

**The University of Texas at Dallas
CS 6322.001
Information Retrieval
Fall 2018**

Class Project Proposal

Project TITLE: Food Recipes Engine

Students

Rutuj Puranik, RXP180014@utdallas.edu

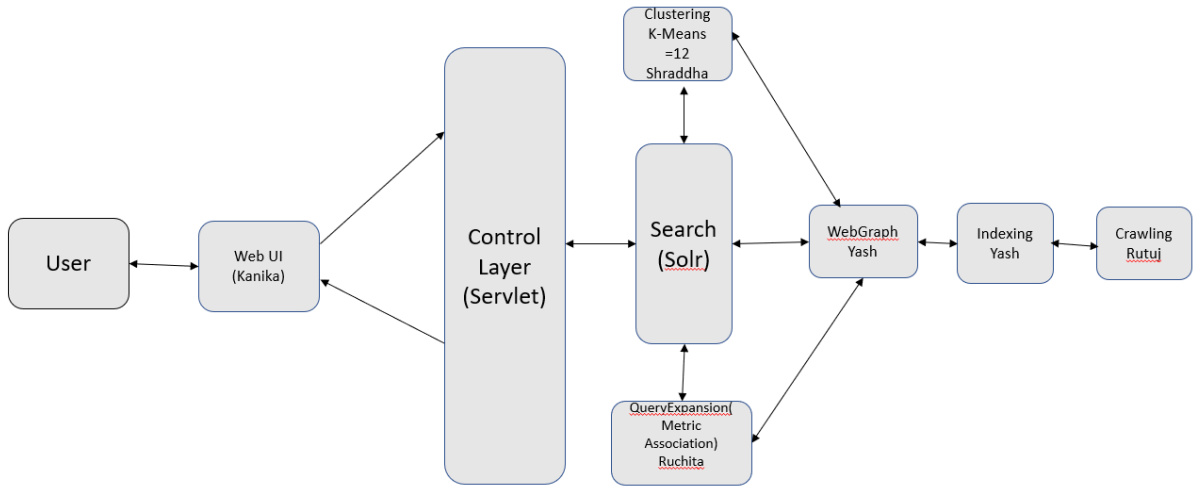
Yash Hemant Jain, YXJ180004@utdallas.edu

Kanika Rawat, KXR180008@utdallas.edu

Shraddha Bang, SXB180041@utdallas.edu

Ruchita Deshmukh, RLD170003@utdallas.edu

ARCHITECTURE



1. Crawling – Rutuj Puranik

For crawling web pages associated with food, The Apache Nutch Framework was utilized for crawling as well as feeding the Fetched Content from crawling to the Solr Framework hosted on localhost for indexing the fetched web pages as well as creating web graphs for implementing Page Rank and the HITS algorithms.

The number of pages crawled were 1,286,377

The number of web pages crawled & fetched were 166,180.

For crawling web pages associated with music, The Apache Nutch Framework was utilized for crawling as well as feeding the Fetched Content from crawling to the Solr Framework hosted on localhost for indexing the fetched web pages as well as creating web graphs for implementing Page Rank and the HITS algorithms.

The number of pages crawled were 1,700,000

The number of web pages crawled & fetched were 166,180.

The Apache Nutch Framework was made to crawl allrecipes.com, epicurious.com, homechef.com, yumly.com etc.

The Nutch Framework was fed 151 URLs as seed URL to start crawling. These URLs mainly constituted Famous Chefs, Varieties of food, All Time Favorites, Famous Food Channels on online video streaming websites, Famous food cultures and Famous food Distributors.

The following are a sample list of URLs used:

<https://www.tarladalal.com>

<https://www.sanjeevkapoor.com>

<https://www.gordonramsay.com/gr/recipes/>

<https://www.thepetitecook.com/easy-stuffed-mushrooms/>

<http://www.manjulaskitchen.com/recipes/appetizers/>

<https://www.sainsburysmagazine.co.uk/recipes/starters>

<https://yourvegrecipe.com/veg-starters-snacks>

<https://gimmedelicious.com/2018/03/30/15-minute-one-pan-shrimp-and-rice/>

<https://sweetcsdesigns.com/the-best-one-pot-lemon-garlic-butter-shrimp-recipe-ever/>

<https://www.theslowroasteditalian.com/2015/05/lemon-garlic-shrimp-recipe.html>

<https://www.jocooks.com/recipes/spicy-new-orleans-shrimp/>

<http://www.bumblebee.com/recipes/shrimp-fried-rice/>

<https://www.theworktop.com/breakfast-brunch-recipes/orange-juice-brunch-mocktail/>

<https://www.playpartyplan.com/citrus-strawberry-mocktail-recipe/>

<https://www.bowlofdelicious.com/honey-blackberry-mint-mocktails/>

<https://www.realsimple.com/food-recipes/recipe-collections-favorites/mocktail-recipes>

<https://www.monin.com/us/recipes/mocktail-recipes>

<https://twohealthykitchens.com/9-all-time-best-healthy-easy-seafood-and-fish-recipes/>

