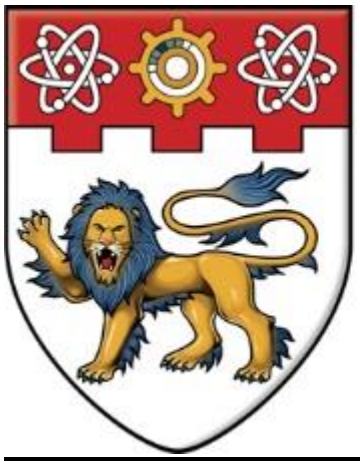

NANYANG TECHNOLOGICAL UNIVERSITY

CZ4034 Information Retrieval Assignment Report

Assignment Group	17
Group Member	**Matriculation Number**
Huang Jian Wei	U1521567A
See Xin Yee	U1520918B
Lim Zi Yang	U1522218E
Ivan Teo Wei Jing	U1421071L
Yong Guo Jun	U1440217C

School of Computer Science and Engineering

2017/2018

Table of Contents

List of Figures

Background

It is hard to talk about food nowadays without talking about Instagram.

For the past few years, food trends have become more focused on what's “instagrammable”. Chefs, social media influencers, and marketing managers believe that for better or worse, Instagram has influenced how we eat. According to research by Zizzi, 18 to 35-year olds spend five whole days a year browsing food images on Instagram, and 30 per cent would avoid a restaurant if their Instagram presence was weak. In fact, it’s now normal to sit down in a restaurant having already decided what you are going to order because you’ve spent a few minutes stalking on Instagram in advance.

Whether you like it not, people do flock to certain places from all over just because dishes are undeniably “insta-famous”.

Motivation

Having a search engine to purely search Instagram for restaurants around Singapore can not only satisfy someone who is purely looking for new and/or popular places to eat, it can also satisfy a social media fiend in need of some new content, or even help restaurants know which kinds of food are in-trend currently and allow them to innovate new food which can help boost their business by “following the trend”, or allow them to track the popularity of the food they have in their menu.

Therefore, we decided to develop an information retrieval system to retrieve Instagram posts from more than 130 local restaurants’ Instagram accounts. By using this system, users can search for food of different types of cuisine, sorted according to their date posted as well as popularity.

Objectives

For this assignment, we are required to complete the following tasks:

1. Crawl a text corpus of interest
2. Build a search engine to query over the corpus
3. Performing text classification and clustering

Crawling a text corpus of interest

The first goal is to crawl **Instagram** for relevant posts from **restaurants in Singapore** and pre-process the information before indexing the documents to Solr.

Build a search engine to query over the corpus

The second goal is to create a web search engine based on the data stored in Solr. The web search engine provides a front-end user interface for users to **find food or restaurants of their preference**. Our group will explore innovative methods in enhancing the speed of search queries and ranking to, hopefully, suit every user's needs.

Performing text classification and clustering

The third goal is to perform classification on the collected information to identify interesting patterns which might provide initially unseen trends of information. This presents a **logical categorisation of the latest food trends among other Instagram users** to the user.

Limitations

Instagram moves the location of its media from time to time, hence some URL linking to a specific post or photo may be broken.

The attributes retrieved from Instagram is directly based on the crawler we used. This also applies a constraint on our Weka Classification Task.

1. Crawling

1.1 How you crawled the corpus (e.g. source, keywords, API, library) and stored them (e.g. whether a record corresponds to a file or a line, meta information like publication date, author name, record ID)

First, we searched online for Instagram accounts of restaurants located in Singapore and note down the various information regarding each restaurant in an Excel file called "ListOfRestaurants.csv". The information stored includes:

- Name of restaurant
- Instagram username
- No. of posts
- No. of followers
- No. of people following
- Type of cuisine

	A	B	C	D	E	F	G
1	Name of Restaurant	iG Account	No. Of Post/Records	No. Of Followers	No. Of Following	Remarks	Type of Cuisine
2	burgermonster.sg	burgermonster.sg	387	250	149	burgers	American
3	25 Degrees in Singapore	25degreesinsingapore	296	1611	381	Burgers	American
4	The Market Grill	themarketgrillsg	287	937	163	Burgers, Steak, Lobster Rolls, S	American
5	littlediner	littlediner	197	415	1233	Comfort American food	American
6	KFC	kfc_sg	956	27.3k	15	Food	American
7	everything with fries	ewf_sg	91	423	72	Western	American
8	The Boiler	theboilersg	377	1393	159	Seafood and Beer	American
9	Bedrock Bar & Grill	bedrocksg	147	993	1657	Steak	American
10	Meat N Chill	meatnchill	156	1128	1820	steakhouse	American
11	Morganfield's Singapore	morganfieldssingapore	526	2477	1612	American Barbecue and Grill	American
12	IZY Dining and Bar	izysingapore	290	623	244	American-Japanese izakaya	American/Japanese
13	Bochinche	bochinchesg	531	1750	884	Argentine cusine	Argentinian
14	PappaRich Singapore	papparichsg	383	646	482	Authentic Malaysian delights	Asian
15	Nine Fresh 九鮮	ninefresh	237	3607	16	taiwanese taro ball desserts	Asian
16	Sawadee Thai Cuisine	sawadeethaicuisine	101	683	44	thai food	Asian
17	talaykata	talaykata	81	204	3	Thai Seafood BBQ	Asian
18	Rumah Rasa SG	rumahrasasg	80	3290	101	halal indonesian restaurant	Asian
19	Folklore Singapore	folkloresg	70	591	17	Tales of Singapore's heritage th	Asian
20	Coriander Leaf	coriander_leaf	207	360	36	Traditional and interpreted dis	Asian
21	Elemen 元素	elemensg	204	675	281	vegetarian	Asian
22	Sea Tripod Seafood Paradis	seatripodsg	52	96	194	abalone, prawns, scallops, sea	Asian
23	state land café	statelandcafe	329	5140	3041	café	Asian
24	Say Chiizu Singapore	saychiizu.sg	10	1637	36	Hokkaido stretchy cheese toast	Asian
25	Modus	modussg	20	195	25	quinoa bowl	Asian
26	Pimp My Salad	pimpmysalad_sg	273	936	395	salad bar	Asian
27	Paddy Hills	paddyhills.sg	316	3567	2	Specialty Coffee	Asian/Australian/Japanese
28	The Halia	thehalia	389	2443	491	Food, people	Asian/European
29	Bincho at Hua Bee	binchosg	652	2108	761	Yakitori Restaurant & Cocktail B	Asian/Japanese
30	Rokeby	rokeby_bistro	175	631	216	Australian inspired bistro w/ A	Australian
31	RWS Dining Artisans	rwsdiningartisans	512	1253	164	innovative contemporary Austr	Australian
32	The Lokal Singapore	thelokalsingapore	1715	3932	174	australian	Australian
33	Burnt Ends	burntends_sg	306	16.3k	489	Modern Australian BBQ	Australian
34	Salt tapas & bar	salttapasandbar	205	924	449	A modern tapas bar with a twis	Australian/European/Mediterranean

Figure 1.1: ListOfRestaurant.xlsx

Next, we wrote a program with the help of an open-source Instagram crawler API by Raiym (https://packagist.org/packages/raiym/instagram-php-scraper) to crawl Instagram, looping through the list of restaurant Instagram accounts that we have found previously (in Figure 1.1). We experimented with several Instagram APIs before settling on the current one.

Using the program that we wrote, we crawled more than 10000 records of posts from over 130 accounts and stored the results in file called “output.csv”.

