated, and then the single method of fastcluster library is called to compute the clusters to which a URL belongs to.
- A dendrogram is plotted as well to show the results of clustering visually.
- The labels are extracted by looping over the dictionary returned by the dendrogram function. The labels correspond to the URLs of the documents. The labels are then used to create a dictionary that maps each URL to its corresponding cluster number.
- Finally, the script stores the clustering results in a file. The results are stored in a **pandas** Data Frame with columns for the document IDs and cluster numbers. The **to_csv** function is used to write the Data Frame to a text file.

Single Link Agglomerative Clustering

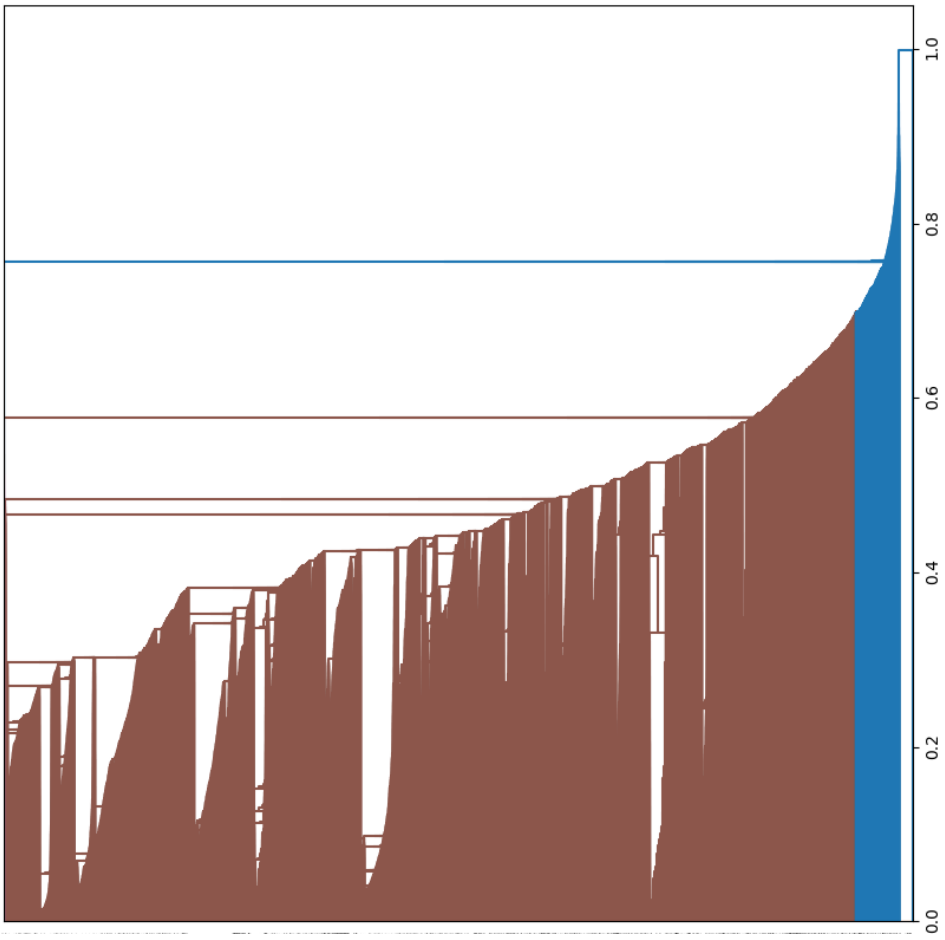
Number of clusters obtained.

The number of clusters is determined by the hierarchical clustering algorithm itself. Specifically, the **fastcluster.linkage** function is used to perform the hierarchical clustering using single linkage method, which produces a dendrogram representing the hierarchical relationships between the data points. The dendrogram can then be visually inspected to determine an appropriate number of clusters based on the shape and structure of the dendrogram. The number of clusters were determined by finding the maximum vertical distance in the dendrogram and passing a horizontal line in the middle. The optimal number of clusters will be the number of vertical lines intersecting it. 10 clusters were constructed for our dataset, with all the URLs falling into one of these clusters. They were represented on the user interface by marking the cluster number once the results are fetched.

A sample of the output file is shown below:

https://alpineclubofhimalaya.com/trip/everest-circuit-three-high-passes-5-summits/,6.0
https://alpineclubofhimalaya.com/trip/everest-base-camp-leisurely-trek/,6.0
https://alpineclubofhimalaya.com/trip/langtang-cultural-tour-with-gosainkunda-10-days/,6.0
https://alpineclubofhimalaya.com/trip/annapurna-base-camp-short-trek/,6.0
https://alpineclubofhimalaya.com/trip/ama-dablam-base-camp-trek/,6.0
https://alpineclubofhimalaya.com/trip/namche-bazaar-trek-and-helicopter-tour/,6.0
https://alpineclubofhimalaya.com/trip/tengboche-monastery-trek/,6.0
https://alpineclubofhimalaya.com/trip/everest-base-camp-and-gokyo-lake-trek-19-days/,6.0
https://alpineclubofhimalaya.com/trip/ama-dablam-base-camp-trek-9-days/,6.0
https://alpineclubofhimalaya.com/trip/gokyo-valley-trek-12-days/,6.0
https://alpineclubofhimalaya.com/trip/kanchenjunga-base-camp-trek/,6.0
https://alpineclubofhimalaya.com/trip/dhaulagiri-pass-trek/,6.0
https://alpineclubofhimalaya.com/trip/dhaulagiri-base-camp-trek/,6.0
https://alpineclubofhimalaya.com/trip/dhaulagiri-circuit-trek-19-days/,6.0
https://alpineclubofhimalaya.com/trip/namche-bazaar-trek/,6.0
https://alpineclubofhimalaya.com/trip/panchase-trek/,6.0
https://alpineclubofhimalaya.com/trip/nar-phu-valley-trek/,6.0
https://alpineclubofhimalaya.com/trip/scenic-flight-to-mt-everest/,6.0
https://alpineclubofhimalaya.com/trip/annapurna-helicopter-tour/,6.0
https://alpineclubofhimalaya.com/trip/australian-camp-trek/,6.0
https://alpineclubofhimalaya.com/trip/langtang-valley-trek-2/,6.0
https://alpineclubofhimalaya.com/trip/annapurna-base-camp-trek-via-poon-hill/,6.0
https://alpineclubofhimalaya.com/trip/everest-view-trek/,6.0
https://alpineclubofhimalaya.com/trip/jhomolhari-trek/,6.0
https://alpineclubofhimalaya.com/trip/everest-base-camp-luxury-trek/,6.0

The results are grouped based on mountain names. As we can see results related to alpine club of himalayas are grouped together as shown above. The dendrogram obtained using single-link agglomerative clustering is shown below. It is generated based on the URLs.



Complete Link Agglomerative Clustering

Number of clusters obtained.

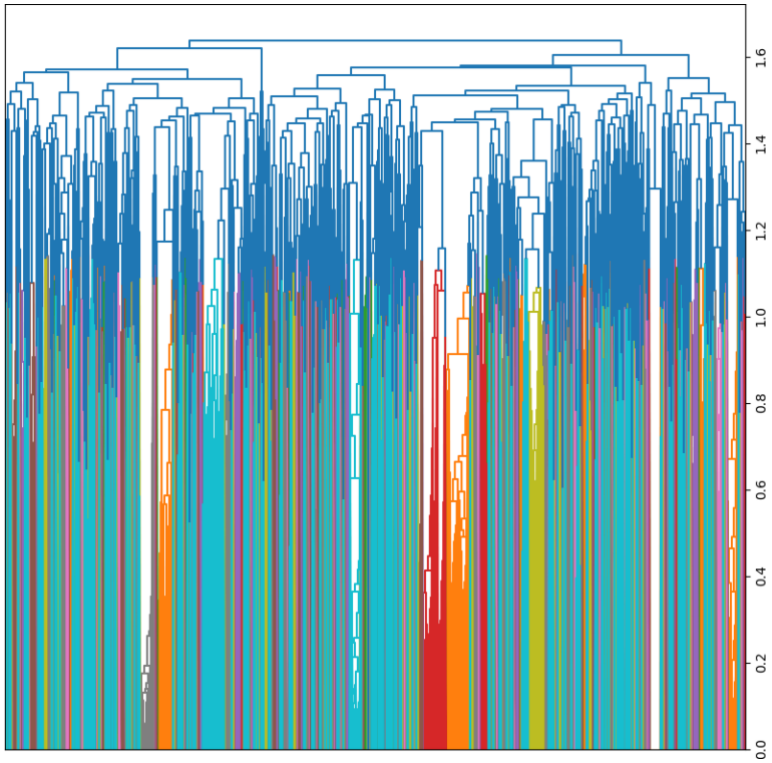
The number of clusters is determined by the hierarchical clustering algorithm itself. Specifically, the **fastcluster.linkage** function is used to perform the hierarchical clustering using the complete linkage method, which produces a dendrogram representing the hierarchical relationships between the data points. The dendrogram can then be visually inspected to determine an appropriate number of clusters based on the shape and structure of the dendrogram. The number of clusters were determined by finding the maximum vertical distance in the dendrogram and passing a horizontal line in the middle. The optimal number of clusters will be the number of vertical lines intersecting it. 10 clusters were constructed for our dataset, with all the URLs falling into one of these clusters. They were represented on the user interface by marking the cluster number once the results are fetched.