<https://www.laylita.com/recipes/fish-and-seafood-recipes/>

<https://www.ibreatheimhungry.com/110-best-keto-seafood-recipes-low-carb/>
<https://www.thecookierookie.com/cheesy-seafood-lasagna/>
<https://www.feastingathome.com/summer-seafood-stew/>
<https://www.americastestkitchen.com/recipes/browse/seafood>
<https://paleoleap.com/category/paleo-diet-recipes/fish-seafood/>
<https://www.cooksillustrated.com/articles/27-bringing-back-baked-alaska>
<https://hipfoodiemom.com/2014/10/28/seafood-pappardelle/>
<https://www.homechef.com/recipes/seafood>
<https://www.tastingtable.com/cook/recipes/baked-alaska-cake-recipe-dessert-party-recipe-ideas-homemade-meringue>
<https://emerils.com/127733/emerils-baked-alaska>
<https://www.dessertfortwo.com/mini-baked-alaskas/>
<https://whatshouldimakefor.com/baked-alaska/>
<https://www.homecookingadventure.com/recipes/baked-alaska>
<https://entertainingwithbeth.com/holiday-baked-alaska-recipe/>
<http://thewoodandspoon.com/baked-alaska/>
<http://www.foodrepublic.com/recipes/l-a-chic-spaghetti-with-sea-urchin/>
<https://norecipes.com/sea-urchin-ceviche-recipe>
<http://holafoodie.com/recipe/galician-sea-urchins-gratin/>
<https://www.epicurious.com/recipes/food/views/sea-urchin-mousse-with-ginger-vinaigrette-232779>
<https://www.yummly.com/recipes/sea-urchin>
<https://food52.com/recipes/30182-rick-stein-s-sea-urchin-pasta>
<https://goop.com/recipes/linguine-sea-urchin-chili/>
<https://www.triedandsupplied.com/saucydressings/how-to-eat-sea-urchins/>
<https://recipeland.com/recipe/v/fresh-pumpkin-puree-for-pumpkin-57531>
<https://www.chowhound.com/recipes>
https://www.chefsteps.com/gallery?generator=chefsteps&published_status=published&difficulty=any&sort=newest&premium=everything
<https://www.beachbodyondemand.com/blog/recipes>
<https://www.fifteenspatulas.com/homemade-sushi/>
<http://secretsofsushi.com/sushi-roll-recipes>
<https://www.kitchensanctuary.com/crispy-sesame-chicken-sticky-asian-sauce/>
<https://www.tastemade.com/recipes/asian>
<https://ohsweetbasil.com/easy-asian-orange-chicken/>
<https://thelemonbowl.com/20-healthy-asian-inspired-recipes/>
<https://www.asian-recipe.com>
<https://www.tasteandtellblog.com/asian-rice-beef-stir-fry-recipe/>
<https://veganheaven.org/all-recipes/50-amazing-vegan-asian-recipes/>
<https://showmethemummy.com/easy-asian-noodles/>
<https://www.cleaneatingmag.com/recipes/recipe-cuisine/asian/>
<https://www.hellofresh.com/recipes/asian-cuisine?order=-rating>
<https://joyfoodsunshine.com/asian-chicken-lettuce-wraps/>
<https://spicysouthernkitchen.com/asian-garlic-tofu/>

<http://www.europeancuisines.com>
<http://www.gourmet-european-recipes.com/quick-and-easy-dinner.html>
<http://delightsofculinaria.com/recipes/european-recipes/>
<https://www.mygourmetconnection.com/cuisine/european/>
<https://cinnamonandcoriander.com/en/category/global-kitchen/europa/>
<https://www.196flavors.com/category/continent/europe/northern-europe/>
<https://adorefoods.com/category/recipes/european-recipes/>
<https://www.curiouscuisiniere.com/around-the-world/eastern-european/>
<https://www.rednumberone.com/eastern-european-recipes/>
<https://www.lavenderandmacarons.com/entrees/steak-chateaubriand-french-recipes-menu/>
<https://cookingisl.com/easy-oven-baked-honey-mustard-chicken-thighs-recipe/>
<https://thesaltymarshmallow.com/crispy-baked-chicken-thighs/>
<https://jenniferbanz.com/crispy-chicken-thighs>
<https://valentinascorner.com/baked-tender-chicken-thighs-recipe/>
<https://theviewfromgreatisland.com/creamy-tuscan-chicken-thighs-recipe/>
<https://www.savingdessert.com/spicy-honey-lime-chicken-thigh-recipe/>
<https://tipbuzz.com/oven-baked-chicken-thighs/>
<https://bakeatmidnite.com/stuffed-chicken-thighs/>
<https://letthebakingbegin.com/garlic-ranch-baked-chicken-thighs/>
<https://www.crunchycreamysweet.com/2018/07/13/instant-pot-chicken-thighs/>
<http://kitchenswagger.com/sweet-and-spicy-korean-chicken-thighs-recipe/>
<https://barefeetintheKitchen.com/perfect-pan-fried-chicken-thighs-recipe/>
<https://www.101cookingfortwo.com/oven-baked-chicken-thighs/>
<http://wholeandheavenlyoven.com/2018/01/22/skillet-honey-garlic-chicken-thighs-roast-potatoes/>
<https://www.thewholesomedish.com/the-best-classic-lasagna/>
<https://www.campbells.com/kitchen/recipes/classic-lasagna/>
<https://cupcakesandkalechips.com/worlds-best-lasagna-or-at-least-that-is-what-the-recipe-says-and-my-gluten-free-alternative/>
<https://www.davidlebovitz.com/perfect-panna-cotta/>
<https://www.bakedbyanintrovert.com/strawberry-panna-cotta/>
<https://www.handletheheat.com/panna-cotta-with-raspberry-sauce/>
<https://headbangerskitchen.com/recipe/keto-panna-cotta/>
<https://www.weightwatchers.com/us/recipe/panna-cotta-with-berries/59b025c8304f3002adfbb231>
<https://swervesweet.com/recipes/strawberry-panna-cotta>
<https://www.culinarynutrition.com/top-30-gluten-free-dinner-recipes/>
<https://chooseveg.com/blog/16-delicious-things-to-cook-with-tempeh/>
<https://homemadehooplah.com/honey-garlic-chicken-wings/>
<https://www.eazypeazymealz.com/crispy-oven-baked-chicken-wings/>
<https://thecookful.com/bake-chicken-wings-crispy/>
<https://www.foodtasticmom.com/crispy-baked-chicken-wings/>
<http://khanakhazana.com>
<https://www.yummytummyarthi.com/2014/07/chicken-lollipop-recipe-restaurant.html>

