CZ4034: Information Retrieval

Review based Amazon Information Retrieval System

Group 26

Prepared by
Lim Jing Qiang (U1722144E)
Arkar Min (U1721052K)
Lau Xin Ru (U1722233B)
Oon Zi Hui (U1722934E)

Tutorial Group CS4

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

Content Page

1. Introduction

The topic for our assignment will be Review based Amazon Information Retrieval System.

When shopping for a product, the first thing a customer would do is to look up for product information such as ratings and reviews online. We chose to crawl data from Amazon because Amazon is one of the top online marketplaces and has adopted the star-based rating system where the products are rated by customers from a scale of 5 stars being the highest to 1 star being the lowest. The overall star-based rating scores is a good indicator for the customers to contemplate whether to purchase the product or not.

However, the star-based rating score is not always accurate. Research has shown that the text reviews from customers must be taken into account in order to get a more accurate rating [1]. Therefore in order to address this problem, our team have decided to develop an information retrieval system that could rank the products based on customer's reviews.

In this assignment, we first crawl the necessary data on Amazon using the crawler we built. Before storing the crawled data into Solr, we will perform preprocessing on the crawled data such as product classification, review classification and review summarization.

Afterward, we will store the preprocessed data into Solr and perform the indexing of the data extracted. Finally, we built the UI before adding additional features that could make the application easy for users to use.

2. Crawling

2.1 How you crawled the corpus and stored them

Due to some limitations of Amazon API, our team has decided to develop a robust crawler+scraper instead of using the API. For example, if the number of requests submitted exceeds the maximum request limit for our account, we might receive an error from the Amazon Product Advertising API. In addition, the Amazon Web Service (AWS) free tier will charge a fee when the AWS Free Tier limit is exceeded. Hence, the API is unsuitable for this assignment as a large amount of data needs to be crawled and it consists of many trial and errors.

The crawling+scraping application can be broken down into two parts:

a) Amazon Standard Identification Number(ASIN) Crawler
b) Product Review Scraper based on ASIN (Corpus)

Amazon Standard Identification Number(ASIN) Crawler

Every product on Amazon is identified by a 10-character alphanumeric unique identifier known as ASIN and is assigned by Amazon. For the ASIN Crawler to work, it will first make an HTTP GET request to Amazon homepage (https://www.amazon.com). The response received from the Amazon server will include hyperlinks and region of interest displayed in the hoverable drop-down navigation menu as shown in Figure 1.1.

The drop-down menu consists of hyperlinks to all the categories that are available on Amazon, and the crawler will always crawl for hyperlinks in the drop-down menu. Therefore, whenever a new category is added, there is no need to modify the application to include the new category. Thus, contributing to the robustness of the crawler.

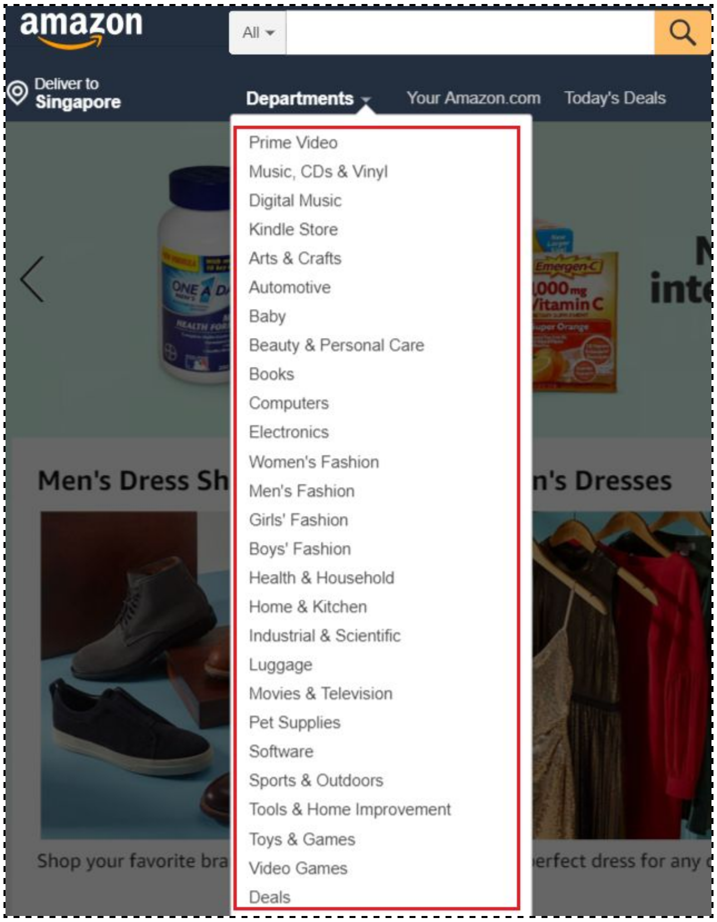
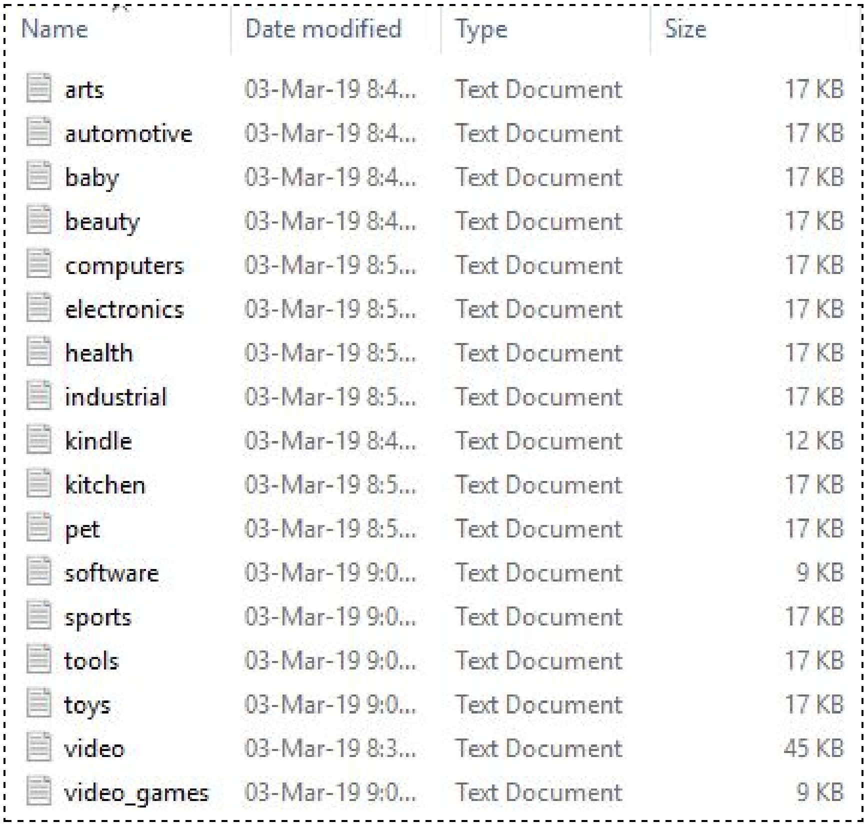


Figure 1.1. Drop-down navigation menu for various departments bounded with a red rectangular box

The hyperlinks will then be stored in an ArrayList called URL Frontier. After the crawling of the various categories is completed, the crawler will proceed to make HTTP requests to crawl for the ASIN of all products in each category. The ASINs will be saved in specific text files. For example, the ASINs belonging to products under the "Computers" category would be saved into "computer.txt" and the ASINs that belongs to the products under "Software" category would be saved into "software.txt".