The figures below show the script we used to crawl the data from Instagram Web itself and the initial data stored in output.csv.

```
for($counter=0;$counter<count($igAccount);$counter++){
$medias = $instagram->getMedias($igAccount[$counter],100);
$data = [];
foreach ($medias as $value) {
    $update = $client->createUpdate();
    $doc = $update->createDocument();

    $url = $value->getLink();                              //Post Url
    $media = $instagram->getMediaByUrl($url);              //Post Image
    $media = $media->getImageHighResolutionUrl();
    $noOfLikes = $value->getLikesCount();                  //No. Of Likes for the Post
    $noOfComments = $value->getCommentsCount();            //No. Of Comments
    $createTime = $value->getCreatedTime();                //DateTime of Post
    $caption = $value->getCaption();                       //Caption of the Post
    $caption = str_replace(",","-",$caption);
    $category = $restaurantType[$counter];                 //Cuisine Category
    $account = $igAccount[$counter];                       //Owner of the Instagram post
    $followers = $noOfFollowers[$counter];
    $data[] = $url.','.$media.','.$noOfLikes.','.$noOfComments.','.$createTime.','
    .$caption.','.$category.','.$account.','.$followers;//Data of each Post

}
```

Figure 1.2: Information crawled from Instagram

	A	B	C	D	E	F	G	H	I
1	IGPost	IGPicture	No. Of Like	No. Of Comments	Date Posted	Captions	Category	IgAccount	No.Of.Followers
2	https://www.instagram.c	https://instagram.f	9	1	1520417870	Help yourselves with Burger	American	burgermo	250
3	https://www.instagram.c	https://instagram.f	7	0	1520417845	Help yourselves with Burger	American	burgermo	250
4	https://www.instagram.c	https://instagram.f	7	0	1520417814	Help yourselves with Burger	American	burgermo	250
5	https://www.instagram.c	https://instagram.f	10	0	1519816553	Pair up your favorite set	American	burgermo	250
6	https://www.instagram.c	https://instagram.f	6	0	1519816531	Pair up your favorite set	American	burgermo	250
7	https://www.instagram.c	https://instagram.f	6	0	1519816511	Pair up your favorite set	American	burgermo	250
8	https://www.instagram.c	https://instagram.f	5	0	1519816474	Pair up your favorite set	American	burgermo	250
9	https://www.instagram.c	https://instagram.f	8	0	1519816452	Pair up your favorite set	American	burgermo	250
10	https://www.instagram.c	https://instagram.f	8	0	1519816424	Pair up your favorite set	American	burgermo	250
11	https://www.instagram.c	https://instagram.f	13	0	1519707535	Burger Monster. One of the	American	burgermo	250
12	https://www.instagram.c	https://instagram.f	15	1	1519707514	Burger Monster. One of the	American	burgermo	250
13	https://www.instagram.c	https://instagram.f	12	0	1519707496	Burger Monster. One of the	American	burgermo	250
14	https://www.instagram.c	https://instagram.f	12	0	1519626963	Get to try this humongous	American	burgermo	250
15	https://www.instagram.c	https://instagram.f	12	0	1519626942	Get to try this humongous	American	burgermo	250
16	https://www.instagram.c	https://instagram.f	12	0	1519626925	Get to try this humongous	American	burgermo	250
17	https://www.instagram.c	https://instagram.f	9	0	1519626896	Get to try this humongous	American	burgermo	250
18	https://www.instagram.c	https://instagram.f	9	0	1519626880	Get to try this humongous	American	burgermo	250
19	https://www.instagram.c	https://instagram.f	7	0	1519626861	Get to try this humongous	American	burgermo	250
20	https://www.instagram.c	https://instagram.f	10	0	1519626827	Get to try this humongous	American	burgermo	250
21	https://www.instagram.c	https://instagram.f	10	0	1519626806	Get to try this humongous	American	burgermo	250
22	https://www.instagram.c	https://instagram.f	7	0	1519626770	Get to try this humongous	American	burgermo	250
23	https://www.instagram.c	https://instagram.f	6	0	1519284340	Tasty burgers for as low as	American	burgermo	250
24	https://www.instagram.c	https://instagram.f	10	0	1519284319	Tasty burgers for as low as	American	burgermo	250
25	https://www.instagram.c	https://instagram.f	11	0	1519284303	Tasty burgers for as low as	American	burgermo	250
26	https://www.instagram.c	https://instagram.f	15	0	1519215605	The bigger the better.	American	burgermo	250
27	https://www.instagram.c	https://instagram.f	14	0	1519215536	One of our customers	American	burgermo	250
28	https://www.instagram.c	https://instagram.f	15	0	1519215374	The ultimate cheat day for	American	burgermo	250
29	https://www.instagram.c	https://instagram.f	18	2	1519006777	Plain but tasty.	American	burgermo	250
30	https://www.instagram.c	https://instagram.f	11	0	1519006025	Traditional marinated beef	American	burgermo	250
31	https://www.instagram.c	https://instagram.f	12	0	1519005135	Kimchi Fries	American	burgermo	250
32	https://www.instagram.c	https://instagram.f	15	0	1518587656	We arenâ€™t just known for	American	burgermo	250
33	https://www.instagram.c	https://instagram.f	18	0	1518587499	Satisfy your seafood craving \	American	burgermo	250
34	https://www.instagram.c	https://instagram.f	10	0	1518587400	Tasty chicken meat with	American	burgermo	250
35	https://www.instagram.c	https://instagram.f	13	0	1518410233	Have some of our Fried Shish	American	burgermo	250
36	https://www.instagram.c	https://instagram.f	11	0	1518410037	Your ball of happiness!	American	burgermo	250
37	https://www.instagram.c	https://instagram.f	9	0	1518409910	Have some of Burger Monste	American	burgermo	250
38	https://www.instagram.c	https://instagram.f	13	0	1518056407	Gong Xi Fa Cai!ðŸ®ðŸ”´	American	burgermo	250

Figure 1.3: output.csv

The information of an Instagram Post which are stored in Solr were as follow:

- url of post
- url of photo
- number of likes for photo
- number of comments for photo
- date-time of post
- caption of post
- category of restaurant
- restaurant's Instagram account username
- restaurant's Instagram account number of followers

```
$handle = fopen("output.csv", "r");
while (($data = fgetcsv($handle, 10000, ","))
!== FALSE) {
    $num = count($data);
    $update = $client->createUpdate();
    $doc = $update->createDocument();

     $row++;
    $doc->id = $count;
        if($data[0]){
            $doc->IGPost = $data[0];        //URL of Post
        }
        if($data[1]){
            $doc->IGPicture = $data[1];     //URL of Photo
        }
        if($data[2]){
            $doc->Likes = $data[2];         //No. Of Likes of photo
        }
        if($data[3]){
           $doc->NoOfComments = $data[3];  //No. Of Comments for photo
           }
       if($data[4]){
           $doc->dateTime = $data[4];      //Date Time of Post
           }
       if($data[5]){
       $doc->Caption = $data[5];           //Caption of the photo
           }
       if($data[6]){
           $doc->Category = $data[6];      //Category of Cusine
       }
       if($data[7]){
           $doc->Account = $data[7];       //Restaurant's IG account username
       }
       if($data[8]){
           $doc->Followers = $data[8];     //Restaurant's IG account no. of followers
       }
       $update->addDocument($doc);
       $update->addCommit();

   // this executes the query and returns the result
       $result = $client->update($update);
```

Figure 1.4: Indexing data to Solr using Solarium

We then used Solarium, an open-source Solr Client Library for PHP, to begin indexing our ~~csv~~ data in the output.csv file into our Apache Solr.

Solarium also provides functions to execute queries to Solr, as we will see in the later part of the report.

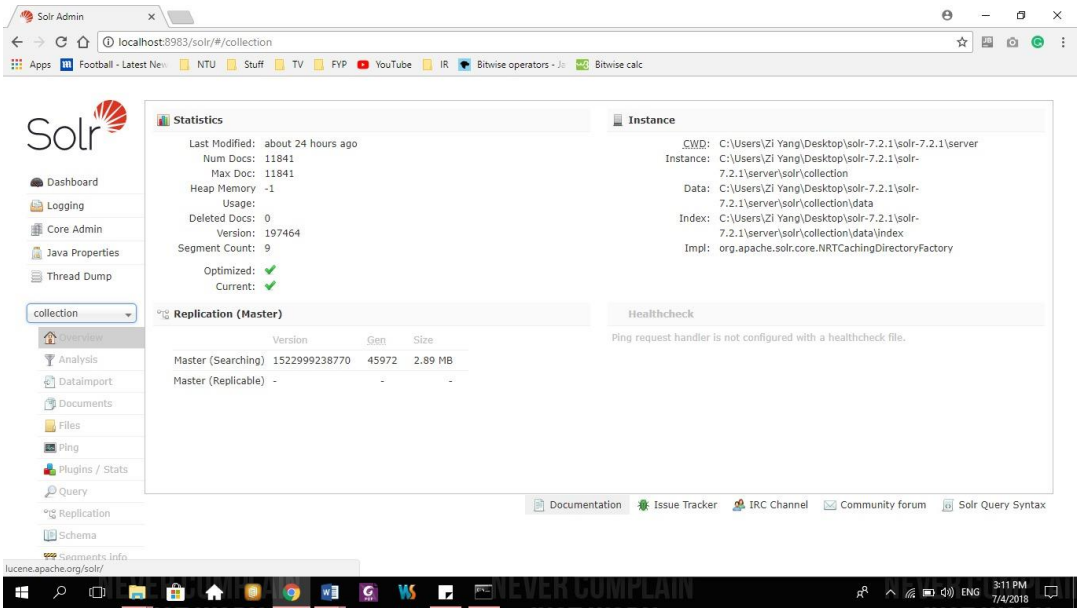

Figure 1.5: Solr populated with data

During Solr Index time, we also make configured the default stopword.txt and added in our stop words. This will improve the accuracy and speed during query time.