http://recipes.wikia.com/wiki/Recipes_Wiki
<https://en.wikibooks.org/wiki/Cookbook:Recipes>
<https://www.dinneratthetoo.com/baked-lemon-chicken/>
<https://cookiesandcups.com/miym-melt-in-your-mouth-chicken-breasts/>
<https://www.delish.com/cooking/recipe-ideas/g2972/chicken-weeknight-dinners/>
<https://www.foodnetwork.com/topics/chicken>
<https://realhousemoms.com/honey-butter-chicken/>
<https://www.myrecipes.com/chicken-recipes>
<https://dinnerthendessert.com/bacon-brown-sugar-chicken/>
<https://iwashyoudry.com/last-minute-chicken-recipe/>
<https://www.tasteofhome.com/recipes/quick-chicken-piccata/>
<https://www.spendwithpennies.com/skillet-orange-chicken-recipe/>
<https://www.tablefortwoblog.com/holy-yum-chicken/>
<https://sweetpeasandsaffron.com/7-slow-cooker-chicken-recipes/>
<https://cafedelites.com/easy-honey-garlic-chicken/>
<https://www.bonappetit.com/recipes/chicken>
<https://thatlowcarbblife.com/spinach-stuffed-chicken-2/>
<https://easyfamilyrecipes.com/salsa-fresca-chicken/>
<https://www.thespruceeats.com/chicken-4162452>
<https://www.readyseteat.com/recipes-Chicken-Burrito-Skillet-7722>
<https://www.recipeetineats.com/oven-baked-chicken-breast/>
<https://www.thekitchn.com/ina-garten-best-chicken-recipes-264848>
<http://www.kitchme.com/recipes/melt-in-your-mouth-chicken-breast>
<https://www.foodandwine.com/meat-poultry/chicken>
<https://www.countryliving.com/food-drinks/g680/chicken-recipes-0109/>
<https://www.epicurious.com/recipes/food/views/hanukkah-chicken>
https://www.bbc.com/food/recipes/easy_spanish_chicken_09987
<https://rasamalaysia.com/chicken-recipes/>
<https://www.skinnytaste.com/main-ingredient/chicken-recipes/>
<https://therecipecritic.com/garlic-alfredo-pasta/>
<https://www.budgetbytes.com/lighter-spinach-alfredo-pasta/>
<https://cravinghomecooked.com/easy-pasta-alfredo/>
<https://natashaskitchen.com/moms-chicken-fettuccine-alfredo/>
<https://www.bettycrocker.com/recipes/chicken-alfredo-pasta-skillet/c1a0d71a-7fab-4888-a98d-a48c376588b3>
<https://prettysimplesweet.com/mushroom-alfredo-pasta/>
<https://www.tastesoflizzyt.com/chicken-alfredo-pasta/>
<https://www.pillsbury.com/recipes/chicken-alfredo-baked-penne/cb946ddc-a330-42bc-8d7e-ef682c25460a>
<https://anitalianinmykitchen.com/alfredo-pasta/>
<https://www.tasteslovely.com/broccoli-chicken-fettuccine-alfredo/>
<https://whatsgabycooking.com/cauliflower-alfredo-pasta/>
<https://tasty.co/recipe/healthier-chicken-alfredo-pasta>
<https://www.splendidtable.org/recipes/simple-one-skillet-chicken-alfredo-pasta>

<https://laurenslatest.com/browned-butter-alfredo-pasta/>
<https://www.onionringsandthings.com/penne-alfredo-broccoli/>
<https://slowcookergourmet.net/slow-cooker-chicken-alfredo-pasta/>
<https://www.shugarysweets.com/sausage-alfredo-pasta/>
<https://www.tablespoon.com/recipes/copycat-olive-garden-chicken-alfredo/417cc46f-cb50-4117-b720-65a3afb60d78>
<https://jessicainthekitchen.com/vegan-garlic-alfredo-pasta/>
<http://www.publix.com/aprons-recipes/mushroom-alfredo-pasta-with-sausage>
<https://thekitchengirl.com/leftover-ham-alfredo-pasta/>

The Command for used for facilitating the Crawling procedure in Apache Nutch is:

bin/crawl URLs/ Crawling/ http://localhost:8983/solr 3

where the parameters imply the following:

URLS: The directory containing the “seeds.txt” file which contains the seed URLs for crawling.

Crawling: The directory which stores the resultant directories generated by the Crawl procedure.

http://localhost:8983/solr: the URL link for the Solr host to which the crawled web pages and their contents are fed for indexing by Solr.

3: The number of iterations for which the Crawl script is executed. In every iteration, the Crawl script keeps crawling and fetching contents from unvisited URLs in the CrawlDB.

Apache Nutch generates 3 folder during the crawling operation:

1. **CRAWLDB:** it maintains the information about URLs such as the fetch status, fetching schedule, metadata, etc.
2. **LINKDB:** For each URL, the LINKDB maintains the incoming and outgoing URLs for that URL which are further used to facilitate PAGE RANKING algorithm and the HITS algorithm.
3. **SEGMENTS:** contains multiple subdirectories within it. During Crawling, the crawl script creates multiple directory to store information for Crawl Fetching, Crawl Content, Crawl Parsing, Parsed Data and Parsed Text.