A sample of the output file is shown below: -

https://www.mountain-forecast.com/maps/Barbados/swell-2/12,2.0
https://www.mountain-forecast.com/maps/Bahrain/swell-2/12,2.0
https://www.mountain-forecast.com/maps/Arizona/swell-2/12,2.0
https://www.mountain-forecast.com/maps/Arkansas/swell-2/12,2.0

https://www.mountain-forecast.com/subranges/midwest-great-lakes-area-1/locations,2.0
https://www.mountain-forecast.com/subranges/bedded-range/locations,2.0
https://www.mountain-forecast.com/subranges/mayacamas-mountains/locations,2.0
https://www.mountain-forecast.com/subranges/sawtooth-mountains-1/locations,2.0
https://www.mountain-forecast.com/subranges/northern-anatolia-black-sea/photos,2.0
https://www.mountain-forecast.com/maps/United-States/swell-2/12,2.0
https://www.mountain-forecast.com/maps/AlaskaUnitedStates/swell-2/90,2.0
https://www.mountain-forecast.com/maps/AlaskaUnitedStates/swell-2/144,2.0
https://www.mountain-forecast.com/maps/United-States/swell-2/6,2.0
https://www.mountain-forecast.com/maps/AlaskaUnitedStates/swell-2/138,2.0
https://www.mountain-forecast.com/subranges/greek-islands/photos,2.0

The dendrogram below represents the clusters formed among the URLs using complete-link agglomerative clustering:



Clustering results incorporation and integration with UI

The process for hierarchical clustering is like that of flat clustering. In the user interface, the "type" parameter in the request query can be set to one of three values: "Flat Clustering", "Single-link Clustering", or "Complete-link Clustering". The chosen clustering mode determines which clusters are used to re-compute the rankings of the retrieved pages. In the case of Agglomerative Clustering, cosine similarity is again used to determine the cluster which is closest to the given query. The URL of the centroid of that cluster will be fetched. The same URL is used to fetch the cluster number from the pre-computed clusters. The results that belong to that cluster are shown first, all the other results belonging to the same cluster in the order of decreasing page rank scores are ranked next. The process repeats for the next highest ranked result and all the pages belonging to its cluster are ranked next. This process repeats till we reach our threshold of 50. The updated ranks are then sent as a response to the UI. The UI added a cluster number to each result.

Number of queries experimented with to improve results

To improve the findings, around 50 queries were employed for all clustering algorithms. In a subsequent subsection, some of the results obtained after running these queries will be listed. Below the screenshots are observations that highlight the impact of clustering on improving the relevancy of the obtained results.

Query testing generation and impact on results and relevancy

To assess the influence of the results of each clustering approach, I utilized 50 queries for each of the three clustering methods. The questions were created by hand, based on the initial results from the original relevancy model, which included page rank and HITS. The clustering results are relevant, as related results are grouped together and displayed earlier than non-relevant results.

Selection of queries

The criteria for query selection are based on the observations made from the types of clusters created by both clustering algorithms. One of the query selection criteria, as indicated in the sample subset of the results obtained above, is continents. Another criterion for selection was the names of mountains. Queries were also selected based on common search terms which include facts about mountains. Various popular mountaineers were also selected as query terms.

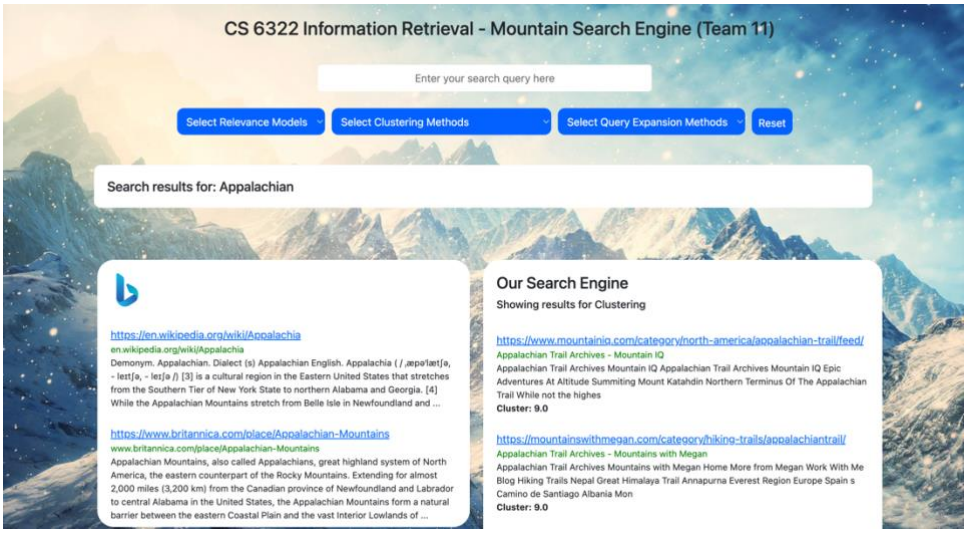
Query examples with clustering

50 queries were tested for flat clustering, single-link agglomerative clustering and complete-link agglomerative clustering. I have included the results for three of these inquiries below. The first graphic depicts the outcomes of using page rank. The outcomes of using flat clustering are shown in the second image. The outcomes of using single-link agglomerative- clustering are shown in the third image and the results of using complete-link agglomerative clustering are shown in the fourth image.

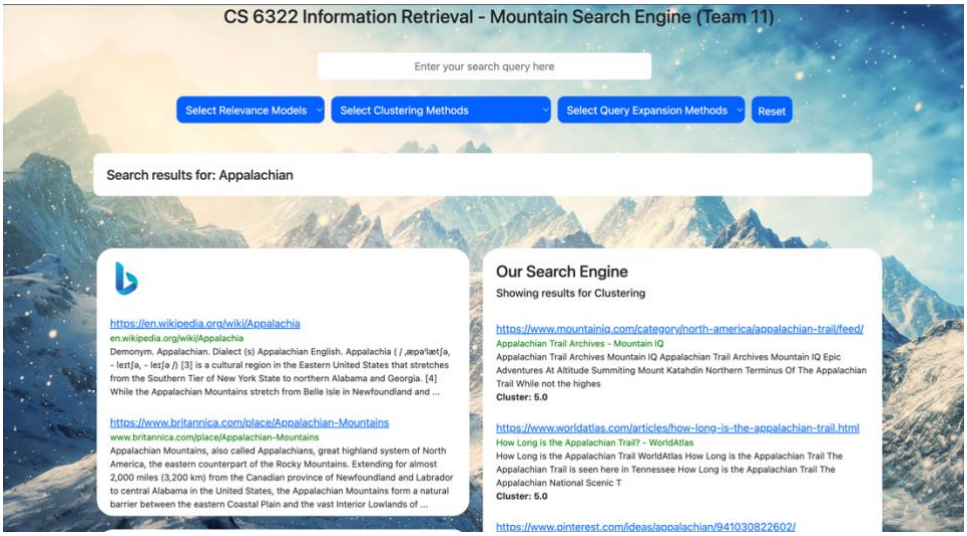
Query1: Appalachian

Observations: On searching for Appalachian. It was observed that page rank was providing some Pinterest URLs. But Flat Clustering was providing more relevant results to it. Clustering handles such details and shows Appalachian results. Such results do not get a top rank when page rank is used.

For Flat Clustering:

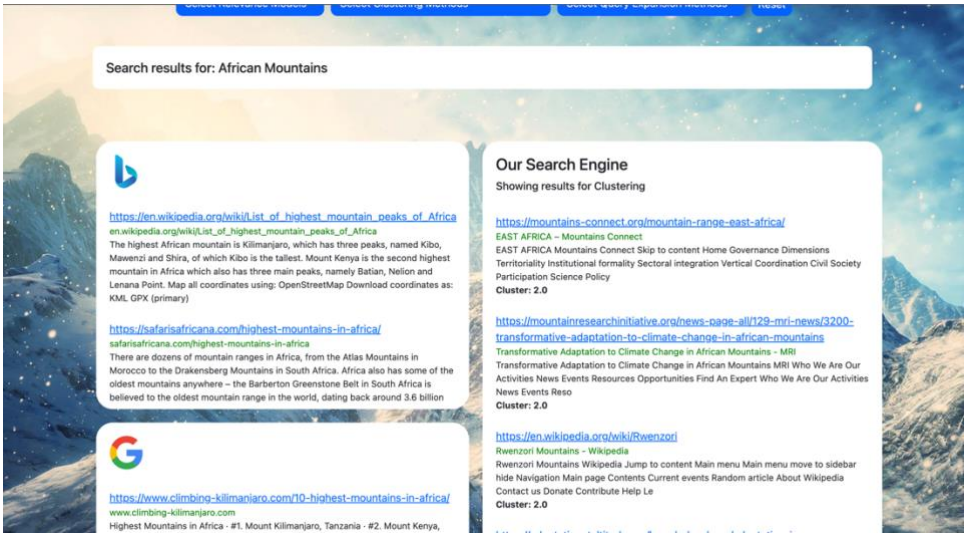


For Single-Link Agglomerative Clustering:

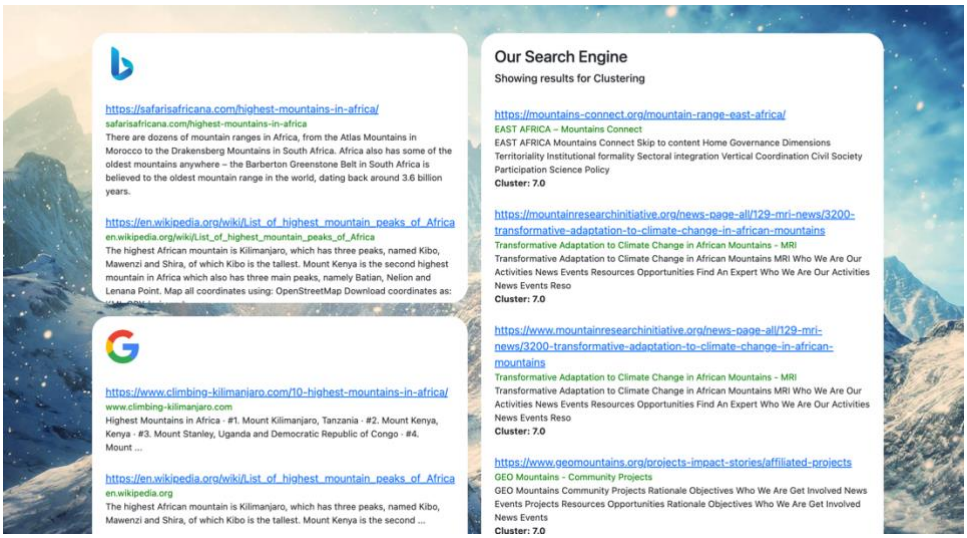


Query2: African Mountains

Observations: With this query, the third result that is fetched through flat clustering is the wikipedia page for Rwenzori Mountains, which is a mountain range in Africa. These results are further down in vector space relevance model.

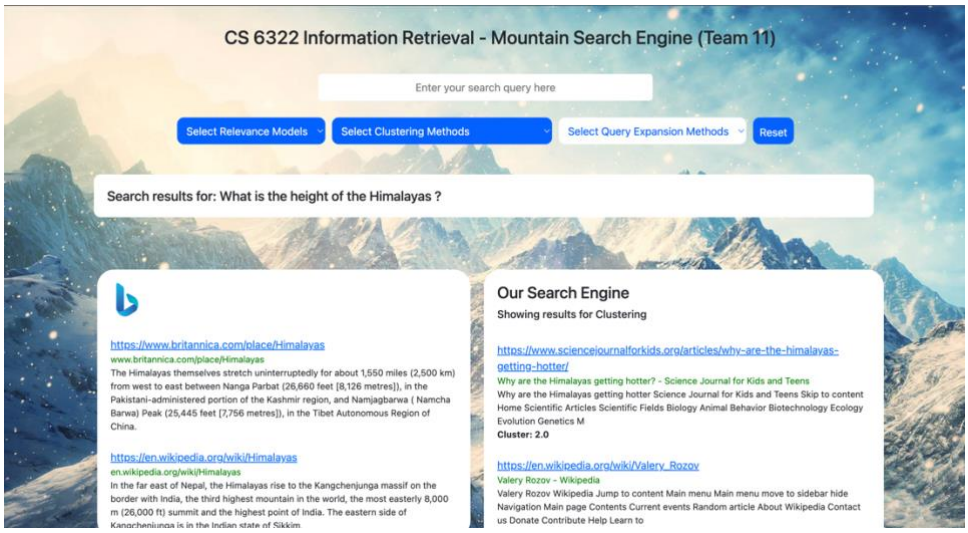


For Complete-Link Agglomerative Clustering:

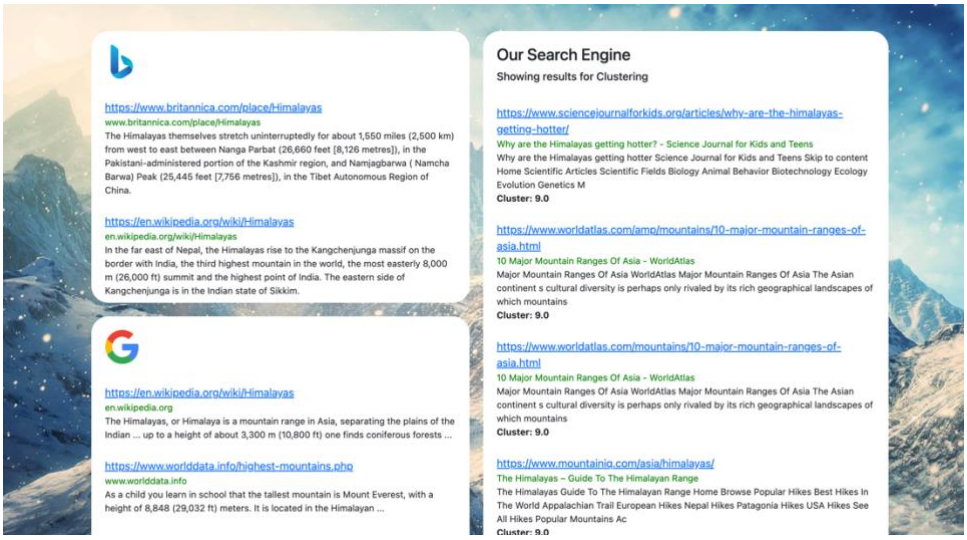


Query3: What is the height of The Himalayas ?

Observations: This query retrieves a page related to The Himalayas using clustering, while it retrieves a page related to Kanchenjunga using page ranking.



For Complete-Link Agglomerative Clustering:



Arpit Singh AXS210204

Query Expansion and Pseudo Relevance Feedback

1. Query Expansion with Rocchio Algorithm

I randomly generated 20 queries for our mountains search engine with the help of chat GPT which gave me a mix of queries having keywords in the range of 2-5.

Also instead of using stems, I have used unique lemmas in this algorithm because it produced better results as compared to stems.

1. Tallest mountains in the world
2. Famous mountain ranges
3. Mountains for hiking
4. Himalayan mountain range
5. Volcanic mountains
6. Mountain biking trails
7. Mountain resorts
8. Mountain cabins
9. Mountain animals
10. Tallest peaks in North America
11. Mountains in america with the best sunsets
12. Mountain weather forecast
13. Mountain ranges in Europe
14. Oldest mountains ranges in the world
15. Highest peak in South America
16. Mountains for skiing
17. Where are rocky mountains located
18. Mountain conservation efforts
19. Highest peak in Asia
20. Most dangerous mountains

Showing relevant and non-relevant documents for the first 5 queries.

- It is extremely difficult to choose relevant & irrelevant results of a query without visiting each & every URLs in the result.
- For our project, we devised a way on which we search a query and get the top 50 results from solr. Out of the top 50 results, we consider the first 10 to be relevant and the last 10 to be irrelevant.

Query 1 - Tallest mountains in the world?

Top 5 Relevant: -

https://www.worldatlas.com/articles/the-world-s-tallest-mountain-ranges.html
https://www.worldatlas.com/articles/tallest-mountains-in-nepal.html
https://www.worldatlas.com/articles/the-tallest-mountains-in-the-alps.html

https://www.worldatlas.com/articles/the-volcanic-seven-summits-of-the-world.html
https://www.worldatlas.com/articles/tallest-mountains-in-tajikistan.html

Top 5 Irrelevant: -

https://www.worldatlas.com/articles/top-10-interesting-facts-about-tajikistan.html
https://www.worldatlas.com/articles/where-are-the-coast-mountains.html
https://www.worldatlas.com/articles/where-are-the-coast-mountains.html
https://www.elastic.co/guide/en/app-search/8.7/result-suggestions-guide.html
https://ntb.gov.np/en/things-to-do/zip-flying

Query 2 – Famous Mountain ranges

Top 5 Relevant: -

https://travel2next.com/mountains-in-india/
https://www.mountainiq.com/africa/
https://www.mountainiq.com/europe/
https://www.mountainiq.com/australia-oceania/
https://www.worldatlas.com/articles/oldest-mountain-ranges-of-the-world.html

Top 5 Irrelevant: -

https://www.milfordflights.co.nz/
https://www.worldatlas.com/rivers/indus-river.html
https://americanalpineclub.org/news/2023/4/12/the-american-alpine-club-announced-2023-cutting-edge-grant-winners-lj849
https://visitsmythcountyva.com/must-sees/the-back-of-the-dragon-2/
https://www.britannica.com/place/Wyoming-state

Query 3 - Mountains for hiking

Top 5 Relevant: -

https://wildlandtrekking.com/destination/great-smoky-mountains-hiking-tours/
https://wildlandtrekking.com/destination/white-mountains-hut-treks/
https://wildlandtrekking.com/destination/north-carolina-hikes/
https://www.worldatlas.com/articles/how-long-is-the-appalachian-trail.html
https://wildlandtrekking.com/destination/new-hampshire-hikes/

Top 5 Irrelevant: -

https://www.pinterest.com/ideas/craggy-gardens/914106643835/
https://www.pinterest.com/ideas/nc-mountains/944404080627/

https://mountainswithmegan.com/category/destinations/north-america/usa/
https://www.blueridgeoutdoors.com/go-outside/get-ready-to-race/
https://www.blueridgeoutdoors.com/go-outside/get-ready-to-race/

Query 4 - Himalayan mountain range

Top 5 Relevant: -

https://www.mountainiq.com/asia/himalayas/
https://travel2next.com/mountains-in-india/
https://internationalmountainmuseum.org/about-us
https://www.worldatlas.com/amp/mountains/10-major-mountain-ranges-of-asia.html
https://www.worldatlas.com/articles/tallest-mountains-in-nepal.html