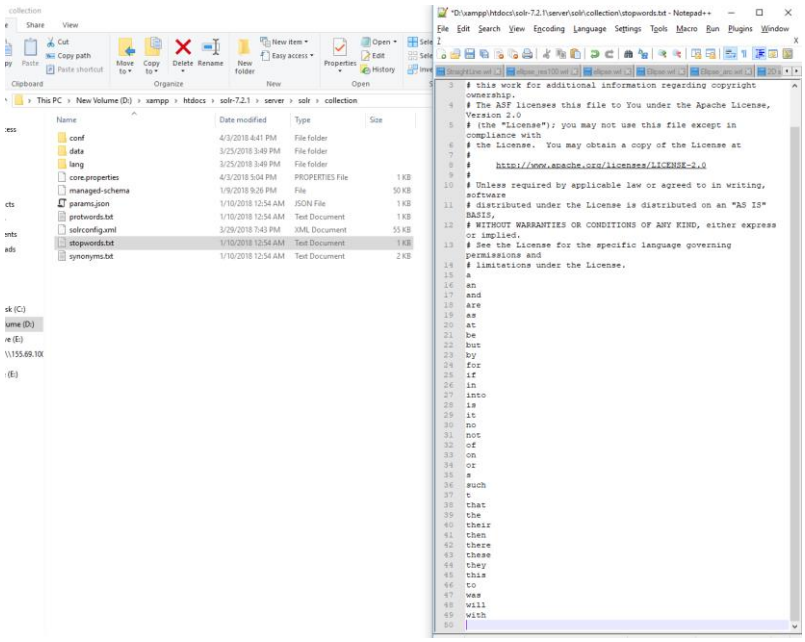

Figure 1.6: stopwords.txt

1.2 What kind of information users might like to retrieve from your crawled corpus (i.e., applications), with example queries

Our project's goal is to retrieve photos from local restaurants related to user's query from Instagram. Users would thus want to retrieve photos with its captions related to their query. For example, a user might query for "chilli crab" to retrieve the photos and captions related to "chilli crab" to aid in his or her decision in choosing a restaurant.

Some other types of queries that users might like to retrieve are listed below:

- posts related to fried chicken in Korean cuisines
- posts related to coffee that has more than 200 likes
- posts related to pancake dated from most recent to least recent
- posts related to pancake dated from most likes to least likes
- 20 posts related to breakfast

Example queries:

- Input "fried chicken" and select "Korean" under the "Type of Cuisine:" filter
- Input "coffee" and select ">200" under the "Minimum Likes:" filter
- Input "pancake" and select "Date (Most Recent to Least Recent)" under the "Sort By:" filter
- Input "pancake" and select "Popularity (Most Likes to Least Likes)" under the "Sort By:" filter
- Input "breakfast" and select "20" under the "Results Returned:" filter

1.3 The number of records, words, and types (i.e., unique words) in the corpus

The table below represents the number of words for records, words and unique words.

Number of records	14275
Number of words	548490
Number of unique words	27014

The figure below represents some of the most common words and the number of it found in the corpus.

Word	**Frequency**
us	2838
singapore	2317
sgfood	2174
new	1724
day	1495
Food	1427
Igsg	1385
today	1332
available	1305
dinner	1301

2. Indexing and querying

2.1 Build a simple Web interface for the search engine (e.g., Google)

A simple web interface has been designed to cater to the searching of Instagram posts of restaurants in Singapore. We used HTML, Javascript, Jquery, and PHP to build this interface. Below are three figures that show the web interface design.