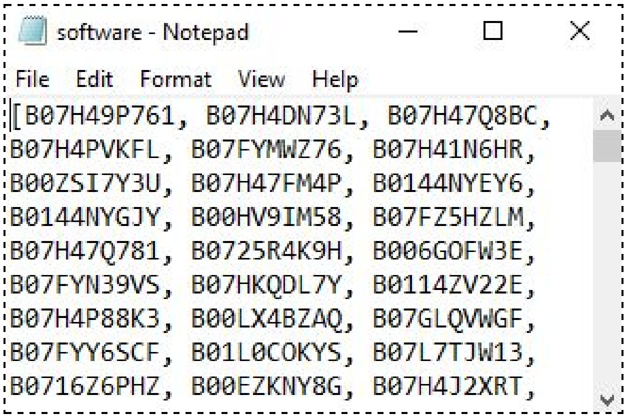
For this assignment, the crawler will only crawl for products on the first 20 pages of each category before proceeding to the next category. The files created by the crawler and an example of the contents in the created file are as shown in Figure 1.2 and Figure 1.3 respectively.

*** Note: The files created by the ASIN crawler is not part of the corpus.*



Name	Date modified	Type	Size
arts	03-Mar-19 8:4...	Text Document	17 KB
automotive	03-Mar-19 8:4...	Text Document	17 KB
baby	03-Mar-19 8:4...	Text Document	17 KB
beauty	03-Mar-19 8:4...	Text Document	17 KB
computers	03-Mar-19 8:5...	Text Document	17 KB
electronics	03-Mar-19 8:5...	Text Document	17 KB
health	03-Mar-19 8:5...	Text Document	17 KB
industrial	03-Mar-19 8:5...	Text Document	17 KB
kindle	03-Mar-19 8:4...	Text Document	12 KB
kitchen	03-Mar-19 8:5...	Text Document	17 KB
pet	03-Mar-19 8:5...	Text Document	17 KB
software	03-Mar-19 9:0...	Text Document	9 KB
sports	03-Mar-19 9:0...	Text Document	17 KB
tools	03-Mar-19 9:0...	Text Document	17 KB
toys	03-Mar-19 9:0...	Text Document	17 KB
video	03-Mar-19 8:3...	Text Document	45 KB
video_games	03-Mar-19 9:0...	Text Document	9 KB

Figure 1.2. Files created by the ASIN crawler.



```
[B07H49P761, B07H4DN73L, B07H47Q8BC,
B07H4PVKFL, B07FYMWZ76, B07H41N6HR,
B00ZSI7Y3U, B07H47FM4P, B0144NYEY6,
B0144NYGJY, B00HV9IM58, B07FZ5HZLM,
B07H47Q781, B0725R4K9H, B006GOFW3E,
B07FYN39VS, B07HKQDL7Y, B0114ZV22E,
B07H4P88K3, B00LX4BZAQ, B07GLQVWGF,
B07FYY6SCF, B01L0COKYS, B07L7TJW13,
B0716Z6PHZ, B00EZKNY8G, B07H4J2XRT,
```

Figure 1.3. Contents in software.txt

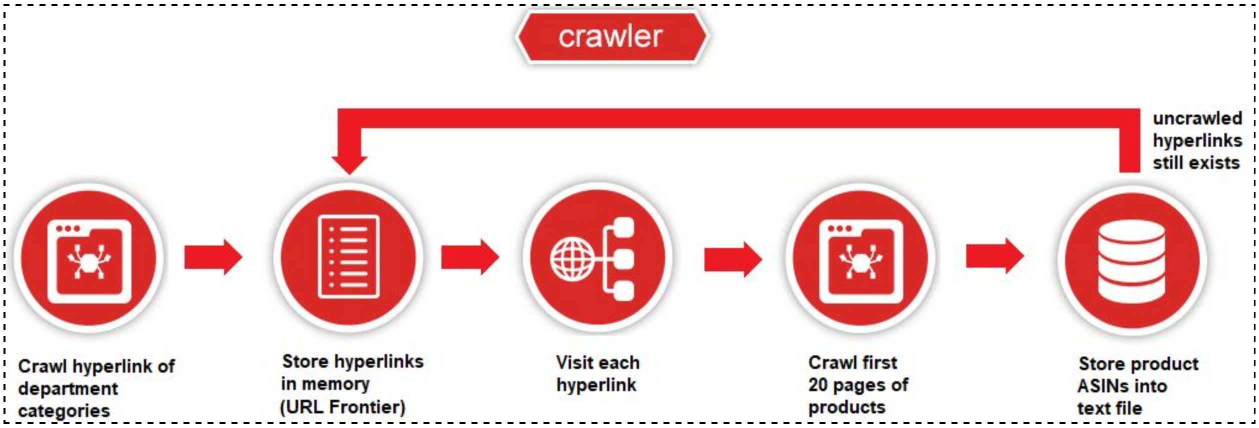
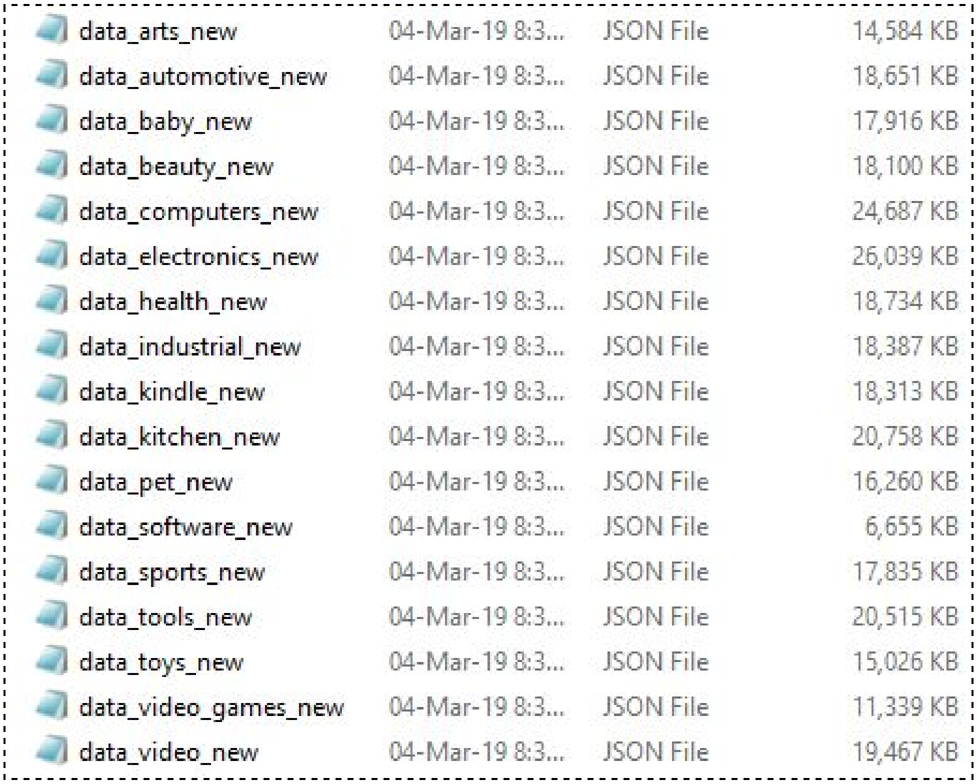

crawler
uncrawled hyperlinks still exists
Crawl hyperlink of department categories
Store hyperlinks in memory (URL Frontier)
Visit each hyperlink
Crawl first 20 pages of products
Store product ASINs into text file