Top 5 Irrelevant: -

https://www.icimod.org/initiative/air-pollution-solutions/
https://www.icimod.org/initiative/servir-hkh/
https://www.worldatlas.com/plateaus/tibetan-plateau.html
https://www.mountainresearchinitiative.org/resources-opportunities/publications-of-interest/3037-new-publications-october-2021
https://www.himalayandatabase.com/hawleyobit.html

Query 5 - Volcanic mountains

Top 5 Relevant: -

https://www.worldatlas.com/articles/how-are-mountains-formed.html
https://www.worldatlas.com/amp/articles/how-are-mountains-formed.html
https://wildlandtrekking.com/destination/lassen-volcanic-national-park-hiking-tours/
https://www.worldatlas.com/articles/europe-s-highest-volcano.html
https://www.worldatlas.com/articles/the-volcanic-seven-summits-of-the-world.html

Top 5 Irrelevant: -

https://www.worldatlas.com/amp/what-is-a-plateau.html
https://www.worldatlas.com/maps/guatemala
https://wildlandtrekking.com/trip-styles/fall-hiking-trips/page/2/
https://www.britannica.com/browse/Physical-Geography-Land
https://www.worldatlas.com/webimage/countrys/namerica/camerica/camland.htm

Original Queries with the expanded queries

1. Tallest mountains in the world ➔ **world mountain tallest rank one**
2. Famous mountain ranges ➔ **range famous mountain find highest**
3. Mountains for hiking ➔ **hiking mountain result appalachian including**
4. Himalayan mountain range ➔ **himalayan mountain range central one**
5. Volcanic mountains ➔ **volcanic mountain volcano ' different**
6. Mountain biking trails➔ **trail mountain biking sign photo**
7. Mountain resorts ➔ **resort mountain lake policy sign**
8. Mountain cabins ➔ **cabin mountain one lake group**
9. Mountain animals ➔ **animal mountain life deer area**
10. Tallest peaks in North America ➔ **peak america north tallest second**
11. Mountains in america with the best sunsets ➔ **sunset america best mountain town**
12. Mountain weather forecast ➔ **mountain forecast weather avalanche service**
13. Mountain ranges in Europe ➔ **range mountain europe peak western**
14. Oldest mountains ranges in the world ➔ **mountain range world oldest fact**
15. Highest peak in South America ➔ **south highest america peak continent**
16. Mountains for skiing ➔ t**ip mountain hiking hiker adventure**
17. Where are rocky mountains located ➔ **located rocky mountain park national**
18. Mountain conservation efforts ➔ **effort conservation mountain specie contact**
19. Highest peak in Asia ➔ **peak asia highest russia oldest**
20. Most dangerous mountains ➔ **dangerous mountain weather also one**

Pseudo-Relevance Feedback

There are three methods for pseudo relevance feedback mentioned in out report.

1. Associative Clustering
2. Metric Clustering
3. Scalar Clustering

The 50 queries selected for the pseudo-relevance feedback could be found below.

1. Best mountains for beginners
2. Mountains in the Rocky Mountain range
3. Tallest mountain in South America
4. Mountains in the Andes range
5. **Mount Everest facts**
6. Swiss Alps skiing
7. Patagonia glaciers
8. Kilimanjaro height
9. Mount Rainier views
10. Banff National Park
11. Mount McKinley Alaska
12. Yosemite hiking trails
13. Yosemite hiking trails

14. Best mountain towns in Colorado
15. Tallest mountain in North America
16. Machu Picchu Inca Trail
17. Caucasus range Europe
18. Appalachian range USA
19. Tian Shan Asia
20. Mount St. Helens eruption
21. Dinaric Alps trekking
22. Banff National Park lakes
23. Cordillera Blanca trekking
24. Annapurna climbers famous
25. Matterhorn Switzerland
26. Rocky Mountain wildlife
27. Dolomites hiking trails.
28. Patagonia wildlife watching
29. Famous mountaineers who climbed K2
30. **Interesting facts about Mount McKinley**
31. Highest mountain in Europe
32. Mountain ranges in New Zealand
33. Best time to visit the Austrian Alps
34. Mount Etna eruption history
35. Tenzing Norgay Everest
36. **Reinhold Messner Everest**
37. Doug Scott climbing
38. Lynn Hill Yosemite
39. Mount Fuji
40. Rwenzori Mountains
41. Mount Blanc
42. Aconcagua
43. Kilimanjaro
44. Mount Kosciuszko Hike
45. Indian Mountains
46. Mountain training schools
47. Sherpa People
48. Underwater mountains
49. Khuiten
50. Ghost stories of the mountains

The following section will consider three queries and show the following information.

1. The local document set.
2. The local vocab (token) set.
3. The local vocab (stem) set.
4. The cluster scores (for associated clusters)
5. The expanded query
6. The screenshot of the actual result.

QUERY ➔ Mount Everest facts **| EXPANSION METHOD ➔** Associative Clustering

- **Document set (Showing some URLs considered in the local document set) Total docs ➔ 20**

  https://en.wikipedia.org/wiki/Mount_Everest
  https://www.worldatlas.com/articles/the-seven-summits.html
  https://www.worldatlas.com/amp/articles/the-world-s-tallest-mountain-ranges.html
  https://www.mountainiq.com/best-time-to-trek-to-everest-base-camp/
  https://www.worldatlas.com/articles/tallest-mountains-in-nepal.html

- **Showing some local vocab tokens. Total token ➔ 4293s**

  ['appearance', 'funjiro', 'cut', 'autobiographical', 'seed', 'ed', 'must', 'summary', 'doctor', 'thing', 'host', 'isolated', 'trait', 'yielding', 'oxford', 'starting', 'drawn', 'entire', 'jengish', 'historic']

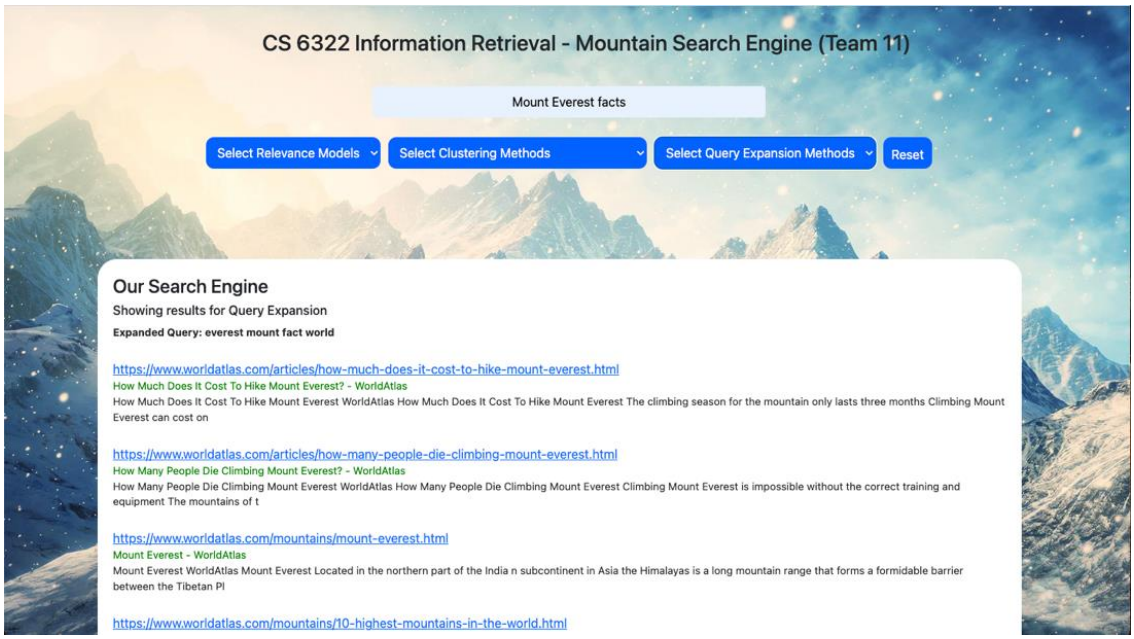
- **Showing some set of local stems. Total Stems ➔ 2856**

  ['glaci', 'indonesi', 'thorough', 'normal', 'common', 'use', 'stand', 'start', 'use', 'wr', 'holiday', 'event', 'marcel', 'agua', 'affect', 'messner', 'sansato', 'attach', 'tributar', 'griffon']

- **Some associated cluster values (The cluster sorted in descending order; we choose top n values)**

  [('fact', 'mount', 0.3333333333333333), ('fact', 'everest', 0.3333333333333333), ('fact', 'fact', 0.3333333333333333), ('mount', 'mount', 0.3333333333333333), ('mount', 'everest', 0.3333333333333333), ('mount', 'fact', 0.3333333333333333), ('everest', 'mount', 0.3333333333333333), ('everest', 'everest', 0.3333333333333333), ('everest', 'fact', 0.3333333333333333), ('world', 'mount', 0.3333333333333333), ('world', 'everest', 0.3333333333333333), ('world', 'fact', 0.3333333333333333), ('’', 'mount', 0.32142857142857145)]