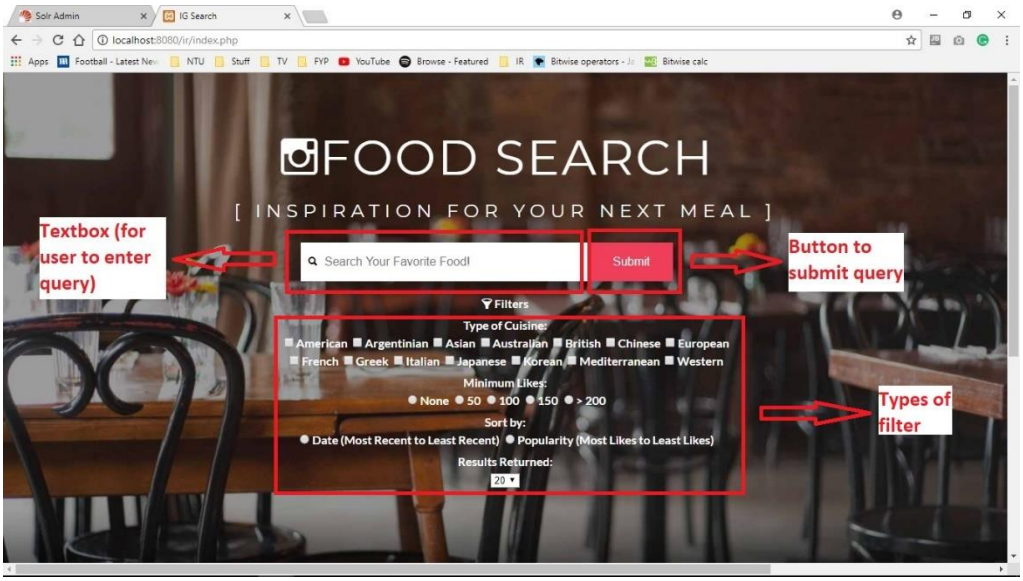

Figure 2.1: Search Engine Home Page

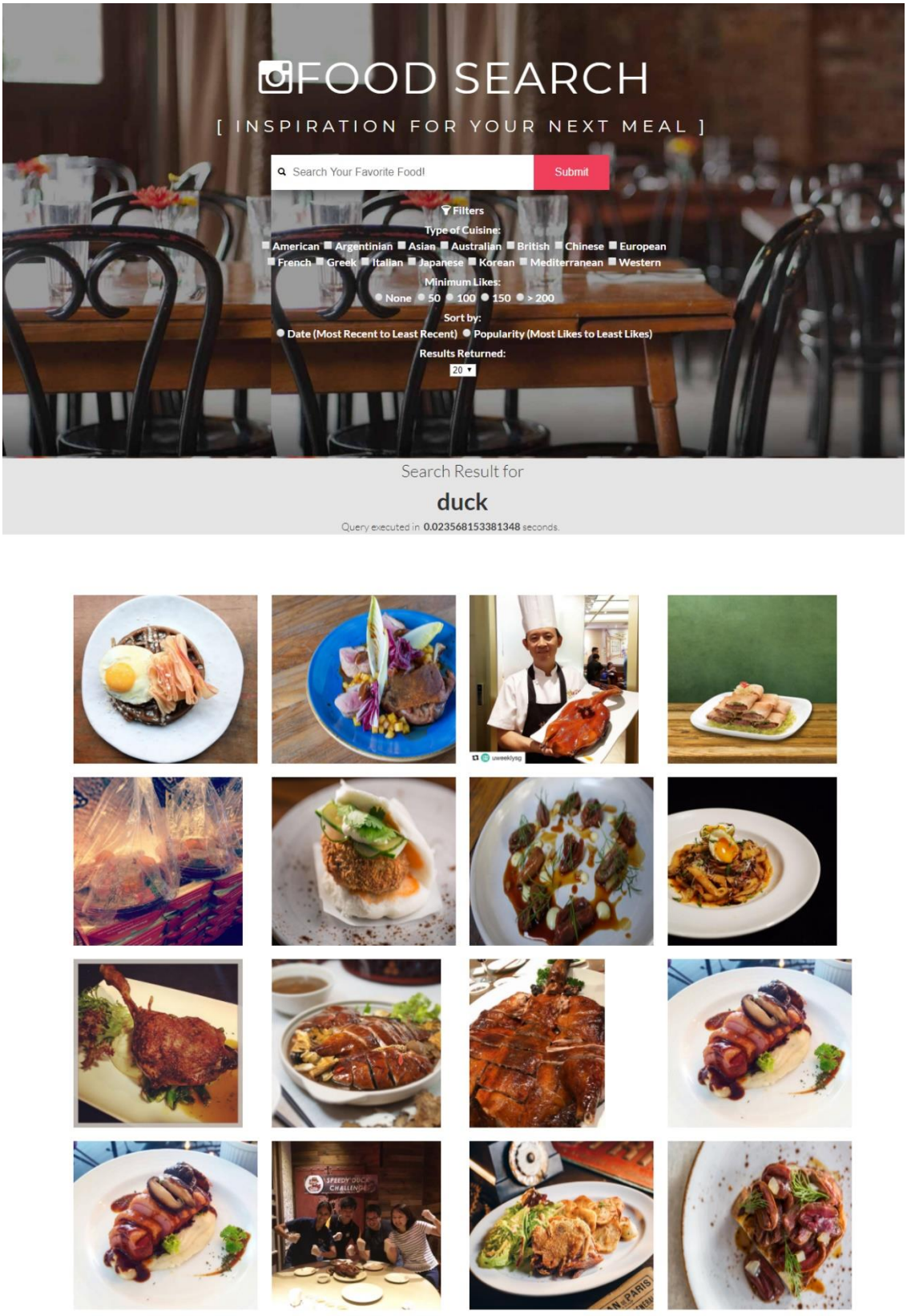

Figure 2.2: Web Interface populated with Results and Query Time

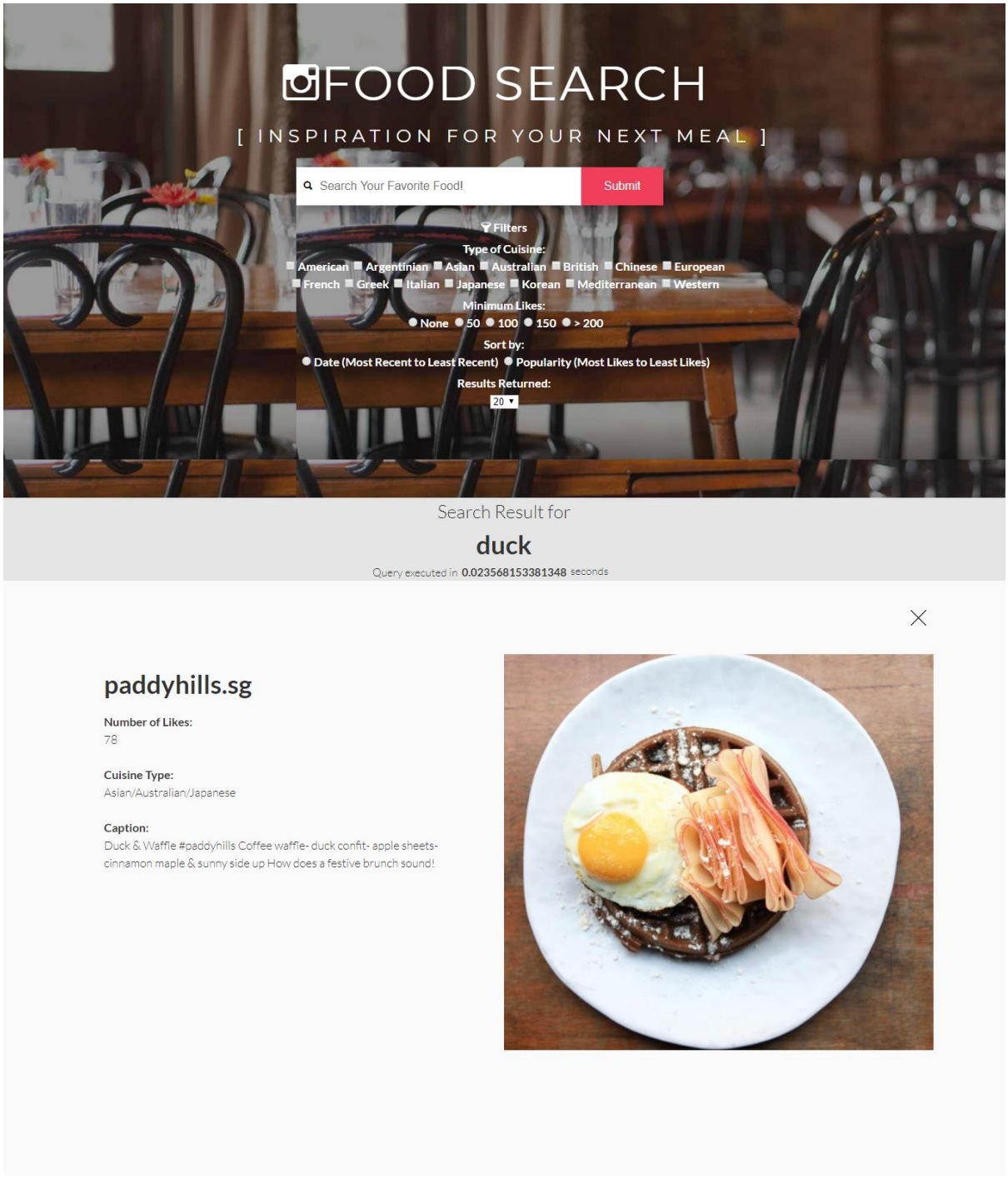

Figure 2.3: When user clicks on a photo, details are shown

2.2 A simple UI for crawling and incremental indexing of new data would be a bonus (but not compulsory)

We decided to build a separate interface for incremental indexing and crawling of new data. To add more data directly into Solr, one can enter the **account username of the account** and the **number of posts that they would like to crawl**. On submitting, the identified posts from the user's account will be crawled and indexed on Solr, adding to the available data. Users can then have access to these posts on our web interface in Figure 2.4.

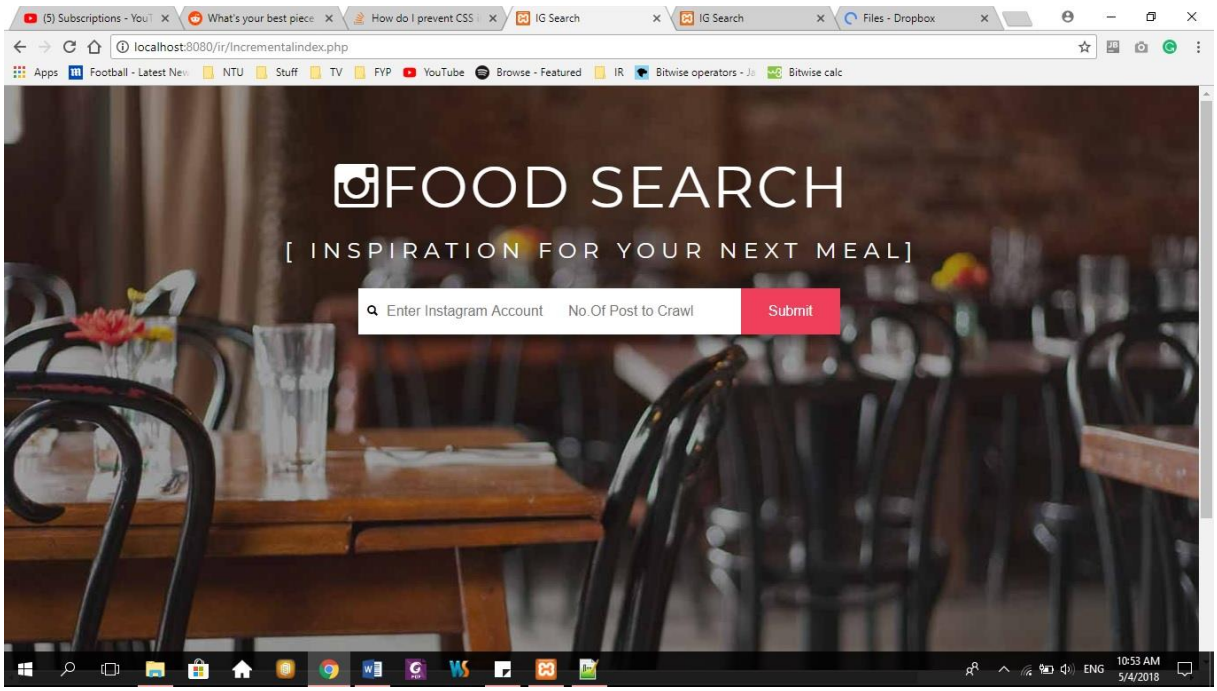

Figure 2.4: UI for crawling and incremental indexing of new data

2.3 Write <u>five</u> queries, get their results, and measure the speed of the querying

2.3.1 Query 1

Find posts on BBQ which are of Korean type and post must have at least 50 likes. Sort posts from most likes to least likes.