Figure 1.4. Visual representation of the crawler

<u>Product Review Scraper based on ASIN</u>

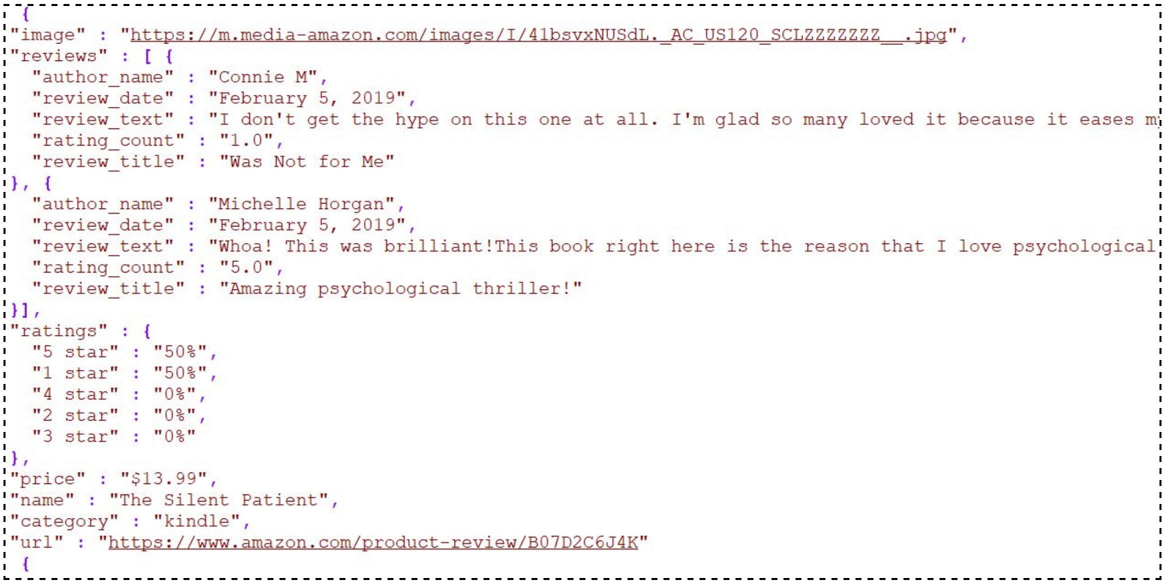
Once the files containing the ASINs are created, the Product Review Scraper will attempt to read from an user-specified text file and extract the first ASIN. The scraper will then make an HTTP request to the product review page (https://www.amazon.com/product-review/**ASIN**) to scrap for product reviews. For this assignment, only the product reviews on the first two pages would be scraped. Upon completion, that ASIN is removed from the text file and the scraper will proceed to extract the next ASIN to scrap for reviews. The strategy of extracting and removing the ASIN from the text file upon completion also contributes to its robustness. In the scenario where the user's PC shuts down unexpectedly, the scraper will know which ASIN to continue from when PC is rebooted.

The scraped data will be stored in a JSON file according to its category (Figure 1.5). A screen capture of the contents in the JSON file is shown in Figure 1.6. The description of each key names of the JSON will be explained in Table 1.



data_arts_new	04-Mar-19 8:3...	JSON File	14,584 KB
data_automotive_new	04-Mar-19 8:3...	JSON File	18,651 KB
data_baby_new	04-Mar-19 8:3...	JSON File	17,916 KB
data_beauty_new	04-Mar-19 8:3...	JSON File	18,100 KB
data_computers_new	04-Mar-19 8:3...	JSON File	24,687 KB
data_electronics_new	04-Mar-19 8:3...	JSON File	26,039 KB
data_health_new	04-Mar-19 8:3...	JSON File	18,734 KB
data_industrial_new	04-Mar-19 8:3...	JSON File	18,387 KB
data_kindle_new	04-Mar-19 8:3...	JSON File	18,313 KB
data_kitchen_new	04-Mar-19 8:3...	JSON File	20,758 KB
data_pet_new	04-Mar-19 8:3...	JSON File	16,260 KB
data_software_new	04-Mar-19 8:3...	JSON File	6,655 KB
data_sports_new	04-Mar-19 8:3...	JSON File	17,835 KB
data_tools_new	04-Mar-19 8:3...	JSON File	20,515 KB
data_toys_new	04-Mar-19 8:3...	JSON File	15,026 KB
data_video_games_new	04-Mar-19 8:3...	JSON File	11,339 KB
data_video_new	04-Mar-19 8:3...	JSON File	19,467 KB

Figure 1.5. JSON files containing the product reviews for each of the category.



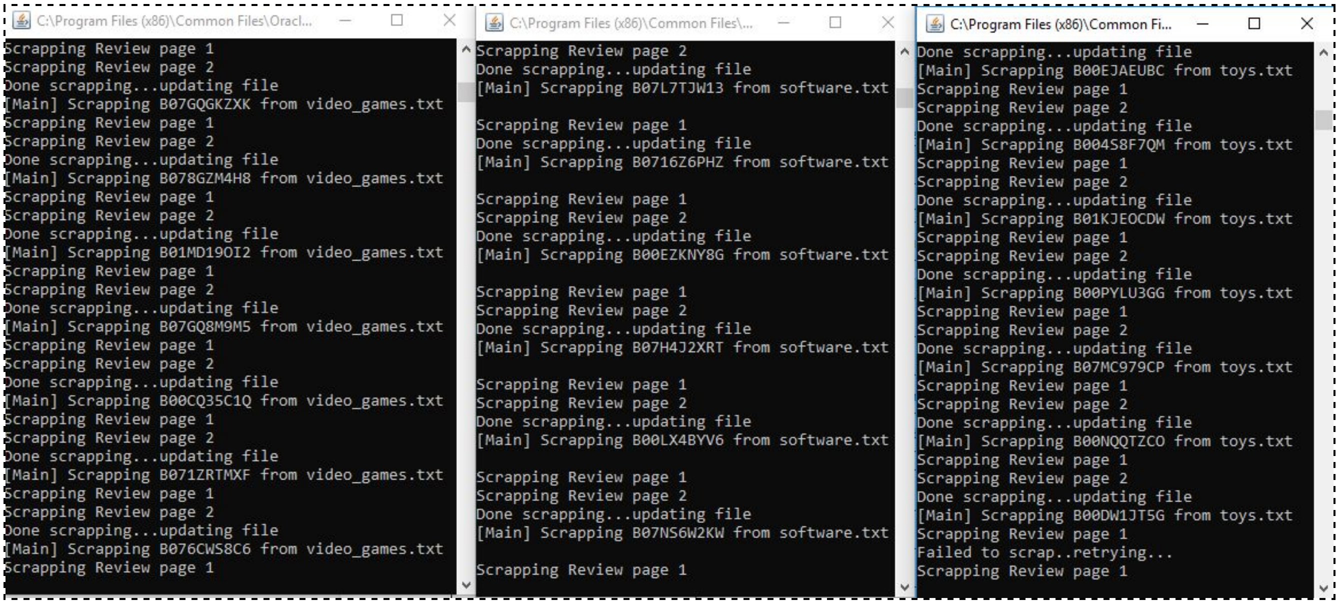
```
 {
"image" : "https://m.media-amazon.com/images/I/41bsvxNUSdL._AC_US120_SCLZZZZZZZ__.jpg",
"reviews" : [ {
  "author_name" : "Connie M",
  "review_date" : "February 5, 2019",
  "review_text" : "I don't get the hype on this one at all. I'm glad so many loved it because it eases m
  "rating_count" : "1.0",
  "review_title" : "Was Not for Me"
}, {
  "author_name" : "Michelle Horgan",
  "review_date" : "February 5, 2019",
  "review_text" : "Whoa! This was brilliant!This book right here is the reason that I love psychological
  "rating_count" : "5.0",
  "review_title" : "Amazing psychological thriller!"
}],
"ratings" : {
  "5 star" : "50%",
  "1 star" : "50%",
  "4 star" : "0%",
  "2 star" : "0%",
  "3 star" : "0%"
},
"price" : "$13.99",
"name" : "The Silent Patient",
"category" : "kindle",
"url" : "https://www.amazon.com/product-review/B07D2C6J4K"
 {
```

Figure 1.6. Screen capture of a scraped product review.