- **Expanded query ➔** everest mount fact world



QUERY ➔ Mount Everest facts **| EXPANSION METHOD ➔** Scalar Clustering

- **Document set (Showing some URLs considered in the local document set) Total docs ➔ 20**

  https://en.wikipedia.org/wiki/Mount_Everest
  https://www.worldatlas.com/articles/the-seven-summits.html
  https://www.worldatlas.com/amp/articles/the-world-s-tallest-mountain-ranges.html
  https://www.mountainiq.com/best-time-to-trek-to-everest-base-camp/
  https://www.worldatlas.com/articles/tallest-mountains-in-nepal.html

- **Showing some local vocab tokens. Total token ➔ 6523**

  ['türkmençe', 'türkçe', 'u', 'ueli', 'uganda', 'uiaa', 'uk', 'ukclimbing', 'ukraine', 'ullman', 'ultimate', 'ultimately', 'ultra', 'un', 'unable', 'unabridged', 'unbearably', 'unbeatable', 'unbeknownst', 'uncertain', 'uncertainty', 'unclear', 'unclimbable', 'unclimbed', 'uncommon', 'unconquerable', 'unconscious', 'uncovered', 'uncovers']

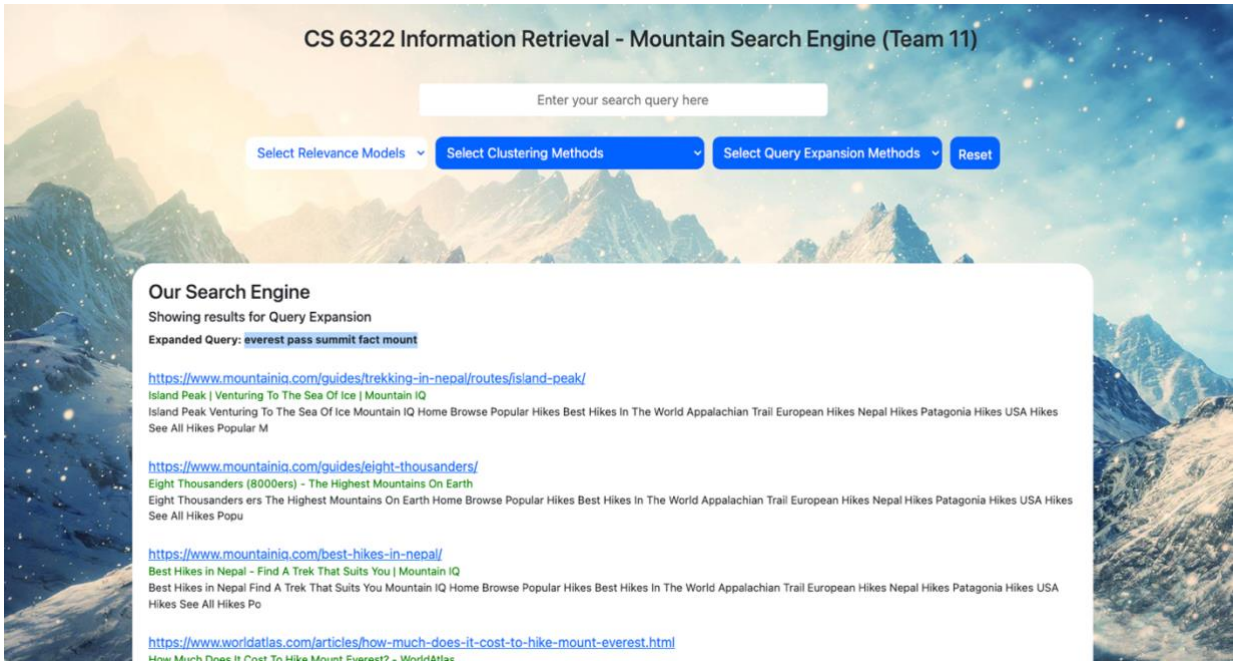
- **Showing some set of local stems. Total Stems ➔ 2956**

  ['dave', 'daughter', 'date', 'databas', 'data', 'dash', 'darren', 'dark', 'darjeel', 'dansk', 'danish', 'daniel', 'danhara', 'danger', 'dangar', 'danc', 'dan', 'damian', 'damag', 'dali', 'daley', 'dalai', 'daili', 'dablam', 'dabba', 'da', 'czech', 'cypru', 'cymraeg', 'cymerman', 'cylind', 'cycl', 'cyanobacteria', 'cwm', 'cvlll']

- **Expanded query.**

  everest pass summit fact mount

- **Screenshot**



QUERY ➔ Mount Everest facts **| EXPANSION METHOD ➔** Metric Clustering

- **Document set (Showing some URLs considered in the local document set) Total docs ➔ 20**

  https://en.wikipedia.org/wiki/Mount_Everest
  https://www.worldatlas.com/articles/the-seven-summits.html
  https://www.worldatlas.com/amp/articles/the-world-s-tallest-mountain-ranges.html
  https://www.mountainiq.com/best-time-to-trek-to-everest-base-camp/
  https://www.worldatlas.com/articles/tallest-mountains-in-nepal.html

- **Showing some local vocab tokens. Total token ➔ 6523**

  ['occasional', 'obtaining', 'obtained', 'obstacles', 'obstacle', 'observed', 'observatory', 'observations', 'observation', 'oblivion', 'objections', 'obituaries', 'obama', 'o2', 'nāhuatl', 'nzers', 'nzac', 'nz', 'nw', 'nuwer', 'nutritional', 'nushik', 'nuru', 'nuptse', 'nup', 'numerous', 'numerals]

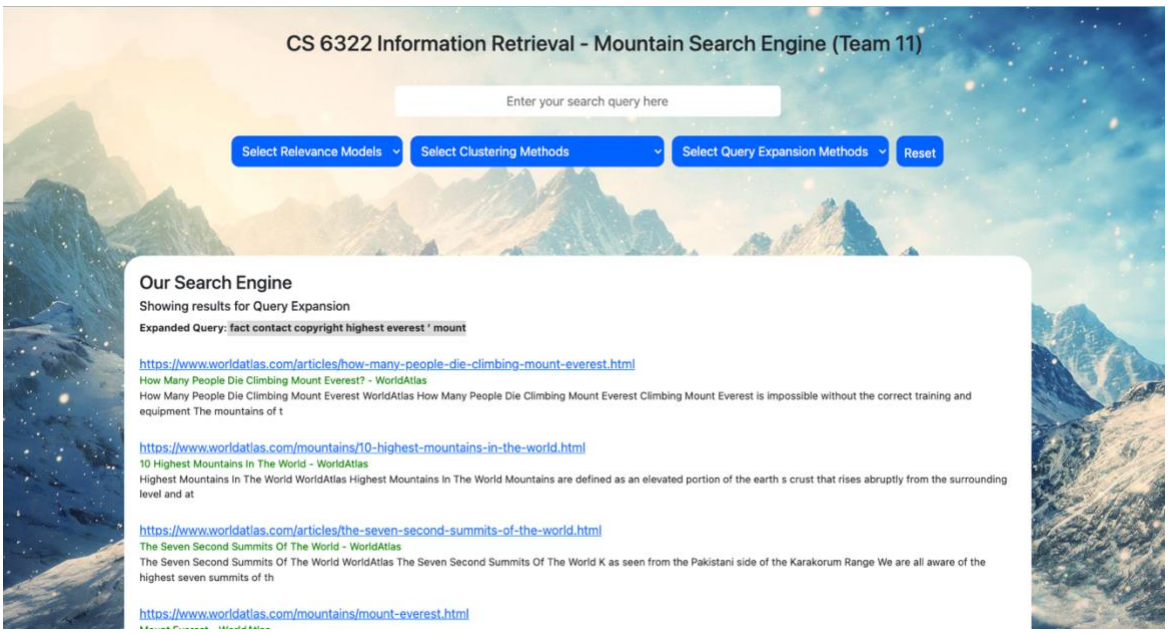
- **Showing some set of local stems. Total Stems ➔ 2956**

  ['salween', 'salticida', 'salon', 'salman', 'salkeld', 'salisburi', 'sale', 'sakai', 'saint', 'sailor', 'said', 'sahara', 'sagarmāthā', 'sagarmatha', 'sagar', 'safeti', 'safer', 'safeguard', 'safe', 'safari', 'sacr', 'sack', 'saburo', 's2cid', 'récord', 'rwanda', 'ruttledg', 'rustl', 'rustic', 'russian', 'russia', 'russel']

- **Expanded query**

  fact contact copyright highest everest ’ mount

- **Screenshot**



QUERY ➔ Reinhold Messner Everest **| EXPANSION METHOD ➔** Associative Clustering

- **Document set (Showing some URLs considered in the local document set) Total docs ➔ 20**

  https://en.wikipedia.org/wiki/Reinhold_Messner
  https://theuiaa.org/members-area/honorary-members/reinhold-messner/
  http://library.americanalpineclub.org/everest
  https://www.himalayandatabase.com/Research/Hillary-Interview.html
  http://www.alanarnette.com/climbing/climbinglinks.php

- **Showing some local vocab tokens. Total token ➔ 6876**

  ['artikel', 'hatched', 'popular', 'slider', 'impressive', '◌ī', 'danish', 'web', 'wirklich', 'interested', 'injured', 'match', 'e9', 'stein', 'happy', 'greater', 'dharamsala', 'capsule', 'andrew', 'indien']