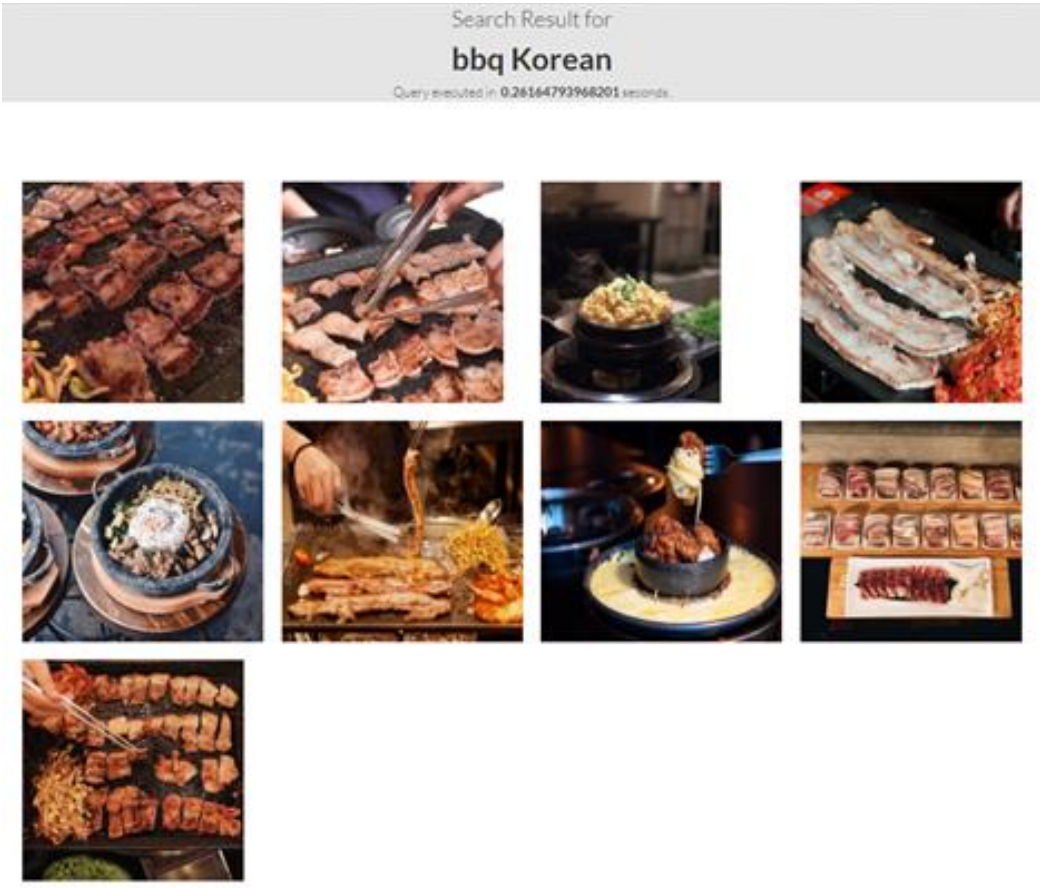

Figure 2.5: Query on "bbq Korean"

Results Returned: 9

Query Time: 0.261 seconds

2.3.2 Query 2

Find posts on seafood which are of Asian type

Figure 2.6: Query on "seafood" of Asian cuisine category

Results Returned: 2

Query Time: 0.15 seconds

2.3.3 Query 3

Find posts on salmon with more than 200 likes. Sort posts from most recently posted to least recently posted.

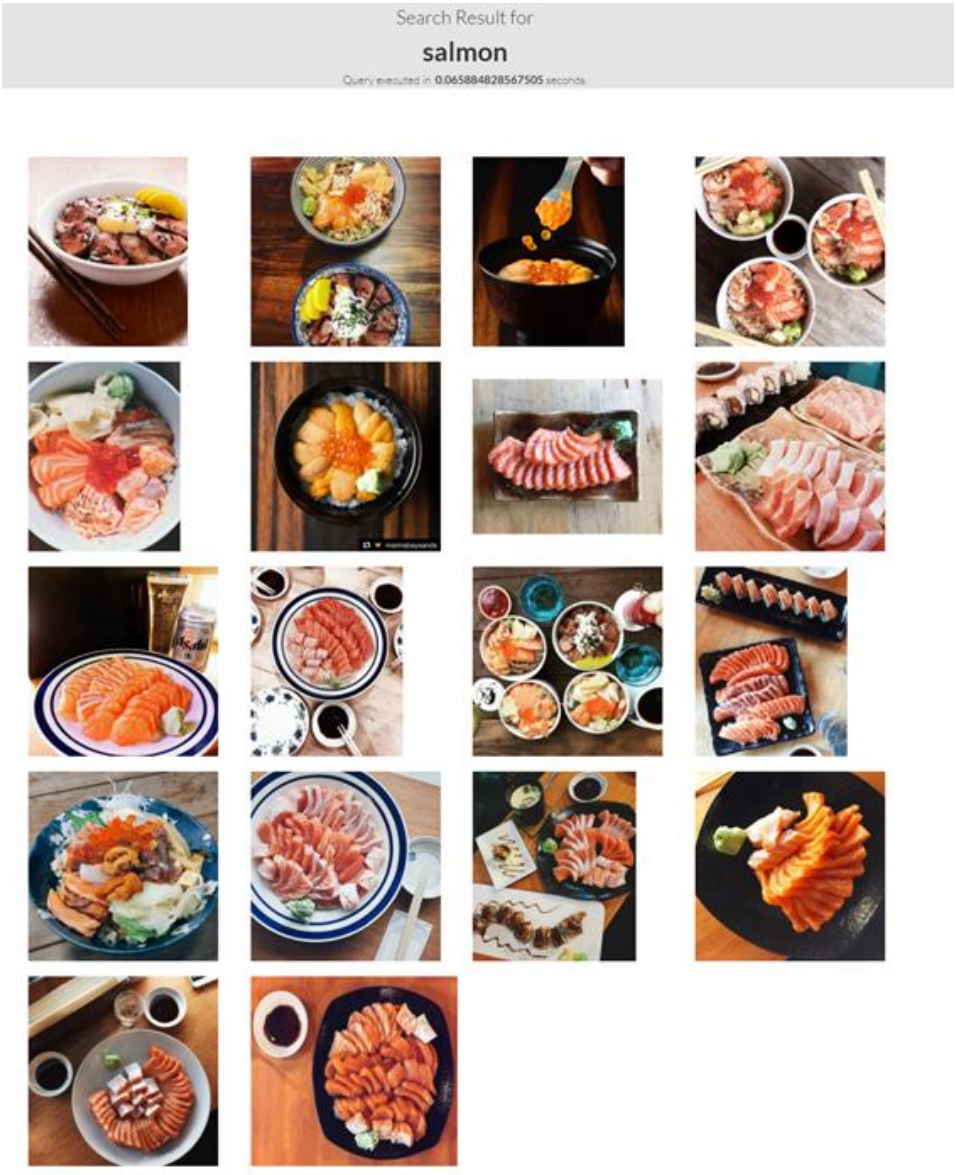

Figure 2.7: Query on "salmon" with more than 200 likes

Results Returned: 18

Query Time: 0.065 seconds

2.3.4 Query 4

Find posts on cake with more than 50 likes.