The Crawling Method can be described by the following methods:

1. First Seed URLs are injected into the Crawl Database of Nutch. It maintains the list of URLs which are parsed by Nutch or are still in pending to be parsed.
2. Nutch next generates Segments for the injected Seed URLs. During the Crawl procedure, more than one segment may be generated. This helps Nutch determine which URLs have been crawled and parsed and maintains consistency with the REGEX filters.
3. Nutch begins Crawling and parsing the URL links in the Segments. It also stores information about the incoming and outgoing links in the URL.

4. When Nutch has finished crawling a set of URLs, these URLs are then added to the Crawl Database of Nutch. Also LINKDB is updated with the incoming and outgoing links of each URL.

To generate indexing, Page Rank and HITS algorithm, I collaborated with the student responsible for Indexing, PageRank and Hits. Next for implementing Page Rank and HITS algorithm, we shall require a Dump of the LINKDB in order to utilize the incoming and outgoing links of every URL for implementing these algorithms. The command used for creating the Dump is:

bin/nutch readlinkdb Crawling/LinkDB -dump LinkDBDUMP

This generates a Dump containing the incoming and outgoing links for each URL.

2. Indexing, PageRank and HITS – Yash Jain

2.1 Indexing

For indexing we used apache solr. After running the crawler, we get our output in three folders which are linkdb, crawl db and segments. Apache solr makes use of these 3 folders and generates the index of the documents

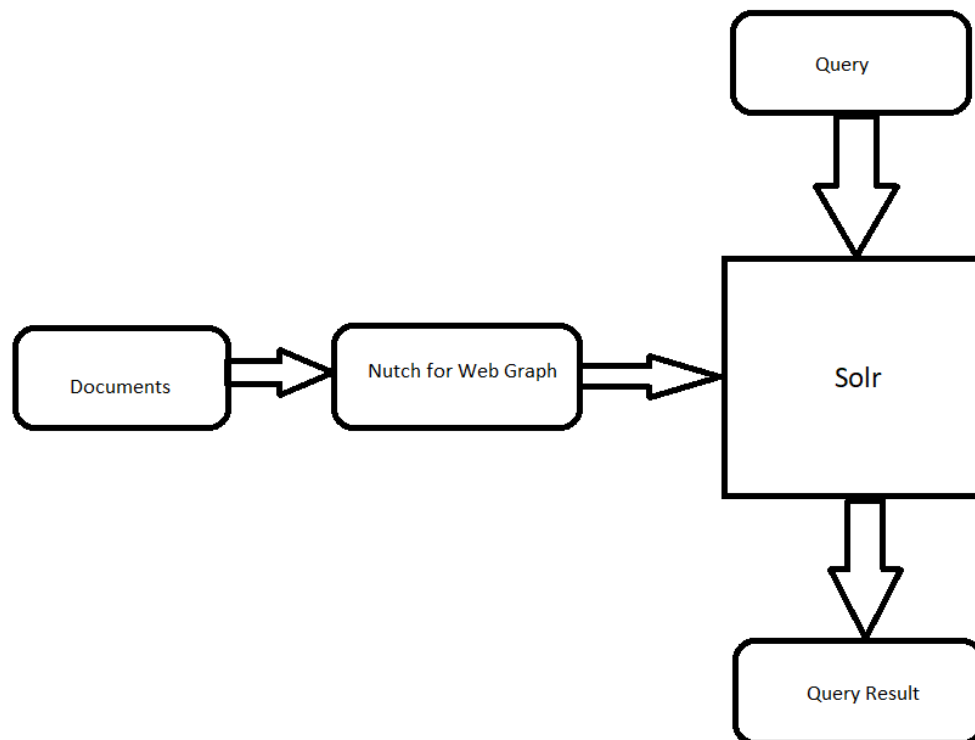
The commands used to feed the three folders (linkdb,crawl db and segments) is:

```
bin/nutch index -Dsolr.server.url=http://localhost:8983/solr crawl -linkdb crawl/linkdb -dir
crawl/segments
```

2.2 PageRank

To generate pagerank we used a webgraph which contains the number of inlinks , outlinks and the score of a given node using the pagerank algorithm. This webgraph is generated by nutch and is then fed into solr, which takes into account the TF-IDF score and the pagerank score in the webgraph to display the relevant results. (The damping factor here is 0.85)

The below figure shows the how recommendation takes place



The following steps were involved in recommending results:

1. Generate the webgraph using nutch. The command used is :
`bin/nutch org.apache.nutch.scoring.webgraph.LinkRank -webgraphdb crawl/webgraphdb`

2. Update the pagerank score of the crawldb, using the command:
`bin/nutch org.apache.nutch.scoring.webgraph.ScoreUpdater -crawldb crawl -webgraphdb crawl/webgraphdb`