Table 1. Description of key names

Key name	**Description**
image	A hyperlink to the product’s thumbnail.
reviews	A JSONArray containing all the reviews on the specific product.
author_name	The name of the author who wrote the review.
review_date	The date of the review when it was written.
review_text	The textual review provided by the customer.
rating_count	The number of stars the customer rated on the specific product.
review_title	The title of the specific review by individual customers.
ratings	Number(in terms of percentage) of ratings from 5 stars to 1 star.
price	Price of the product.
name	Name of the product.
category	Category the product belongs to.
url	The hyperlink to the product review.

As the files containing ASIN are created separately based on its category, scraping for each category can be executed concurrently by parsing in the individual ASIN files. Thus, making the scraper scalable and efficient. A screen capture of the concurrent crawling is shown in Figure 1.7.


C:\Program Files (x86)\Common Files\Oracl...
C:\Program Files (x86)\Common Files\...
C:\Program Files (x86)\Common Fi...

```
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B07GQGKZXK from video_games.txt
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B078GZM4H8 from video_games.txt
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B01MD19OI2 from video_games.txt
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B07GQ8M9M5 from video_games.txt
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B00CQ35C1Q from video_games.txt
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B071ZRTMXF from video_games.txt
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B076CWS8C6 from video_games.txt
Scrapping Review page 1
```

```
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B07L7TJW13 from software.txt

Scrapping Review page 1
Done scrapping...updating file
[Main] Scrapping B0716Z6PHZ from software.txt

Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B00EZKNY8G from software.txt

Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B07H4J2XRT from software.txt

Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B00LX4BYV6 from software.txt

Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B07NS6W2KW from software.txt

Scrapping Review page 1
```

```
Done scrapping...updating file
[Main] Scrapping B00EJAEUBC from toys.txt
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B004S8F7QM from toys.txt
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B01KJEOCDW from toys.txt
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B00PYLU3GG from toys.txt
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B07MC979CP from toys.txt
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B00NQQTZCO from toys.txt
Scrapping Review page 1
Scrapping Review page 2
Done scrapping...updating file
[Main] Scrapping B00DW1JT5G from toys.txt
Scrapping Review page 1
Failed to scrap..retrying...
Scrapping Review page 1
```

Figure 1.7. Concurrent scraping of video games, software and toys category

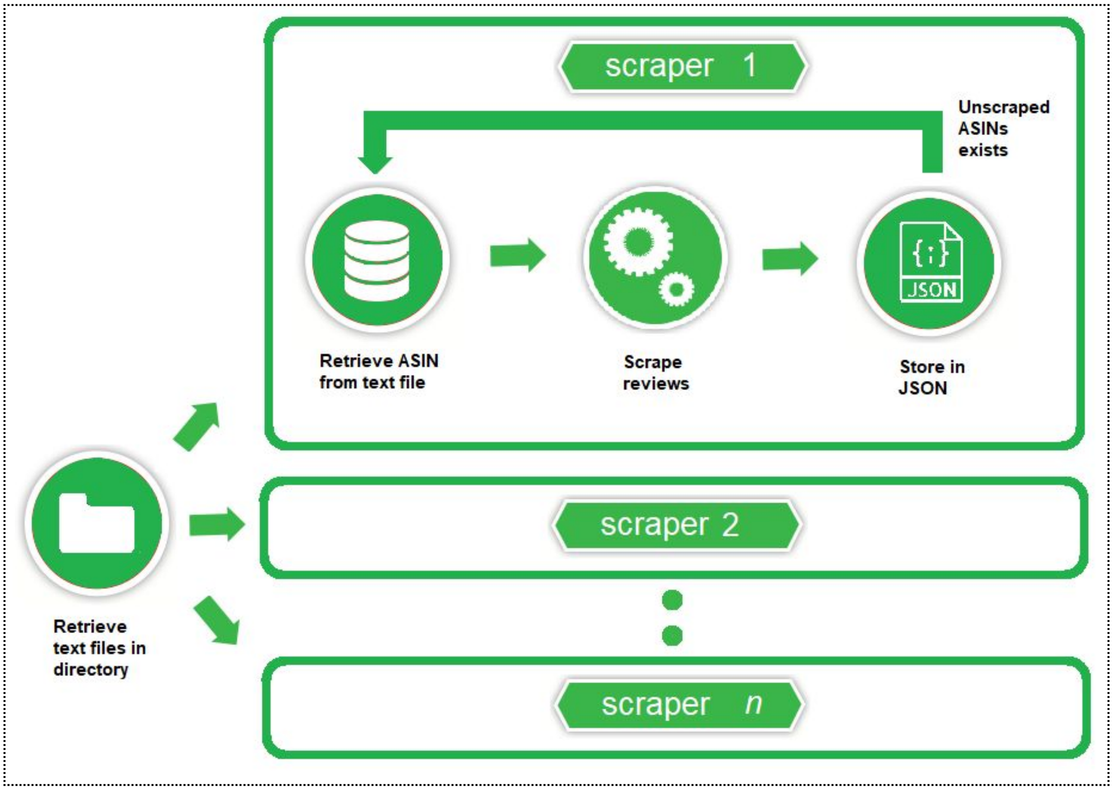


Figure 1.8. Visual representation of concurrent scraper

2.2 What kinds of information users might like to retrieve

The user could retrieve the reviews of a particular product through our system. For example, the user can search for “iPhone X” and product reviews related to “iPhone X” would be displayed to the user.

Example queries are:
1. iPhone X
2. Nokia
3. Best Phone

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

There are a total of 169,178 product reviews extracted from 10802 products from Amazon. A breakdown of the corpus statistics will be shown in Table 2. The number of unique words in the corpus is 318964. These statistical data would remain the same as we will be using static data in this project.

Table 2. Breakdown of the corpus statistics

Category	Product count	Number of reviews	Number of unique words in Product Name, Review Text & Title
Beauty	1434	22927	44801
Computers	1437	22692	44741
Cellphone	1322	22984	42335
Kindle	959	15249	29877
Software	674	8763	16631
Sports	1425	22167	45675
Toys	1400	21794	43626
Video Games	671	10495	31699
Video	1480	22107	19579
Total	10802	169178	318964

2.4 Review Summarization

There are a lot of reviews for each product. It is inconvenient for users to read through all the reviews although they play as an important factor in deciding whether the product is good or not. So, text summarization will be done on the reviews for each product so that users can judge the quality of the product in a very short time.

There are two types of text summarization, abstractive summarization, and extractive summarization. Abstractive summarization chooses words based on semantic understanding even they are not included in the source documents. It aims at producing the summary in a new way which means it will try to understand the meanings of the sentences and write a new summary with its words. Extractive summarization tries to select a subset of the word which retains the most important points. It weighs the important part of sentences and uses the same sentence to form the summary, which means it will not create new sentences, however, it will use the existing sentences to write the summary.