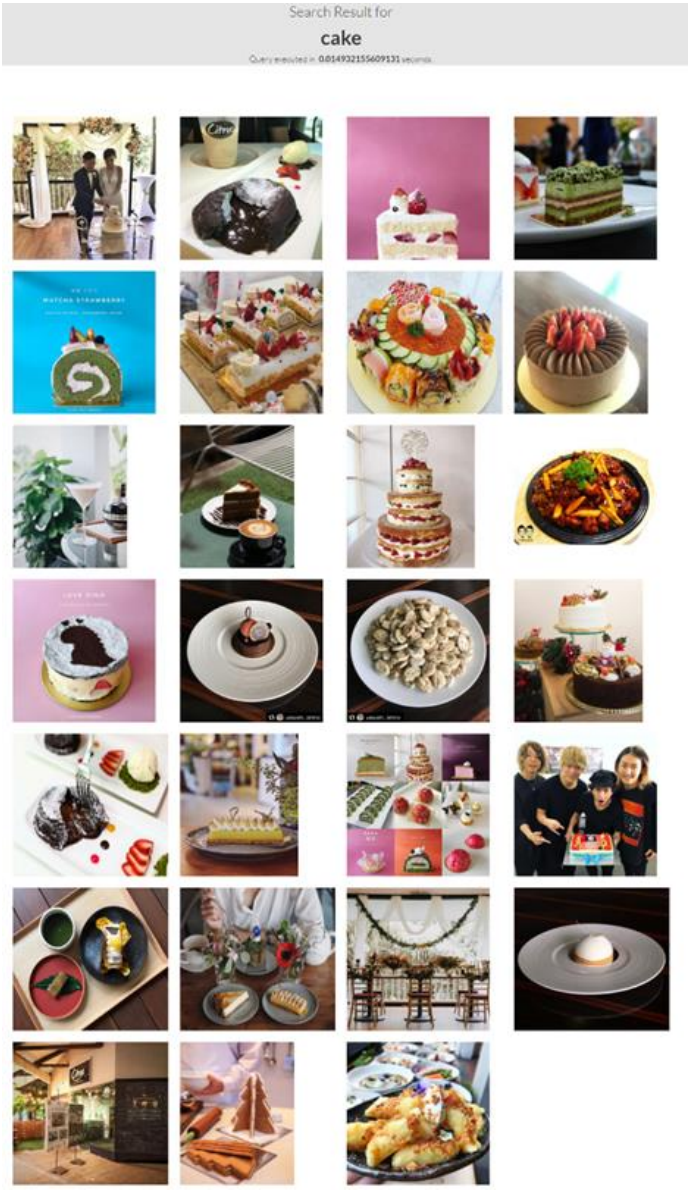

Figure 2.8: Query on "cake" with more than 50 likes

Result Returned: 27

Query Time: 0.014 seconds

2.3.5 Query 5

Find posts on steak with more than 50 likes

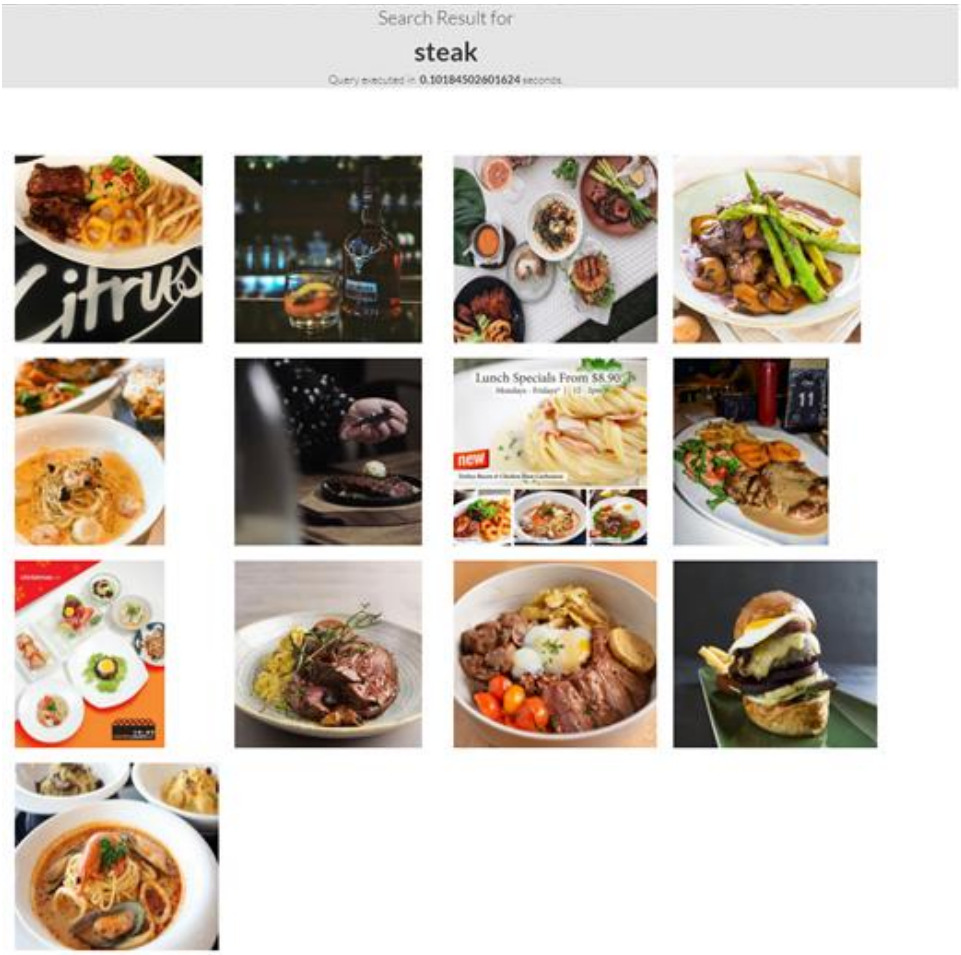

Figure 2.9: Query on "steak" with more than 50 likes

Results Returned: 13

Query Time: 0.101 seconds

The table below summarizes the number of results found and the query time of each example query.

Query	No. of results found	Query time (s)
Bbq Korean. Sort by Likes.	9	0.261
Seafood with Asian filter	2	0.15
Salmon with 200 Likes. Sort by Date.	18	0.065
Cake with >50 Likes	20	0.014
Steak with >50 Likes	13	0.101

3. Innovations for Indexing and Querying

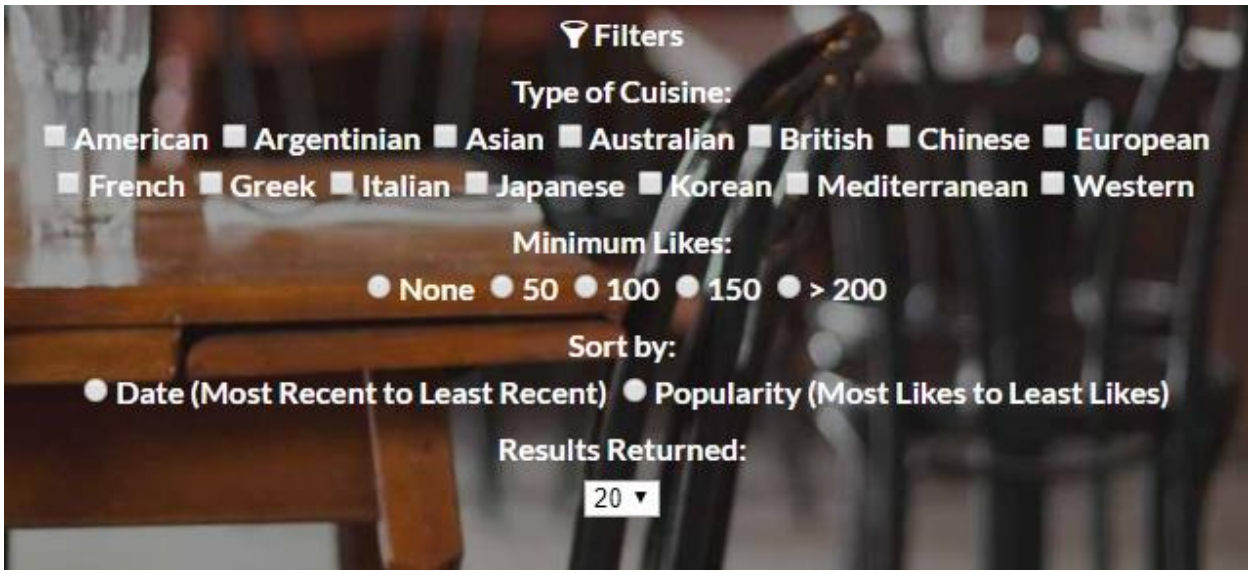

Figure 3.1: Filters to narrow down search

3.1 Sort by Type of Cuisines

This feature retrieve only Instagram posts of a certain type of cuisine. This will be useful for users who have an idea of what kind of cuisine they are looking for. Example: User wants to look at the Korean cuisines containing “fried chicken” so they enter “fried chicken” in the textbox and select the radio button for “Korean” under the “Type of Cuisine:” filter

3.2 Sort by Minimum Likes

This feature retrieve only Instagram posts with a minimum number of likes. This will be useful for users who are looking for posts with certain level of popularity. Example: User wants to look at the posts containing “coffee” with more than 200 likes so they enter “coffee” in the textbox and select the radio button for “>200” under the “Minimum:” filter

3.3 Sort by Latest Posts

This feature will sort the retrieved Instagram post and sort them based on their posted date, from the most recent to the least. This will be useful for users who may want to look at latest posts first. Example: User wants to look at the latest post regarding “Pancake” so they enter “Pancake” in the textbox and select the radio button for “Date” under “Sort by:” filter.

3.4 Sort by Popularity (Number of Likes)

This feature will sort the retrieved Instagram post and sort them by their popularity, based on their number of likes, from the posts with most likes to the post with the least number of likes. This will be useful for users who may want to look at more popular posts first. Example: User wants to look at the latest post regarding “Pancake” so they enter “Pancake” in the textbox and select the radio button for “Popularity” under “Sort by:” filter.

3.5 Limit the Number of Results Returned

This feature will limit the number of Instagram posts to be retrieved. This will be useful for users who may want to look at only a limited number of posts at a time. Example: User wants to look at posts regarding “breakfast” so they enter “breakfast” in the textbox and select “20” in the dropdown box under “Sort by:” filter.