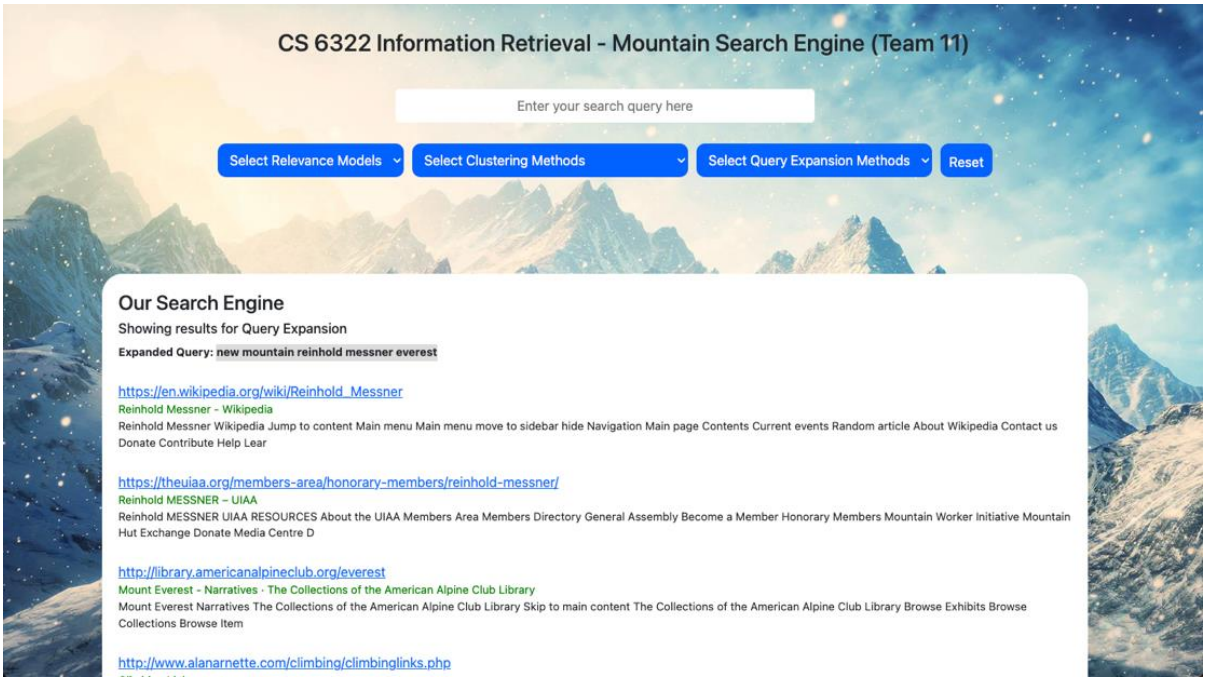
- **Showing some set of local stems. Total Stems ➔ 4705**

  ['locat', 'thru', 'else', 'metre', 'holder', 'wilder', 'legend', 'roof', 'title', 'sake', 'office', 'throw', 'attribut']

- **Expanded query ➔** new mountain reinhold messner everest

- **Some associated cluster values (The cluster sorted in descending order; we choose top n values)**

  [('messner', 'reinhold', 0.3333333333333333), ('messner', 'messner', 0.3333333333333333), ('mountain', 'everest', 0.3333333333333333), ('new', 'everest', 0.3333333333333333), ('everest', 'everest', 0.3333333333333333), ('reinhold', 'reinhold', 0.3333333333333333), ('reinhold', 'messner', 0.3333333333333333), ('messner', 'everest', 0.32142857142857145), ('climbing', 'reinhold', 0.32142857142857145), ('climbing', 'messner', 0.32142857142857145), ('mountain', 'reinhold', 0.32142857142857145), ('mountain', 'messner', 0.32142857142857145)]



QUERY ➔ Reinhold Messner Everest **| EXPANSION METHOD ➔** Scalar Clustering

- **Document set (Showing some URLs considered in the local document set) Total docs ➔ 20.**

  https://en.wikipedia.org/wiki/Reinhold_Messner
  https://theuiaa.org/members-area/honorary-members/reinhold-messner/
  http://library.americanalpineclub.org/everest
  https://www.himalayandatabase.com/Research/Hillary-Interview.html
  http://www.alanarnette.com/climbing/climbinglinks.php

- **Showing some local vocab tokens. Total token ➔ 8267**

  [‘caked', 'caird', 'cai', 'cadore', 'cabot', 'cable', 'c', 'ből', 'bührer', 'bücher', 'böse', 'bírt', 'bärenhaare', 'bären', 'bär', 'bámulja', 'bzw', 'byrne', 'byrd', 'bypass', 'bylot', 'buying', 'buy', 'buttress', 'busy', 'business', 'bushes', 'bush', 'buscar', 'bus', 'bury', 'burton', 'burns', 'burnham', 'burke', 'buried']

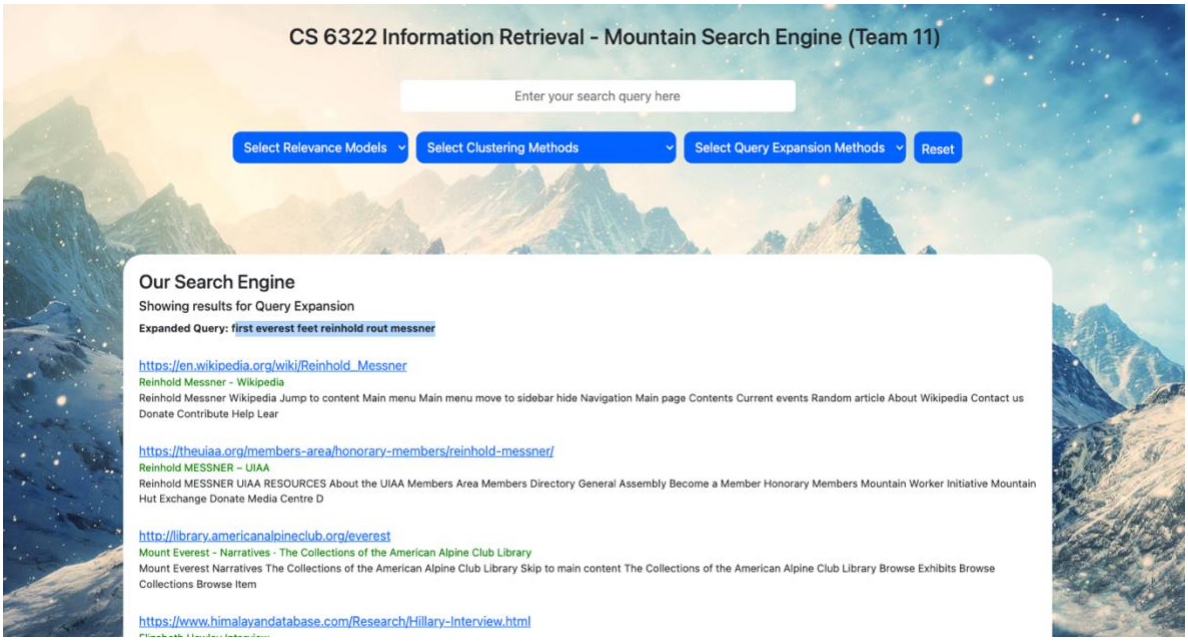
- **Showing some set of local stems. Total Stems ➔ 7263**

  ['paid', 'pagina', 'pagin', 'page', 'padova', 'padoan', 'padding_top', 'padding_bottom', 'pad', 'pack', 'pacifico', 'pacif', 'pace', 'pablo', 'pa', 'p1417', 'p', 'o‘zbekcha', 'oönachrichten', 'oö', 'oyu', 'oxygen', 'oxigeno', 'oxfordi', 'oxford', 'ownership', 'owner', 'ovtsyn', 'ovest', 'overwhelm', 'overse', 'overnight', 'overland', 'overcom']

- **Expanded query.**

  irst everest feet reinhold rout Messner

- **Screenshot**



QUERY ➔ Reinhold Messner Everest **| EXPANSION METHOD ➔** Metric Clustering

- **Document set (Showing some URLs considered in the local document set) Total docs ➔ 20**

  https://en.wikipedia.org/wiki/Reinhold_Messner
  https://theuiaa.org/members-area/honorary-members/reinhold-messner/
  http://library.americanalpineclub.org/everest
  https://www.himalayandatabase.com/Research/Hillary-Interview.html
  http://www.alanarnette.com/climbing/climbinglinks.php

- **Showing some local vocab tokens. Total token ➔ 8267**

  ['yetis', 'yeti', 'yet', 'yesterday', 'yes', 'yerupaja', 'yermak', 'yeren', 'yelena', 'yeh', 'years' , 'yards', 'yagihara', 'xuèrén', 'xix', 'xinjiang', 'xavi', 'xabier', 'x', 'wändle', 'während', 'wächter', 'wyss', 'www', 'wyoming', , 'wurden', 'wurde', 'wrote', 'wrong', 'written', 'writing', 'writes', 'writer']