Extractive summarization will be used for this assignment since a lot of advanced NLP techniques are needed to be used for abstractive summarization and extractive summarization uses techniques like cosine similarity, and stop words removal.

So, firstly, sentences from review text from JSON files are retrieved. Stopwords from NLTK (Natural Language Toolkit) are used to detect and clean the stopwords from review sentences. Then, they are converted to lowercase and converted into word vectors. Then, the cosine similarity matrix is constructed by computing cosine distance between one sentence and the remaining sentences. Then, it is passed to NetworkX, a package which ranks the sentences based on the score. After that, the top five sentences are retrieved and passed to the new summary field in each product inside the JSON file.

The source code is in Summarize.py.

3. Indexing and Querying

3.1 Indexing

Solr + Lucene + Jetty is used to index the crawled data. Inside crawled data, each product is a JSON object containing the following fields.

Table 3. Description of each field in the JSON object

Key name	**Description**
image	A hyperlink to the product's thumbnail.
reviews	A JSONArray containing all the reviews on the specific product.
author_name	The name of the author who wrote the review.
review_date	The date of the review when it was written.
review_text	The textual review provided by the customer.
rating_count	The number of stars the customer rated on the specific product.
review_title	The title of the specific review by individual customers.
ratings	Number(in terms of percentage) of ratings from 5 stars to 1 star.
price	Price of the product.
name	Name of the product.
category	Category the product belongs to.
url	The hyperlink to the product review.
summary	The summary of product reviews

Among all the fields, only name, summary, category, price, and ratings are indexed since they bring the most important information about the product. When indexing price, category, and ratings, TF-IDF, normalization and position vectors are omitted since they are either number or one-word category. The rests are stored but not indexed.

If we define only field schema, we need to specify the field that we want to search when we do querying. So, we need to add an additional field which is called "copy field" in Solr which will search the query in all fields without needing to specify.

3.2 Query Expansion

Query Expansion is needed to be done since query like “best antivirus” is targeted. The word “best” may or may not be included in good reviews. Therefore, own thesaurus is defined in Solr for those words. For example, when users are for the query “best antivirus”, the search engine also need to look for the term like “glad, nice, good, happy” terms to give the best result.

3.3 Build a simple web interface for the search engine

A web interface was built using HTML, Javascript, Jquery, and AJAX to search and retrieve product reviews based on the data we crawled from Amazon. Below is the design of the web interface.

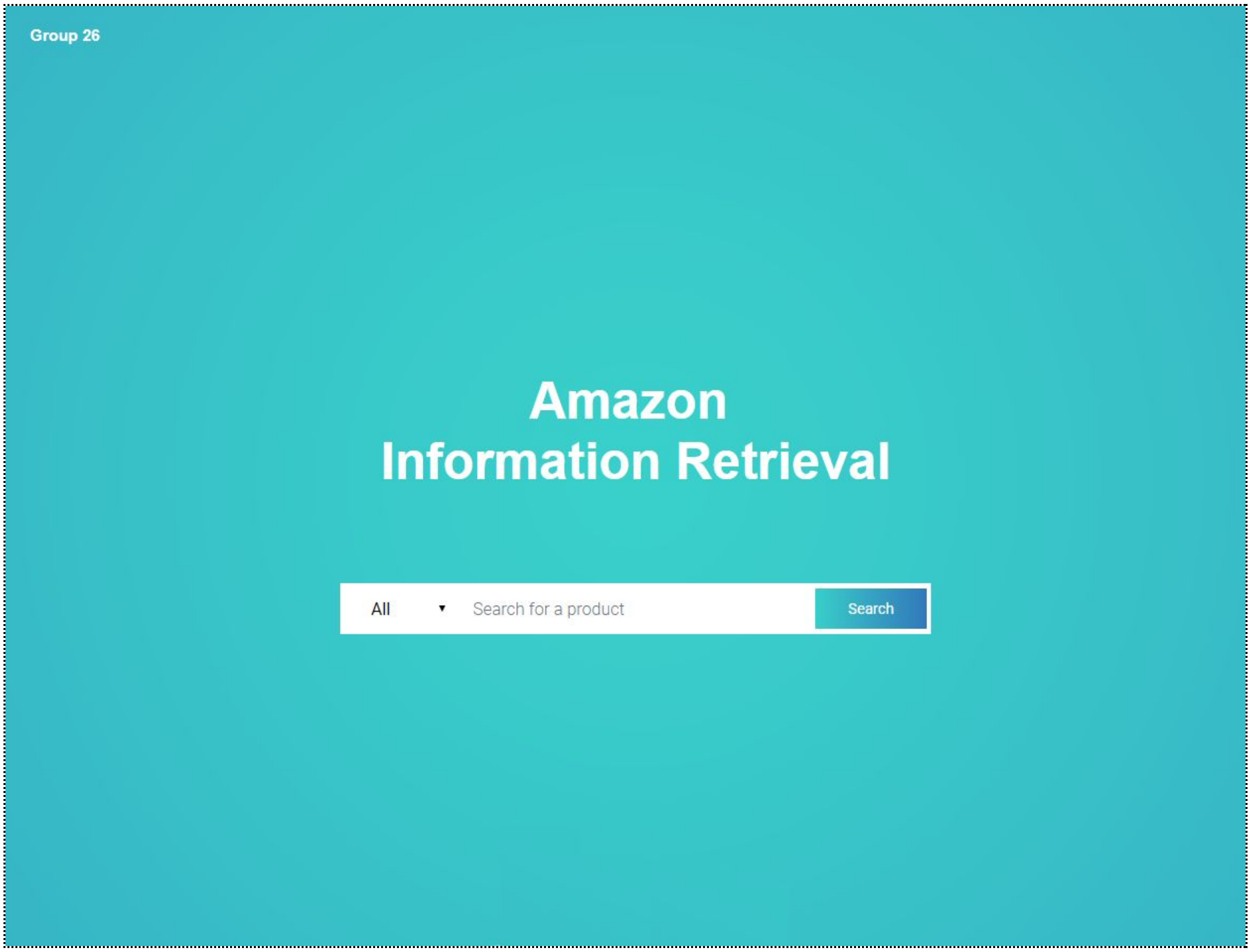
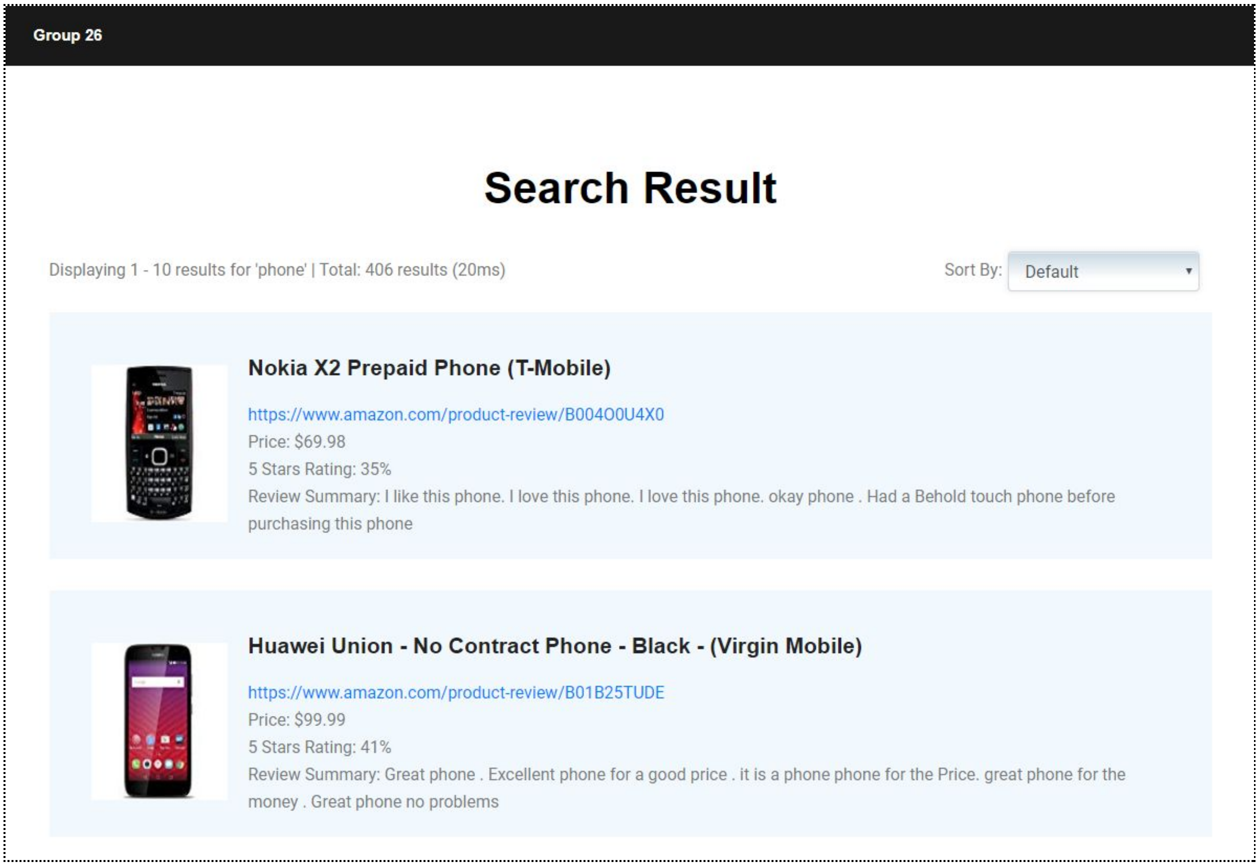