3.6 Check and Modify User’s Query to Improve Accuracy

3.6.1 Remove Stop Words

Our search engine also run the user’s query through a list of stop words and remove them, if found. This allows us to retrieve posts that are more accurate to the user’s query. Moreover, as more specific words are being used to retrieve posts, non-accurate posts will not be retrieved, as compared to before pre-processing was done, leading to a shorter query time.

3.6.2 Stemming

Stemmer was also used for our application to reduce user’s query into their common base forms to help improve on the accuracy of the retrieved data. If stemming is not performed, the same word in different derivative forms will be considered as different words instead of the same word, having an adverse effect on accuracy of the retrieved data, or even affect query time as less accurate post are retrieved as well.

3.6.3 Spelling Check

In addition. We also implemented a spell checker using an API by jaimeburnap (https://github.com/jaimeburnap/spell) as seen in Figure 17. It is a naïve php spell checker based on Peter Norvig’s python implementation.

Super naive php spell checker based on Peter Norvig's python code

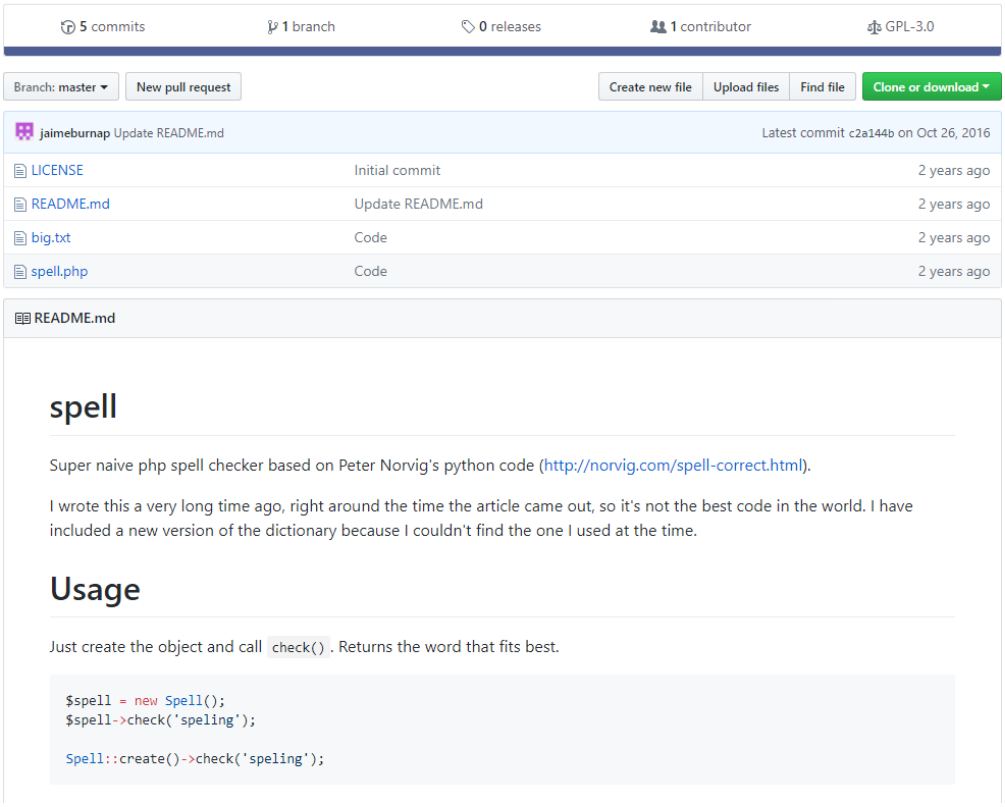

Figure 3.2: Filters to narrow down search using Spelling Check

4. Classification

4.1 Motivate the choice of your classification approach in relation with the state of the art

Our group have decided on using Weka as our classification tool. Weka contains a collection of tools which supports our data mining task for this project such as data pre-processing and classification.

Our approach to this classification task is limited by the attributes we retrieved after crawling. Some of the attributes (photo URL link, Datetime of post, No. of comments and IgAccount) were not suitable for the classification task.

We decide to work with two approaches for our classification task,

1. To classify an Instagram post into a "Review" class (Popular, Unpopular, Neutral) using the "Caption" attribute.
2. To classify an Instagram post into a "Cuisine Category" class (e.g. Asian, Western") using the "Caption" attribute.

We took on a text classification approach for our experiments, the algorithms suitable for our approach are the Naïve Bayes, Support Vector Machine(SVM) and J48 – a Weka Implementation of decision tree.

4.1.1 Naïve Bayes

Naïve Bayes is a classification method that uses supervised learning of document-label assignment function. It is fast learning, simple to implement and is highly scalable. In addition, Naïve Bayes can be used for both binary and multi-class classification problem.

4.1.2 Support Vector Machine (SVM)

SVM is a kind of discriminative classifier formally defined by a separating hyperplane. In our experiment, we will use SVM and compare the results with Naïve Bayes and J48.

4.1.3 J48 Decision Tree

J48 is an implementation of algorithm ID3 developed by WEKA project team. The idea of the decision tree is to find the information gain for a specific attribute, in our case the words in our caption attributes.

4.2 Discuss whether you had to preprocess data and why

4.2.1 Pre-processing Collected Data

The data retrieved from Instagram are raw, and hence pre-processing is required for some of the data collected before Weka can run its classification algorithm.

For our assignment, the collected data were based on the 7 attributes mentioned below:

1. Instagram Post
2. High Resolution Photo
3. Number of Likes
4. Number of Comments
5. Time
6. Caption
7. Cuisine Type

From these attributes, 'Number of Likes' and 'Caption' were chosen to determine the relation between the popularity of an Instagram post and the caption used, using Weka's **text classification analysis**.

First and foremost, to prevent complications and conflicts during the processing of data to be done by Weka, we first had to pre-process the data in the output.csv file by replacing or removing some of the regular expressions. The table below shows the content of the data that had to be pre-processed (in order).

Regular Expression	Explanation
\s\s+	\s matches any whitespace character (equal to [\r\n\t\f\v]) and \s+ matches any whitespace character (equal to [\r\n\t\f\v]) between one and unlimited times, as many times as possible, giving back as needed (greedy). Thereafter, replacing them with a single whitespace
\”	Removing any characters that matches the character “ literally (case sensitive)
\’	Removing any characters that matches the character ‘ literally (case sensitive)
[^$normal_characters]	Removing any characters that matches $normal_characters where $normal_characters = “a-zA-Z0-9\s`~!@#$%^&*()_+-={}|:;<>?,.\/\”\’\\\[\]” (case sensitive)
https:\/\/www.instagram.com	As the existing expression removes all new lines, we replace it with “\nhttps://www.instagram.com” such that new line is added, thus allowing weka to identify new rows.
\.	Removing any characters that matches the character. literally (case sensitive)

4.3 Build an evaluation dataset by manually labelling 10% of the collected data (<u>at least 1,000 records</u>) with an inter-annotator agreement of at least 80%

An evaluation data set comprises of 1114 records was labelled between two members of the group. Cohen’s Kappa formula

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e},$$

was to calculate the inter-annotator agreement between the two rater.

	A	B	C	D	E
1		Popular	Neutral	Unpopular	Row Total
2	Popular	367	21	6	394
3	Neutral	21	291	32	344
4	Unpopular	30	22	324	376
5					
6					
7	Column	418	334	362	Overall = 1114

Agreement between the two rater = P(A) = 328+232+292 = 982

Expected Frequency (1) = 418*394/1114 = 147.83

Expected Frequency (2) = 334*344/1114 = 103.13

Expected Frequency (3) = 362*376/1114 = 122.18

Sum of EF = 147.83+103.13+122.18 = 373.14

Kappa = (982-373.14)/ (1114-373.14) = 608.86/740.86 = 0.8218

An agreement of 82.18% was achieved between the two members on 1114 records.

4.4 Provide evaluation metrices such as precision, recall, and F-measure and discuss results

4.4.1 Experiment 1 – Classify an Instagram Post into a Cuisine Category class

Motivation: To successfully classify/predict whether which category of cuisine it belongs to base on caption.

Training Data needed: Captions(attribute), Category of cuisine(Class)

Possible type of cuisine is as followed:

American
Japanese
Asian
European
French
Italian

Korean
Thai
Western
Mexican

The steps to obtain our result are as followed:

1. Load pre-processed data into Weka explorer
2. Under “Pre-process” Tab, choose “Class Assigner” as the filter and select category as the class (Figure 18)
3. Under “Classify” Tab, choose classifier->meta->filteredClassifier and choose wordToStringVector filter. We then configure and set TFTransform and IDFTransofrm to be true.
4. We also select IteratedLovinsStemmer as our stemmer during classification, rainbow list for stopword and Naïve Bayes as our classifier.