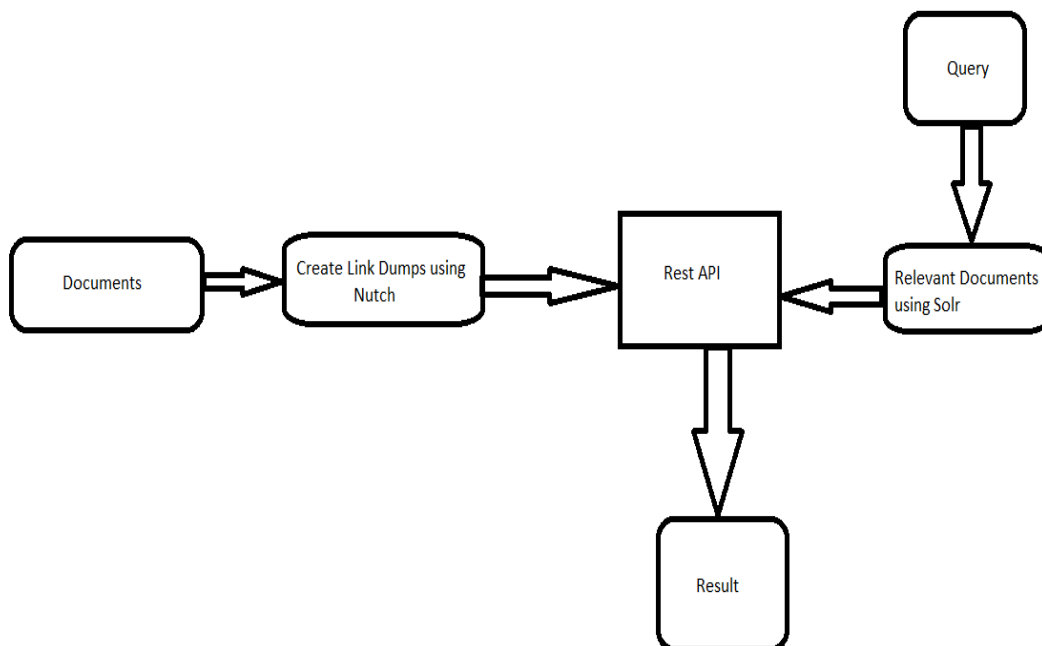
3. Index the documents using Apache solr using the command:
`bin/nutch index -Dsolr.server.url=http://localhost:8983/solr crawl crawl/crawldb -linkdb crawl/linkdb -dir crawl/segments`

2.3 HITS Score

For the HITS score we made our own rest API and wrote our own code in python to generate the authority and the hub score in python.

In python we used the **networkx library** to create the graph and ran the hits algorithm on the graph to generate the hub and the authority scores. To generate the graph we needed the inlinks and outlinks to a given document. To get these inlinks and outlinks we used nutch and generated a dump of the all the documents crawled. To do this we used the command:
`bin/Nutch readlinkdb crawl/linkdb -dump linkdbDump`

The below image shows how hits score is generated and how it is used to show the results



3. User Interface – Kanika Rawat

Technologies Used: HTML5, JavaScript, jQuery, CSS, Bootstrap, Java Servlets

3.1 Design:

The design part has three layers which are described below.

3.1.1 Presentation Layer (UI Layer):

The front end is rendered using HTML/CSS/Java script and it is powered by Java Servlets in the backend. We also used AJAX to make asynchronous calls to the back end. Bootstrap CSS library is used to design the page layout.

The search functionality enables to query for food recipes. We have shown the results of the query in five different tabs on the same page.

The webpage of our search engine has following components:

- Logo of the search engine
- Search text box with a search button
- Multiple tabs one each for showing results from various models.

The tabs to show results consist of -:

- *Google* – It shows results from Google using Google Search API
- *Bing* – Display results from Microsoft Bing Search API
- *Vector Space + Pagerank + HITS*: Results from HITS server after applying HITS over pagerank results from Solr
- *Vector Space + Pagerank + Flat Clustering*: Results from Clustering server after applying K-means clustering over Pagerank results
- *Vector Space + Pagerank + Query Expansion*: Results from Query Expansion server after expanding the query using relevance feedback

AJAX requests are used to send request to the server and then the responses are being handled inside JavaScript.

3.1.2 Application Layer (Controller layer):

The search requests from the user are handled by the application layer which accepts GET and POST requests via AJAX calls from UI. There are three controller programs written in Java which performs the handling of AJAX calls.

ServletHandler : Handles AJAX requests related to Google, Bing, Solr and HITS

ClusterServletHandler: Handles AJAX requests related to K-means clustering

QueryExpansionServletHandler: Handles AJAX request related to Query Expansion

3.1.4 DataAccessLayer:

The request that we get from the UI layer is routed to the corresponding server and then the result is given back to the Controller layer.

3.1.5 Business layer:

There are 3 servers running locally that contains the business logic for various relevance models.

Server 1: Solr server for handling indexing and Pagerank requests

Server 2: Python server for handling HITS/query expansion requests

Server 3: Python server for handling Flat Clustering requests

All requests from the Data Access layer are routed to these three servers. JSON is chosen as the data exchange format for ease of implementation and support by various programming languages.

3.2 Collaborations

3.2.1 Collaboration with student responsible for Indexing/Pagerank/HITS:

We have used Solr for indexing and pageranking. The server for indexing and Pagerank exposes an API for accessing the indexed results. The method creates the URL for a select query on Solr and fetches the output from the execution of that query in a JSON format.

I call Solr server directly using the information provided by the student responsible for indexing and Pageranking. The API also allows us to set some control parameters for e.g number of pages to be returned or format of the response. We have set the number of pages to be returned to be 15 and JSON format is used for the exchange of the data.

The following fields are selected for display in the UI:

- URL: the URL address of the page
- Title: Title of the page
- Snippet: the HTML content of the page (with HTML tags removed) – truncated to 200 characters for display in the UI as a short description about the webpage

The HITS algorithm is implemented in python and hosted in a separate server by this student. So the information about this python API was collected from this student to make the API call. Since the HITS algorithm is dependent on the indexing results from Solr, results from Solr API was sent to this API via POST and the response was collected back.