Figure 2.1. Web Interface of Amazon Information Retrieval


Group 26

Search Result

Displaying 1 - 10 results for 'phone' | Total: 406 results (20ms)

Sort By: Default

Nokia X2 Prepaid Phone (T-Mobile)

https://www.amazon.com/product-review/B004O0U4X0

Price: $69.98

5 Stars Rating: 35%

Review Summary: I like this phone. I love this phone. I love this phone. okay phone . Had a Behold touch phone before purchasing this phone

Huawei Union - No Contract Phone - Black - (Virgin Mobile)

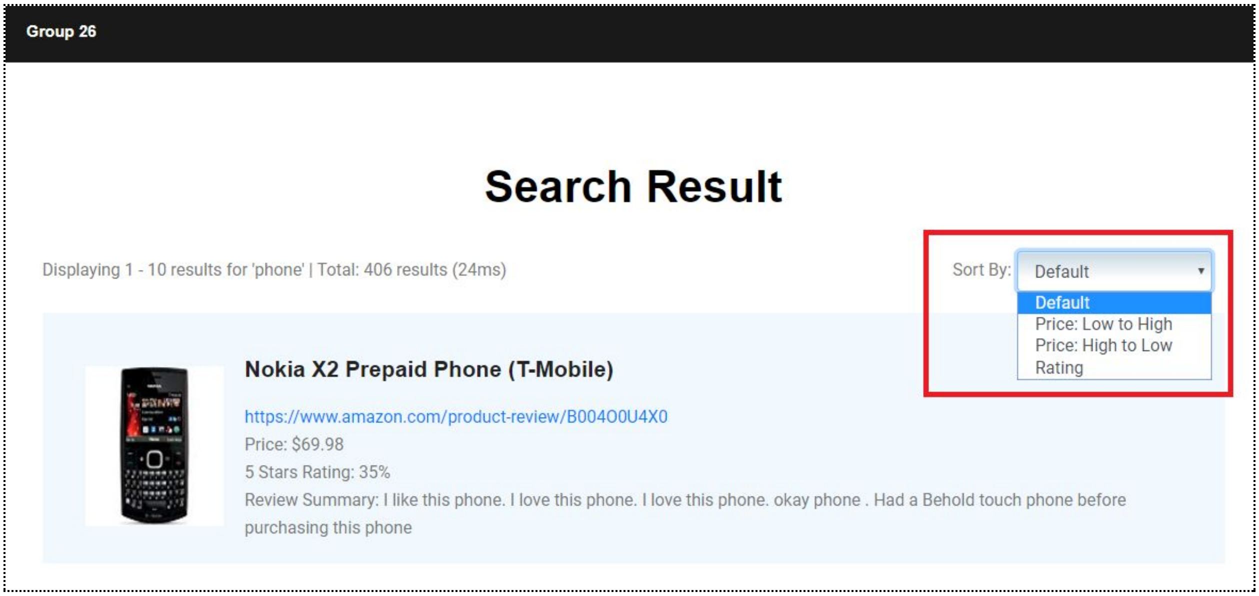
https://www.amazon.com/product-review/B01B25TUDE

Price: $99.99

5 Stars Rating: 41%

Review Summary: Great phone . Excellent phone for a good price . it is a phone phone for the Price. great phone for the money . Great phone no problems

Figure 2.2. Display of Search Results on Web Interface


Group 26

Search Result

Displaying 1 - 10 results for 'phone' | Total: 406 results (24ms)

Sort By: Default

Nokia X2 Prepaid Phone (T-Mobile)