In our classification task, we used 10-fold cross validation to obtain our result

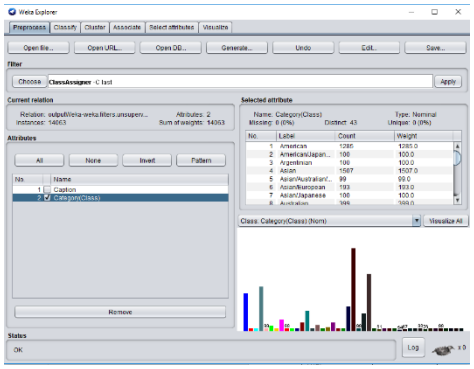

Figure 4.1: Loading the data file and assigning the class

Naïve Bayes Tree Results

	Precision	Recall	F-measure
American	1.0	0.024	0.047
Japanese	0.202	1.0	0.336
Asian	0.895	0.011	0.022
European	0	0	0

French	0	0	0
Italian	0	0	0
Korean	1	0.046	0.088
Thai	0	0	0
Western	0.962	0.013	0.025
Mexican	0	0	0

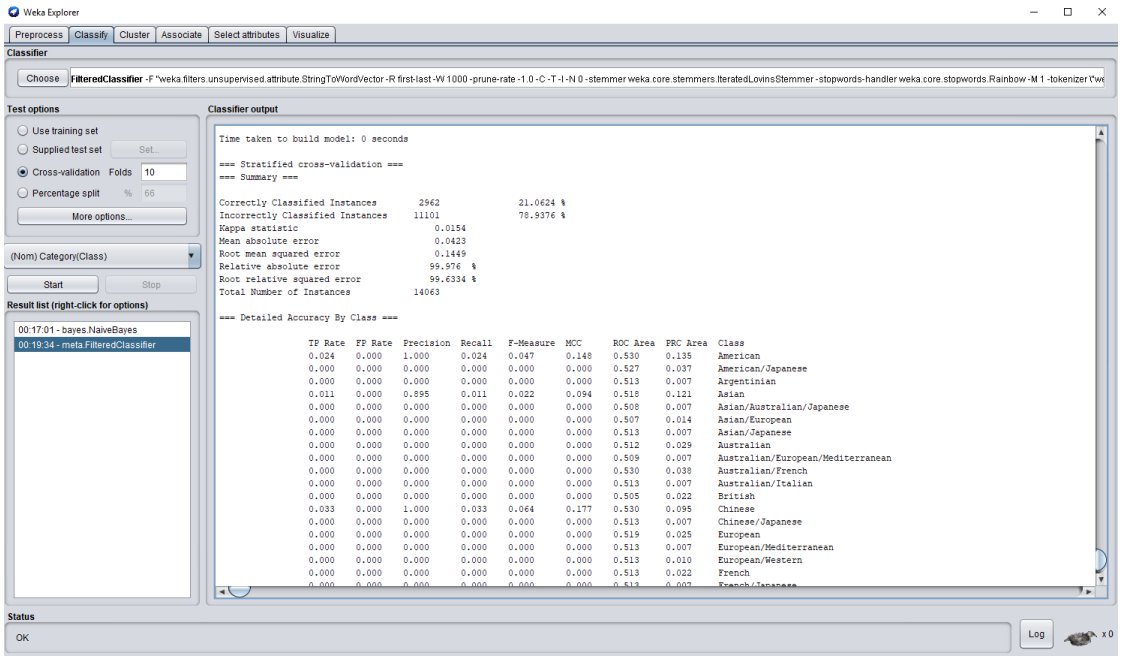

Figure 4.2: Classification results

In our experiment, we did not achieve a satisfactory accuracy for our classification, hence the result is inconclusive. This may be because there is no clear relation between the words of the captions and the class itself, hence the algorithm was not able to build a model to correctly classify the data.

4.4.2 Experiment 2 – Classify an Instagram Post into Review Class (Popular, Unpopular, Neutral)

Motivation: To successfully classify/predict whether the post will be popular, unpopular or neutral based on the caption attribute.

Training Data needed: Captions(attribute), Review(Class)

Possible type of review is as followed:

Unpopular
Popular
Neutral

The steps to obtain our result are similar as Experiment 1.

	Precision	Recall	F-measure
Popular	0	0	0
Neutral	0	0	0
Unpopular	0.501	1	0.668

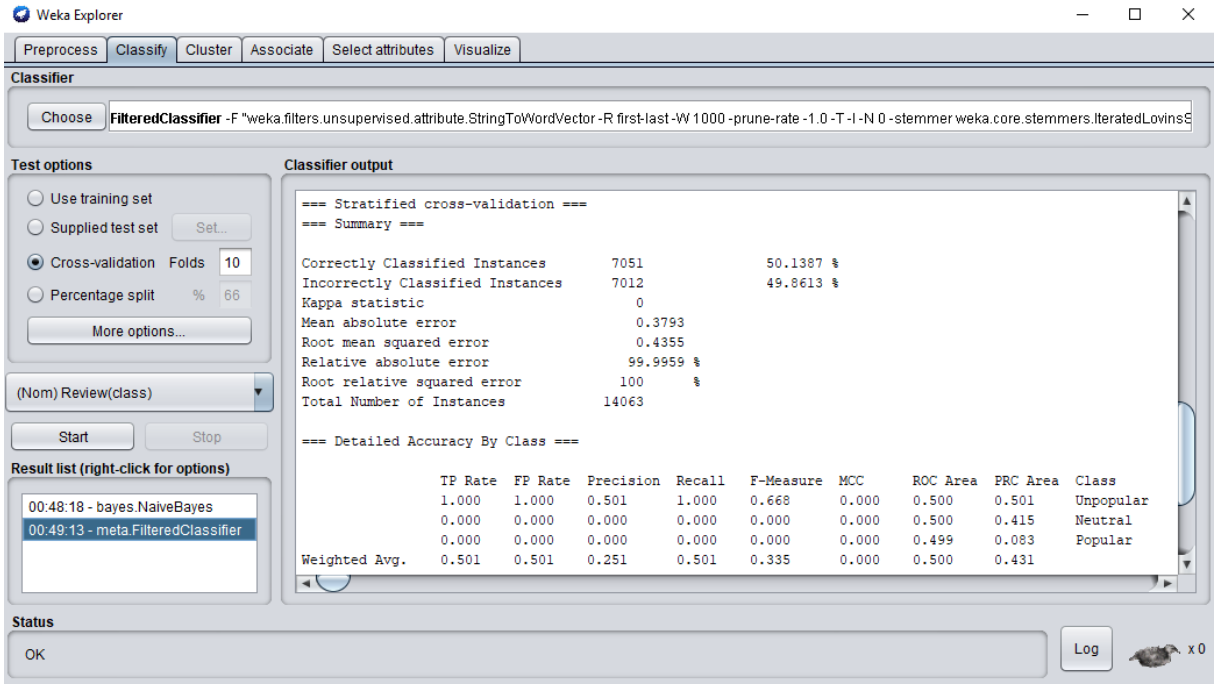

Figure 4.3: Classifying into review class

Similar as Experiment 1, the correct classified instances versus the incorrect classified instance was not satisfactory; 50.1387% against 49.8613%.

From both experiments, we could see that caption was not a useful attribute for either of the classification task.

5. Explore some innovations for enhancing classification. Explain why they are important to solve specific problems, illustrated with examples.

As Instagram was chosen as topic, it may be possible that text classification may not be ideal for performing the above-mentioned experiments. There may be many other deciding factors for example, in deciding how a post is popular.

For example, a nicely taken photo would likely to be more popular (more likes), compared to a poorly taken one. However, there are also exceptions where a poorly taken photo might garnered more likes due to the Instagram account having more followers or better publicize.

One possible way of exploring classification on Instagram could be the use of Image classification, where similar group of photos could determine a specific class of category or a class of review.

If a model can be built to correctly predict whether an image taken, or post would attract more likes, it could also aid restaurants in improving their publicity through Instagram.

Presentation Video URL

https://youtu.be/nGgwehErODo

Source codes Dropbox URL

https://www.dropbox.com/sh/wrggub37h4biaw2/AADUU_mdZ0TrcmiwSB2PZg0Ea?dl=0

Data Dropbox URL (for Q3 and Q5)

Weka Data under “Classification” Folder from the Dropbox folder in the above link

Interactive Search Q3 – under “Search Engine” Folder. Implemented in Index.php