3.2.2 Collaboration with student responsible for Clustering:

The clustering algorithm KMeans has been implemented and we have maintained the clustering information in a file which contained url and the corresponding cluster number of that url. The top results from the solr are passed to the cluster based reordering algorithm. In the program, the most relevant cluster is identified and the documents which belong to the same cluster are brought up and the result are retrieved. The top 15 results are being shown on the UI based on the chosen cluster. The response is similar used for Solr.

3.2.3 Collaboration with student responsible for Query Expansion:

The query expansion algorithm is implemented and hosted in the same server as HITS. The input to this Query Expansion API is the initial query string made by the user and the initial Solr output. Final output from the API is the expanded query string. This expanded string is then used to retrieve a new set of results from the Solr API.

3.3 Comparison with Google and Bing:

In order to compare the results with Google and Bing results, Google web search API and Bing Search API were used to retrieve their results. Following observations were made during the evaluation:

- a. For certain queries which contained Food item names, our results from Pagerank/HITS/Clustering were closely in accordance with results from Google and Bing. We believe this is due to the selection of very good seed URLs to cover most of the food recipe possible. Also, from the type of pages that has been crawled, it is observed that the information about different food recipes is static information and has good keywords.
- b. For certain queries requesting food recipes with multiple ingredients, our search engine did not retrieve relevant pages in comparison to Google and Bing. Upon analyzing the results from Google and Bing with our document collection, it is observed that the websites that are popular for such information like allrecipes.com denied access to our crawler. Hence those pages were missing from our index.

- c. Using seed URLs from Google and Bing proved useful in retrieving the top pages during the search. Especially, Wikipedia pages were immensely useful in improving our search results and were retrieved by our search engine despite the limited number of inlinks and outlinks in comparison to the entire web used by Google and Bing. This proved the effect of Vector Space model combined with Pagerank.

3.4 Testing strategy:

Testing was done throughout the development of the project to provide immediate feedback to students implementing the relevance models. Approximately 200 queries were used to test the results of various models. About 50 queries about food item name were used in collaboration with the students building the relevance models and the rest of queries were generated based on various topics such as 'ingredient name + food item name' or 'food from different cuisine'.

When a particular page from the top results of Google/bing does not come up with our relevance model, the page was first checked for existence and if present, the page rank score of the page was compared with the top scoring page from our relevance model. In few cases, the score was too low due to insufficient inlinks and outlinks and in many cases, the page in question was not present in our collection.

In HITS and Query expansion testing, certain times, the algorithms did not converge within the given number of iterations and were communicated to the student in charge and algorithm parameters were appropriately improved.

3.5 Results

The queries for demonstration were chosen from the testing we have done and gave good results. Each query chosen was targeting a different area such different cuisine or ingredient name + food item or ingredient name or only food item.

Query 1: *Chocolate*

Top 4 results from our model (after clustering) :

1. <https://www.shugarysweets.com/chocolate-frosted-chocolate-cupcakes/>
2. <https://www.cookingclassy.com/chocolate-cake-chocolate-buttercream-frosting>
3. <https://www.bigbearswife.com/melted-chocolate-bunny-hot-chocolate/>
4. <https://www.barbarabakes.com/hersheys-perfectly-chocolate-chocolate-cake-and-perfectly-chocolate-chocolate-frosting>

Top 4 results from Google:

1. <https://en.wikipedia.org/wiki/Chocolate>

2. <https://www.chocolate.org/>
3. <https://www.chocolateangel.com/>
4. <https://www.chocolateangel.com/menu-2>

Top 4 results from Bing:

1. <https://en.wikipedia.org/wiki/Chocolate>
2. <https://www.chocolateangel.com/#!>
3. <https://www.ghirardelli.com/>
4. https://www.exploratorium.edu/exploring/exploring_chocolate/

Query 2: *chicken*

Top 4 results from our model (After Pagerank):

1. <https://rasamalaysia.com/recipes/chicken-recipes/#>
2. <http://delightsofculinaria.com/recipes/chicken-recipes/>
3. <https://www.cookstr.com/Chicken-Recipes/Sylvies-Chicken>
4. <https://easyfamilyrecipes.com/category/recipes/chicken-recipes/>

Top 4 results from Google:

1. <https://en.wikipedia.org/wiki/Chicken>
2. <https://www.allrecipes.com/recipes/201/meat-and-poultry/chicken/>
3. <https://www.foodnetwork.com/topics/chicken>
4. <https://www.myrecipes.com/chicken-recipes>

Top 4 results from Bing:

1. <https://www.chick-fil-a.com>
2. <https://www.allrecipes.com/recipes/201>
3. <https://en.wikipedia.org/wiki/Chicken>
4. <https://www.foodnetwork.com/topics/chicken>