https://www.amazon.com/product-review/B004O0U4X0

Price: $69.98

5 Stars Rating: 35%

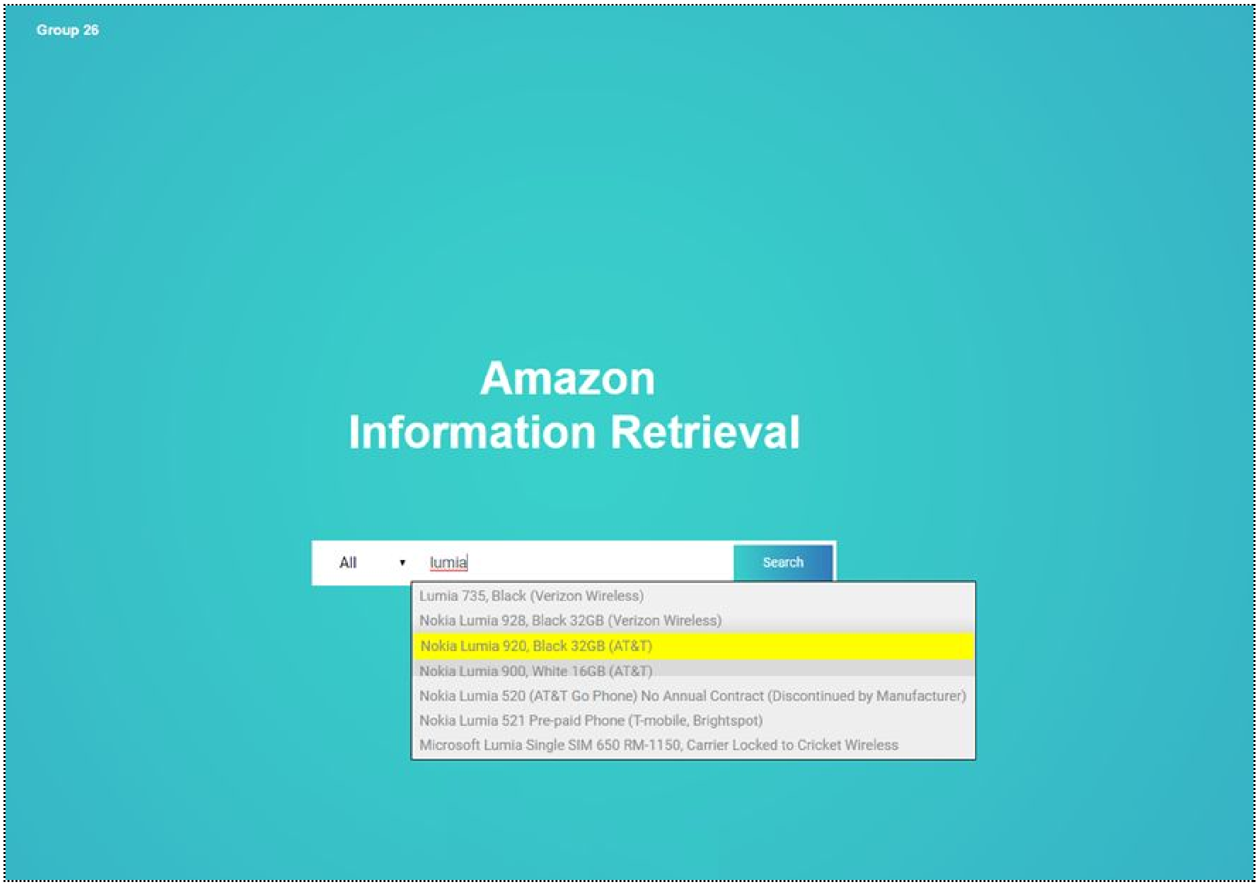
Review Summary: I like this phone. I love this phone. I love this phone. okay phone . Had a Behold touch phone before purchasing this phone

Figure 2.3. Sorting of Search Results

Our web interface also provides user to sort the list of search results by Default, Price: Low to High, Price: High to Low and ratings.

- **Sort by Rating**
  Amazon uses a star rating system for consumers to rate a product from a scale of 1 to 5, with 5 being the highest rating. Our web interface allows the sorting of the product reviews based on the product with the highest 5 stars ratings crawled from Amazon.

- **Sort by Price**
  Price is one of the factors to look out for when shopping online. To provide convenience, our web interface allows the sorting of product reviews either from lowest to highest price or highest to lowest price.


Group 26
Amazon
Information Retrieval
All ▾
Search
Lumia 735, Black (Verizon Wireless)
Nokia Lumia 928, Black 32GB (Verizon Wireless)
Nokia Lumia 920, Black 32GB (AT&T)
Nokia Lumia 900, White 16GB (AT&T)
Nokia Lumia 520 (AT&T Go Phone) No Annual Contract (Discontinued by Manufacturer)
Nokia Lumia 521 Pre-paid Phone (T-mobile, Brightspot)
Microsoft Lumia Single SIM 650 RM-1150, Carrier Locked to Cricket Wireless

lumia

Figure 2.4. Suggestions based on Search Value

Our web interface provides suggestions to users based on what they entered into the search field. This auto-suggestion feature will retrieve the suggested values based on the name of the products. For example, when "nokia" is entered into the search field, a list of products containing the search term "nokia" will be displayed. We choose to retrieve suggested values based on product names so that the user will not enter words that are too vague, which might affect the search results.

3.4 Write five queries, get their results and measure the speed of the querying

Table 4. Results and time taken for example queries

Query	No of Results Found	Top Result	Time Taken
phone	1419	LG G6 - 32 GB - Unlocked (AT&T/T-Mobile/Verizon) - Black	23 ms
antivirus	10	ESET NOD32 Antivirus - 1 Device, 3 Years | CD - ROM	14 ms
1984	3	1984	17 ms
Best phone	5054	LG V20 64GB H918 - Unlocked by T-Mobile for all GSM Carriers (Titan Gray)	11 ms
Pokemon Game	872	Pokemon: Let"s Go, Pikachu!	12 ms

3.5 Explore some innovations for enhancing the indexing and ranking. Explain why they are important to solve specific problems, illustrated with examples.

3.5.1 Clustering

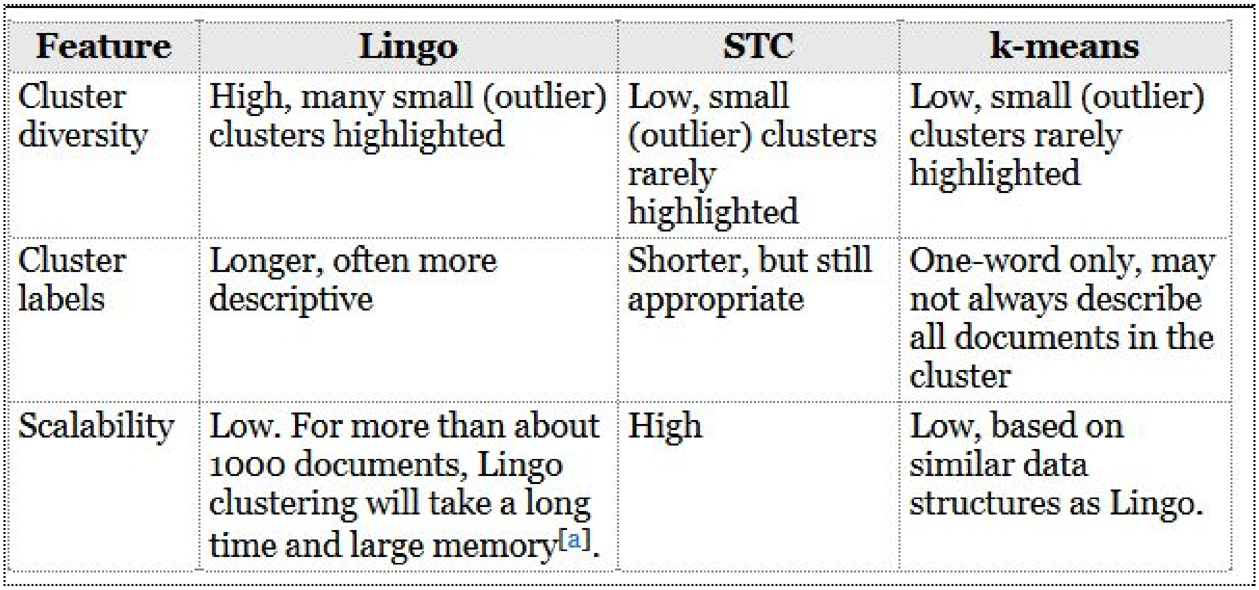
Clustering is the process of grouping related search hits and assigns human-readable labels automatically and all these are not pre-defined. The clustering algorithm is applied to the search result by default in Solr. This would allow the user to find related products easily.

Solr comes with several algorithms implemented in the open source Carrot2 project. In Carrot2, it offers two specialized search results clustering algorithms: Lingo and Suffix Tree Clustering (STC), as well as an implementation of the bisecting k-means clustering.

The key characteristics of the Lingo algorithm first identify labels, then it assigns documents to that label to form a cluster. By building a term-document matrix for all input documents, followed by decomposing the matrix to obtain a number of base vectors that approximate the matrix in a low-dimensional space. Each document containing the label's words are assigned to that label.