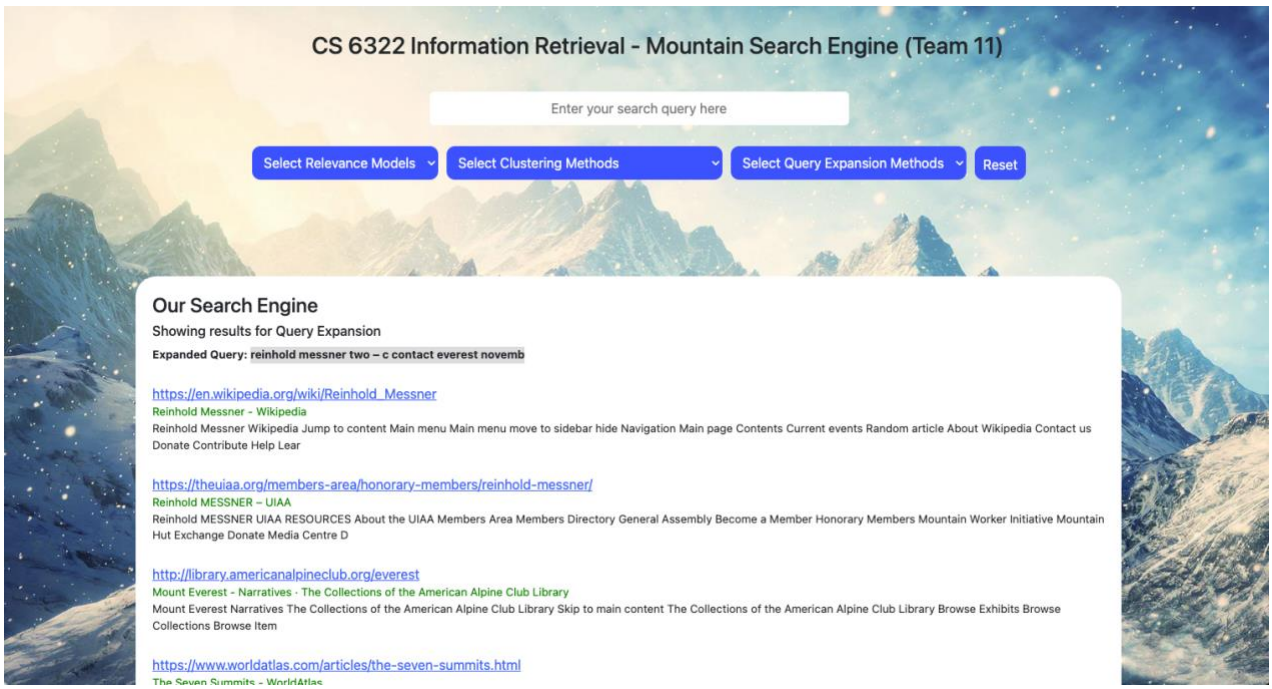
- **Showing some set of local stems. Total Stems ➔ 7263**

  [balti', 'balloon', 'balli', 'ballestero', 'ball', 'bali', 'baldini', 'balchen', 'balanc', 'bahasa', 'bag', 'baffin', 'badygin', 'badli', 'bad', 'backward', 'backpack', 'background', 'backcountri', 'back', 'babi', 'b69', 'b', 'azərbaycanca', 'azzurro', 'azt', 'azonosítóv', 'azonban', 'aziend', 'az', 'ayer', 'axe', 'awesom']

- **Expanded query.**

  reinhold messner two – c contact everest novemb

- **Screenshot**



QUERY ➔ Interesting facts about Mount McKinley **| EXPANSION METHOD ➔** Associative Clustering

- **Document set (Showing some URLs considered in the local document set) Total docs ➔ 20.**

  https://www.worldatlas.com/articles/10-interesting-facts-about-mount-denali.html
  https://www.worldatlas.com/mountains/mount-denali.html
  https://americanalpineclub.org/news/tag/Mount+McKinley
  https://americanalpineclub.org/news/tag/Denali
  https://www.worldatlas.com/amp/mountains/mount-foraker.html

- **Showing some local vocab tokens. Total token ➔ 3739**

  ['questioned', 'hatched', 'hydrated', 'popular', 'star', 'interested', 'legislature', 'happy', 'typhoon', 'india', 'fun', 'mesmerising', 'civilization', 'downloaded', 'photogrammetric', 'thru', 'metre', 'bianco', 'roof', 'legend']

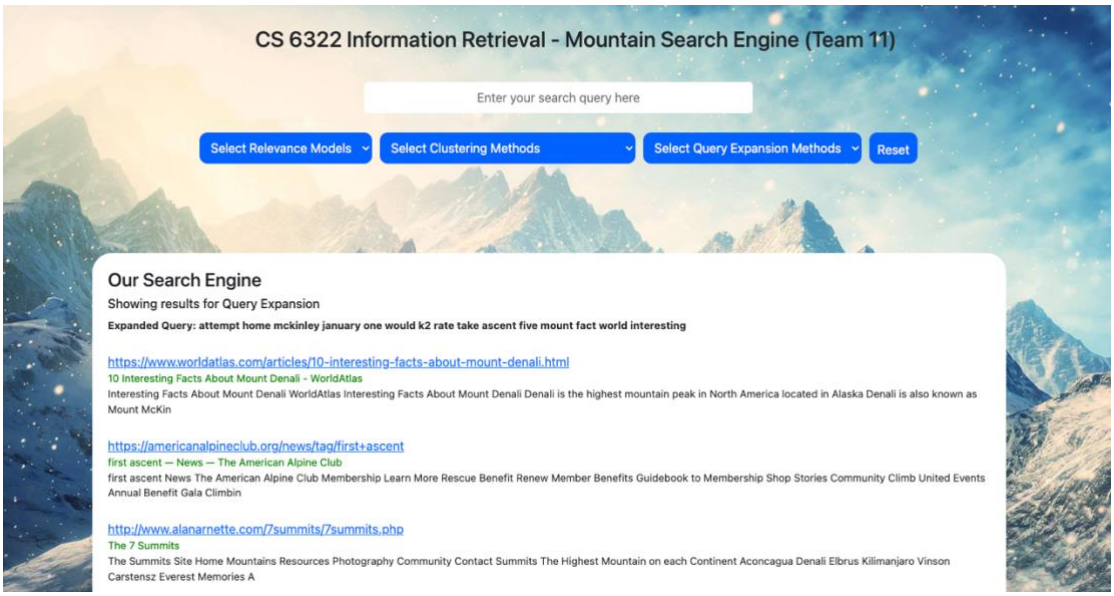
- **Showing some set of local stems. Total Stems ➔ 2290**

  ['overnight', 'dictat', 'frty6', 'beauti', 'claim', 'dozen', 'resourc', 'sen', 'mari', 'pollut', 'chanc', 'taylor', 'patrick', 'columbia', 'jaya', 'instal', 'merlin', 'plaudit', 'major', 'nag']

- **Expanded query. ➔** attempt home mckinley january one would k2 rate take ascent five mount fact world interesting

- **Some associated cluster values (The cluster sorted in descending order; we choose top n values)**

  [('became', 'mckinley', 0.3333333333333333), ('would', 'mckinley', 0.3333333333333333), ('five', 'mckinley', 0.3333333333333333), ('attempt', 'mckinley', 0.3333333333333333), ('january', 'mckinley', 0.3333333333333333), ('take', 'mckinley', 0.3333333333333333), ('interesting', 'interesting', 0.3333333333333333), ('k2', 'mckinley', 0.3333333333333333), ('one', 'mount', 0.3333333333333333), ('fact', 'fact', 0.3333333333333333), ('mckinley', 'mckinley', 0.3333333333333333), ('mount', 'mount', 0.3333333333333333), ('rate', 'mckinley', 0.3333333333333333), ('world', 'interesting', 0.3333333333333333)]



QUERY ➔ Interesting facts about Mount McKinley **| EXPANSION METHOD ➔** Scalar Clustering

- **Document set (Showing some URLs considered in the local document set) Total docs ➔ 20**

  https://www.worldatlas.com/articles/10-interesting-facts-about-mount-denali.html
  https://www.worldatlas.com/mountains/mount-denali.html
  https://americanalpineclub.org/news/tag/Mount+McKinley
  https://americanalpineclub.org/news/tag/Denali
  https://www.worldatlas.com/amp/mountains/mount-foraker.html

- **Showing some local vocab tokens. Total token ➔ 4017**

  ['earth', 'earns', 'earned', 'early', 'earliest', 'earlier', 'e', 'dykh', 'dying', 'dychtau', 'duty', 'duties', 'dutch', 'dushanbe', 'dupetit', 'dunn', 'dunes', 'due', 'du', 'druk', 'drowns', 'drops', 'dropping', 'dropped', 'drop', 'driving', 'driven', 'drinking', 'drink', 'drier', 'dressed', 'dream', 'drastically', 'dramatic', 'drain', 'dragon']

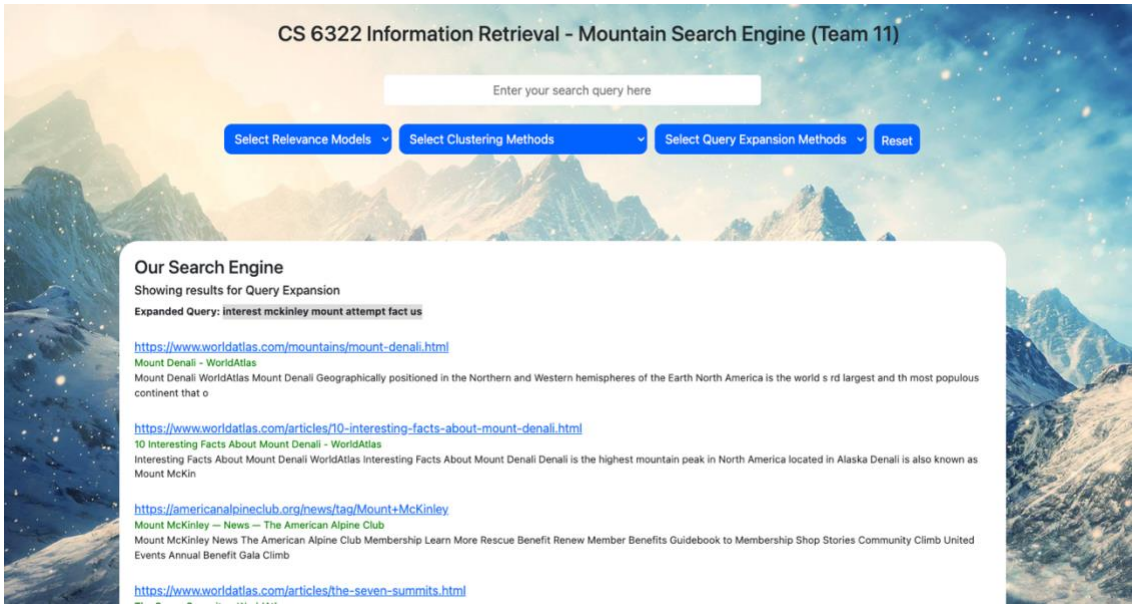
- **Showing some set of local stems. Total Stems ➔ 3211**