Query 3: *eggs*

Top 4 results from our model (after query expansion):s

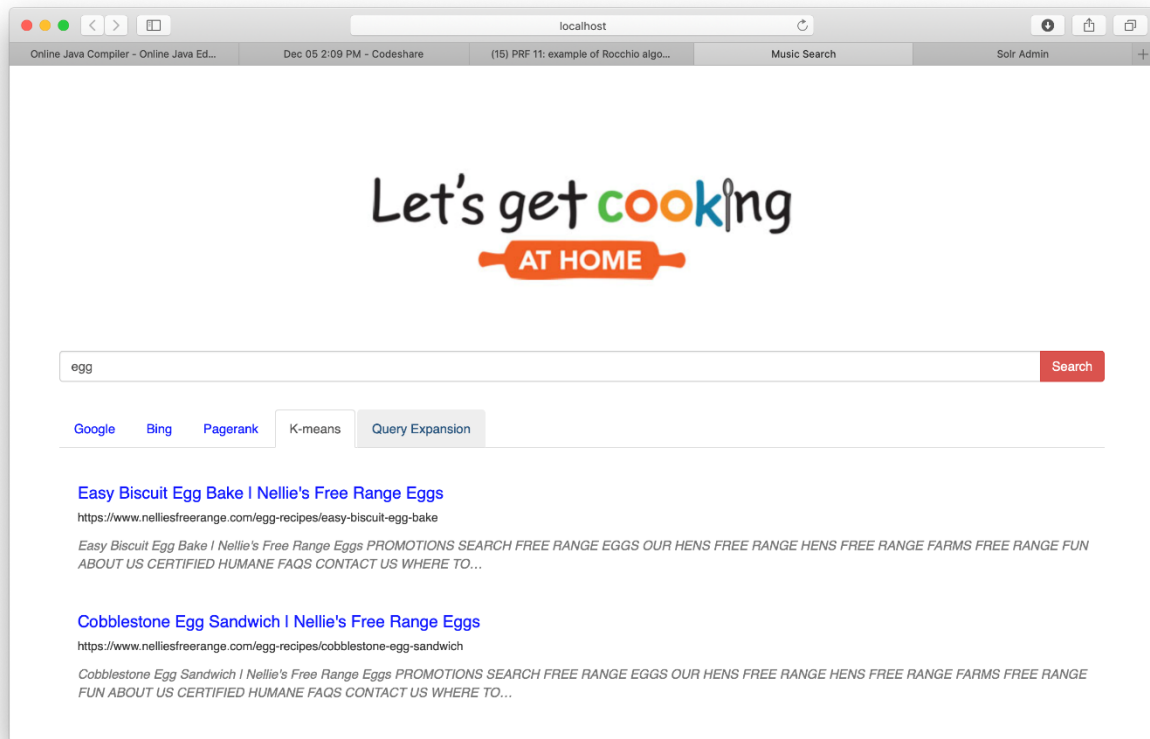
1. <https://www.cookingclassy.com/hot-fudge-cake/>
2. <http://thewoodandspoon.com/hot-fudge-sundae-cake/>
3. <https://www.cookingclassy.com/hot-fudge-cake/>
4. <https://www.foodtasticmom.com/peppermint-hot-fudge-cake/>

Top 4 results from Google:

1. <https://www.incredibleegg.org/egg-nutrition/>
2. <https://www.webmd.com/food-recipes/ss/slideshow-eggs-6-ways>
3. <https://www.foodnetwork.com/topics/eggs>
4. https://en.wikipedia.org/wiki/Egg_as_food

Top 4 results from Bing:

1. www.muscleeegg.com
2. [https://en.wikipedia.org/wiki/Egg_\(food\)](https://en.wikipedia.org/wiki/Egg_(food))
3. <https://www.foodnetwork.com/topics/eggs>
4. <https://www.healthline.com/nutrition/10-proven-health-benefits-of-eggs>



4. Clustering – Shraddha Bang

Performed K-Means clustering with k as 12.

In order to perform clustering of html documents, urls of the documents and its contents were required. These details are fetched from solr, parsed and subjected to K Means clustering .On clustering based on contents in the documents, each URL will be uniquely assigned to one cluster. The url and the cluster number of every document is stored in a flat file before searching starts.

Clustering identifies documents which are similar based on their content. Hence documents in one cluster are similar within themselves and much different from documents in other clusters.

The results of clustering are incorporated in the relevance models. Based on results from solr which are ordered on the basis of higher page ranks, documents are further evaluated for their clusters. The most relevant cluster is identified and documents which belong to the same cluster are pushed up when clustering results were showed up. We had like 341983 records created after crawling. The above files created were used for clustering purpose. We have used flat clustering method and its k -means clustering

K Means Algorithm Description

Given a set of observations (x_1, x_2, \dots, x_n) , where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k ($k \leq n$) sets $S = \{S_1, S_2, \dots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS) (sum of distance functions of each point in the cluster to the K center). In other words, its objective is to find:

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

where μ_i is the mean of points in S_i .

Implementation of K-means Algorithm:

- In order to implement the clustering of URLs parsed we needed the contents of each URL parsed. The contents of each URL parsed in the segments folder of Nutch can be dumped by using readseg command of Nutch .The command is as follows:
Command pattern: /Nutch readseg -dump <segment_dir> <output>
Command used: /Nutch readseg -dump test/segments/ (segment Num)
segmentParsedDump

- As while crawling we configured it for 3 runs it generated 3 segment folders .So we ran the above command for each segments to create the dump. But as those dump was not in a good format to be used we had to preprocess it for further use. So I wrote a python program to read the dump URL and its content in a file .The file content looked as follows:
URL: <https://www.allrecipes.com/recipes/201/meat-and-poultry/chicken/>
Parsed Content :: <Contents>
- We have like 156879 records created at the end of this process. The above file created was used for clustering purpose.
- This file was fed as input to the clustering module. Output of this module is a file which contains the URL and the respective cluster it belongs to.
- Accessing Clustering Information: So for all the urls in the result we collected the cluster id associated with it. For each cluster id generated we have maintained a list of urls near the centroid so we just include those results in our result set if they are not there.
- Now I first retrieved the query results from the pageranking. The top 100 of the pageranking results are checked with the clustered results by comparing how many of these webpages belonged to same clusters and we consider the cluster which contains the maximum webpages and display 15 of the webpages on to the UI. Sklearn library was used to generate clusters and implement K-Means

Choosing – K

The correct choice of k is often ambiguous, with interpretations depending on the shape and scale of the distribution of points in a data set and the desired clustering resolution of the user. In addition, increasing k without penalty will always reduce the amount of error in the resulting clustering, to the extreme case of zero error if each data point is considered its own cluster. Better results were achieved when K value was 12.

Pseudo code for Extracting contents from index files of solr

1. Specify path to file and read contents from file
2. While until there are no more documents
 - a. Read every record and look for 'id' section and save it
 - b. Extract the details in 'content' section and save it
 - c. Process the content to remove html tags
 - d. Save a record with <id> , <content>
3. End the program

Pseudo code for performing K means clustering

1. Read the file with records saved in format <id> , <content>
2. While Until there are no more records in file

- a. Perform k means clustering on the content of data with k value of 12 and divide the content into clusters.
 - b. Save the clusters with <id> , <cluster number> format
3. End the program

Testing

20 queries were used to test the results of clustering. These are the queries I used.

Sample queries

milkshake

- <http://foodviva.com/milkshake-recipes/vanilla-milkshake-recipe/>
- <http://foodviva.com/milkshake-recipes/malted-milkshake-recipe/>
- <http://foodviva.com/milkshake-recipes/peppermint-milkshake-recipe/>
- <http://foodviva.com/milkshake-recipes/chocolate-milkshake-recipe/>

alfredo pasta

- <https://whatsgabycooking.com/cauliflower-alfredo-pasta/>
- <https://cravinghomecooked.com/easy-pasta-alfredo/>
- <https://prettysimplesweet.com/mushroom-alfredo-pasta/>
- <https://anitalianinmykitchen.com/alfredo-pasta/>

chicken

- <https://mexicanfoodjournal.com/chicken-tinga/chicken-tinga-ingredients/>
- <https://iwashyoudry.com/category/main-ingredient/chicken-chicken/>
- <https://mexicanfoodjournal.com/chicken-tinga/chipotle-chicken-tinga/>
- <https://www.cookstr.com/Chicken-Recipes/Sylvies-Chicken>

Chocolate cake

- <https://www.shugarysweets.com/chocolate-frosted-chocolate-cupcakes/>
- <https://www.cookingclassy.com/chocolate-cake-chocolate-buttercream-frosting>
- <https://www.bigbearswife.com/melted-chocolate-bunny-hot-chocolate/>
- <https://www.barbarabakes.com/hersheys-perfectly-chocolate-chocolate-cake-and-perfectly-chocolate-chocolate-frosting>

The criteria for selecting the food queries are based on popularity of food.

After clustering top ranked result pages belonging to same cluster were displayed and it was relevant

Collaboration with other students:

- 1 – I collaborated with Rutuj, who was responsible for crawling, to get the content of the urls from segments.
- 2 – I collaborated with Yash, who was responsible for indexing and page ranking, to get the results from page rank algorithm.
- 3 – I collaborated with Kanika, who was responsible for GUI, to get the query from UI, and send back the results from clustering module.


5. Query expansion and relevance feedback – Ruchita Deshmukh

Rocchio Algorithm

The following algorithm was used to test the Rocchio Algorithm:

- Used in practice:

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

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

Following was the approach used to arrive to a set of weights that fetches a more relevant modified query:

1. Get the query and the results from Solr for the results for the query
2. Fetch the the results from Google
3. Label each result from our search query as either relevant for non-relevant based upon the results of Google.
4. Increment beta, decrement gamma keeping alpha constant to get higher ratio.
5. Repeat step 4 for a fixed number of iterations or until weights stabilize
6. Output the modified query from the query vector

The 8 queries selected to test the algorithms:

The queries were selected such that they are either meats, ingredients, dishes from different cultures or drinks or desserts.

This covers a wide range of food. Recipes and shows the coverage of our search engine.

Eggs	Alfredo Pasta	Chocolate	paneer
Fish	Khichdi	milkshake	Paratha

The results were not promising and a universal set of weights could not be found. Also the running time for convergence was too high. Because of this we shifted to pseudo relevance feedback.

Pseudo relevance feedback:

Here we use local feedback strategies to expand the query with terms correlated to the query terms.

We can find those co-related terms from the local clusters build from the local document set.

Three types of clustering are:

1. Association cluster:

The idea is that stems which co-occur frequently inside documents have a synonymity association

2. Metric cluster:

The idea is that the stems which occur far apart in the document are less co-related to the terms that occur closer like in the same sentence.

3. Scalar cluster:

The idea is that two stems are more co-related if they have similar neighborhood.

The algorithm for this is for each stem in the query add the closest neighbors from the cluster (association /metric /scalar)

Some query expansion results with metric clusters

Initial Query

Expanded Query

chocolate	cake hot chocol
Alfredo pasta	alfredo pasta garlic easi
fish	fish tag seafood
eggs	hen egg farmer
khichdi	rice khichdi sabudana
paneer	paratha achari paneer
paratha	plain palak paratha
milkshake	milkshak vanilla banana
chinese	food chines china

Some challenges:

1. Scalar clustering took too long to run so we can't perform them on the fly
2. The query words need to spell correctly so as to get the correct stems.

Collaboration with GUI:

1. The API calls were made to run the query expansion.
2. The input for the API call were the initial query and SOLR output.
3. After the query expansion the expanded query was returned as an output to the API call.

Coding:

1. Python was the language selected to implement.
2. Pythons NLTK WORNET Porter stemmer was used for stemming.

Project Demo Selection:

For the demo,

For the demo, the query “chocolate” was used since it is a very common ingredient in different dishes

The algorithm used is metric clustering