  ['fale', 'fair', 'failur', 'fail', 'fahrenheit', 'factor', 'fact', 'facil', 'facebook', 'face', 'fabl', 'faanui', 'fa', 'f83d7', 'f3080', 'f2bba', 'eye', 'extrus', 'extrem', 'extinct', 'extent', 'extens', 'extend', 'exposur', 'export', 'explos', 'explor', 'explan', 'explain', 'expert', 'experi', 'expedit', 'expect', 'expand']

- **Expanded query.**

  interest mckinley mount attempt fact us

- **Screenshot**



QUERY ➔ Interesting facts about Mount McKinley **| EXPANSION METHOD ➔** Metric Clustering

- **Document set (Showing some URLs considered in the local document set) Total docs ➔ 20**

  https://www.worldatlas.com/articles/10-interesting-facts-about-mount-denali.html
  https://www.worldatlas.com/mountains/mount-denali.html
  https://americanalpineclub.org/news/tag/Mount+McKinley
  https://americanalpineclub.org/news/tag/Denali
  https://www.worldatlas.com/amp/mountains/mount-foraker.html

- **Showing some local vocab tokens. Total token ➔ 4017**

  ['barrier', 'barrels', 'barrel', 'barometric', 'barely', 'bare', 'barbary', 'bar', 'banned', 'banks', 'banjul', 'bangui', 'bandits', 'band', 'ban', 'bamako', 'baltic', 'balance', 'bahr', 'bags', 'bagged', 'bag', 'baffin', 'badakhshan', 'bad', 'back', 'bachmann', 'b06xzy8g76', 'b00jm90on8', 'b', 'axis', 'axial', 'axe', 'away', 'awareness']

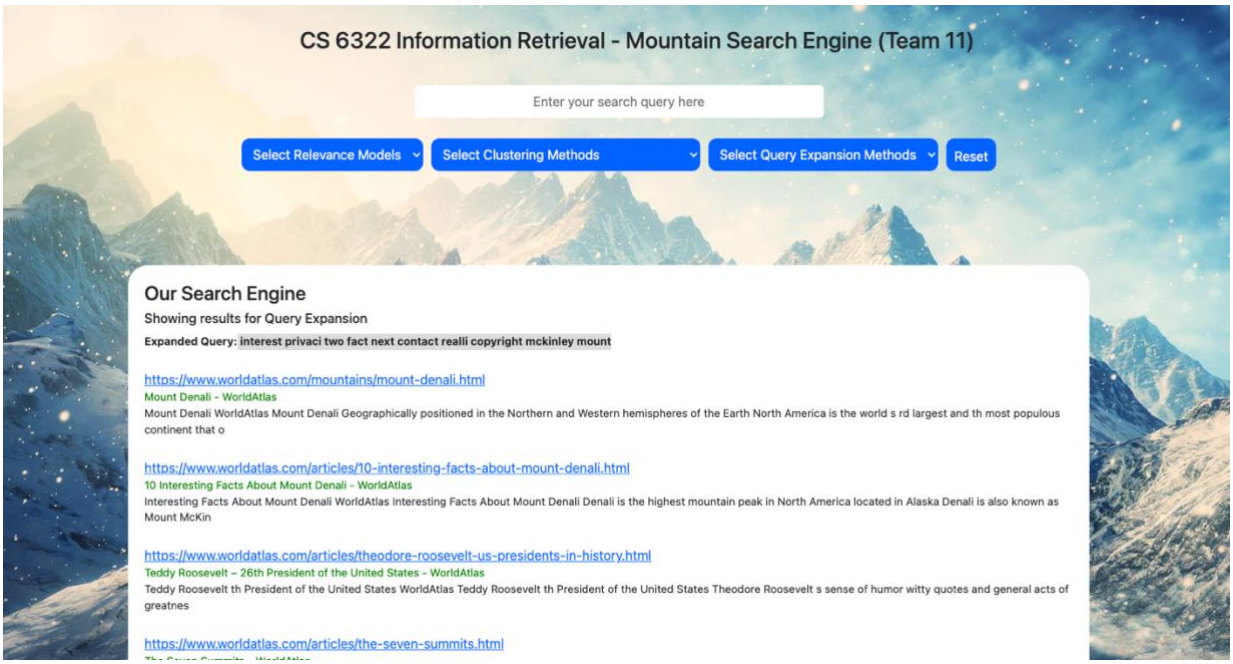
- **Showing some set of local stems. Total Stems ➔ 3211**

  ['baltic', 'balanc', 'bahr', 'bag', 'baffin', 'badakhshan', 'bad', 'back', 'bachmann', 'b06xzy8g76', 'b00jm90on8', 'b', 'axial', 'axi', 'axe', 'away', 'award', 'awar', 'await', 'avro', 'avoid', 'avid', 'aviat', 'avert', 'averag', 'avalanch', 'avail', 'autumn', 'autom', 'authorit', 'author', 'austrian', 'australian', 'australia']

- **Expanded query.**

  interest privaci two fact next contact realli copyright mckinley mount

- **Screenshot**



Collaboration With Other Team Members

- **With X2:-**
    - Worked with X2 to get the vector space relevance model.
    - Retrieved documents associated with the query from the relevance model provided by X2.
    - Used appropriate algorithms to generate the expanded query.
    - Used the relevance model provided by X2 again to get results for the new expanded query.
- **With X3:-**
    - Worked with X3 to retrieve the query entered by the user at the User Interface.
    - Used appropriate query expansion algorithms to get the expanded query and the results associated with it.
    - Sent the expanded query and the results back to X3 so that he can display them on the user interface.

Queries Chosen for demonstration of the project.

There are three main queries that I choose for the demonstration of query expansion part in the project.

1. Mount Everest Facts
2. Interesting facts about Mount McKinley
3. Reinhold Messner Everest

The detailed results for all the above queries are mentioned in the pages above. It is worth mentioning that all the query expansion algorithms performed remarkably well on queries of all sizes. Notably, the Rocchio algorithm outperformed all the other algorithms implemented in this project.

It is important to note that the results of all the algorithms could have been even better if I had access to a computer with more memory and processing capabilities. Due to the limited data available, I had to restrict the local document set provided to each algorithm to just 20 documents. As a result, the algorithms were not able to perform at their full potential. However, considering the limited data available, the results were generally above average and sometimes significant.

Arpit Singh AXS210204
Tanmay Singhal TXS210014
Prem Sharma PXS210046
Karan Jariwala KHJ200000
Anirudh Kiran AXK200227

Discussion

Firstly, we would like to express that the process of creating a search engine is a highly complex endeavor. There are numerous moving parts, and at times, the developers are not in complete control of these parts. For instance, our initial and foremost challenge was web crawling. Determining how a web crawler would navigate and visit links is an intricate task. Through the web crawling process, we gained valuable knowledge about Apache Nutch, an exceptionally powerful tool that formed the core of our project. Assessing the data returned by Nutch was not a simple task, and we had to explore various ways to access and use this data for indexing, ranking, and other tasks. Data handling is one of the most significant challenges in building a search engine, and we faced extreme difficulties when we began the clustering process.

As all the data must be used, we had to reconsider each loop in our code to optimize the computer's memory and processing capabilities to their maximum potential. However, we persevered through the challenging clustering task and eventually produced moderate results. Query expansion was our last and most notorious obstacle. This phase is typically the most challenging because each document has thousands of tokens, and processing all these tokens with respect to the initial query is extremely difficult. Nevertheless, we exerted our best efforts and attempted multiple methods to optimize query expansion, achieving some success. Overall, this project has taught us numerous concepts that we previously had no knowledge of, and we owe our gratitude to this class for providing us with this learning opportunity. Each of us had the chance to gain practical knowledge, and it was one of our best decisions to be part of this class. Finally, we express our heartfelt thanks to our professor for guiding us through this project and for their constant support.

Conclusion

Building a search engine for the mountains was the objective of our project. We gained extensive knowledge about the various aspects involved in creating a functional search engine specifically for the mountains. By implementing the techniques taught in class, we were able to successfully develop a search engine that produced moderately relevant results for mountain related queries. However, this project has also made us aware of the limitations of our knowledge in Information Retrieval, and the complexity involved in creating and maintaining a popular search engine. We are grateful to the professor as well as the TA for providing us with the opportunity to work on this project and for their guiding and continuous support through the project.