The STC algorithm is a Generalized Suffix Tree (GST). The algorithm traverses the tree and identifies words and phrases that occur more than once in that document. Each word or phrase becomes a base cluster. Finally, it merges the base clusters to form the final cluster.

Both algorithms allow overlapping clusters, meaning one document can be assigned to multiple clusters. The bisecting k-means algorithm does not allow for overlapping clusters, and its labels consist of a single word, which may not be descriptive enough to the user.



Feature	Lingo	STC	k-means
Cluster diversity	High, many small (outlier) clusters highlighted	Low, small (outlier) clusters rarely highlighted	Low, small (outlier) clusters rarely highlighted
Cluster labels	Longer, often more descriptive	Shorter, but still appropriate	One-word only, may not always describe all documents in the cluster
Scalability	Low. For more than about 1000 documents, Lingo clustering will take a long time and large memory[a].	High	Low, based on similar data structures as Lingo.

Figure 3.1. Characteristics of Lingo and STC clustering algorithms

STC would be more appropriate in our system since there is a large number of documents to be processed and the cluster labels do not need to be very lengthy. The overlapping clusters also help the user locate a product more easily.

4. Classification

There will be two types of classification. The first type of classification of categories of products and the second type is the classification of reviews. The scikit-learn framework will be used for classification.

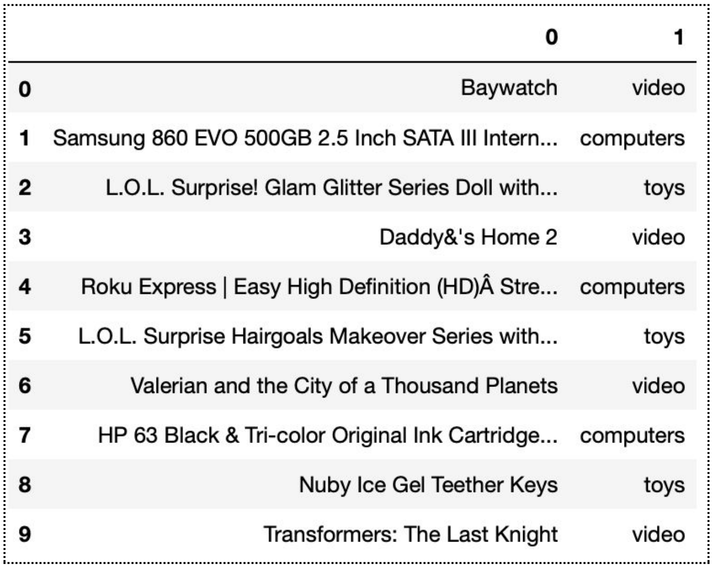
4.1 Classification of categories of product

Classification of products will be done on three categories, video, computer, and toys.

4.1.1 Data Preprocessing

We don't have to manually label the data by ourselves. When the data is crawled, the category field is automatically included. The reason we are doing classification on category even though we already have category field is that in the future, if the product does not include category field anymore, we will be able to put it by ourselves.

Dataset to be trained will be created first. All three categories have more than 1400 products. However, only 1400 products will be taken from each category. Name of the product is taken as an input feature and category of the product is taken as a target feature. Dataset has created in a way that it has a good distribution of all products.

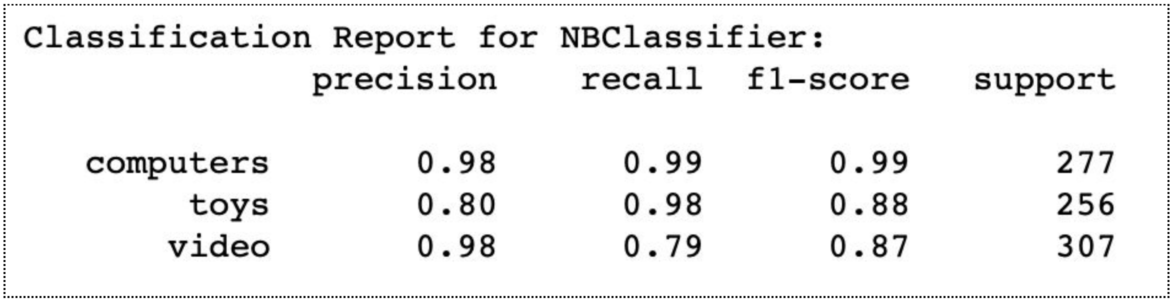


	0	1
0	Baywatch	video
1	Samsung 860 EVO 500GB 2.5 Inch SATA III Intern...	computers
2	L.O.L. Surprise! Glam Glitter Series Doll with...	toys
3	Daddy&'s Home 2	video
4	Roku Express | Easy High Definition (HD)Â Stre...	computers
5	L.O.L. Surprise Hairgoals Makeover Series with...	toys
6	Valerian and the City of a Thousand Planets	video
7	HP 63 Black & Tri-color Original Ink Cartridge...	computers
8	Nuby Ice Gel Teether Keys	toys
9	Transformers: The Last Knight	video

Figure 4.1. Classification of Categories of Products

Naive Bayes classifier (NB) and SVM classifier are chosen as the testing classifier and the best of them will be used as the final classifier.

Training set and testing set are split in the ratio of 80:20. Before, data are put into the classifier, they are changed into word vectors. CountVectorizer from scikit-learn is used for this. Not only that, the weightage of the most common words are reduced by computing TF-IDF. The TfidfTransformer from scikit-learn is used for this. Then, they are trained and evaluated using NB classifier.

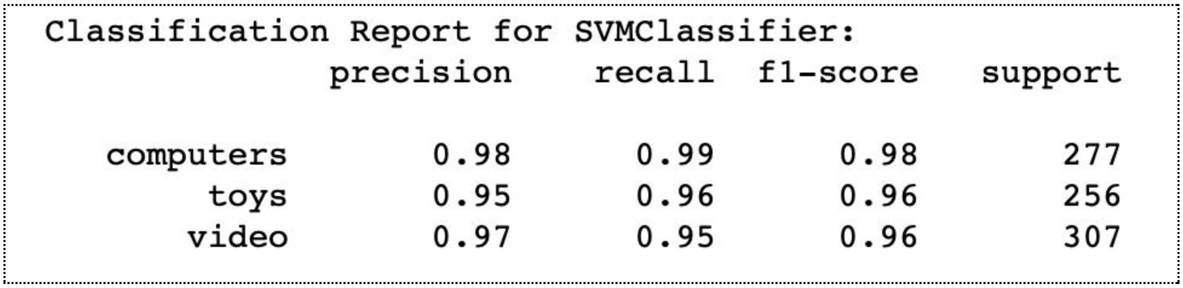


```
Classification Report for NBClassifier:
                precision    recall  f1-score   support

   computers        0.98      0.99      0.99       277
        toys        0.80      0.98      0.88       256
       video        0.98      0.79      0.87       307
```

Figure 4.2. Classification Report for Naive Bayes Classifier

According to the report, classifier did a great job on computers, since both precision, recall, and F1-score of computers are quite high. It could be the case that classifier is overfitting on computer category. For toys, although its precision is the lowest among the three, it has a good recall. The video is in the reverse condition because it has a low recall and highest precision.

Now, SVM classifier will be tested.



```
Classification Report for SVMClassifier:
                precision    recall  f1-score   support

   computers        0.98      0.99      0.98       277
        toys        0.95      0.96      0.96       256
       video        0.97      0.95      0.96       307
```

Figure 4.3. Classification Report for SVM Classifier

All the score improve significantly. So, SVM classifier will be used as the final classifier.

4.1.2 Performance metric

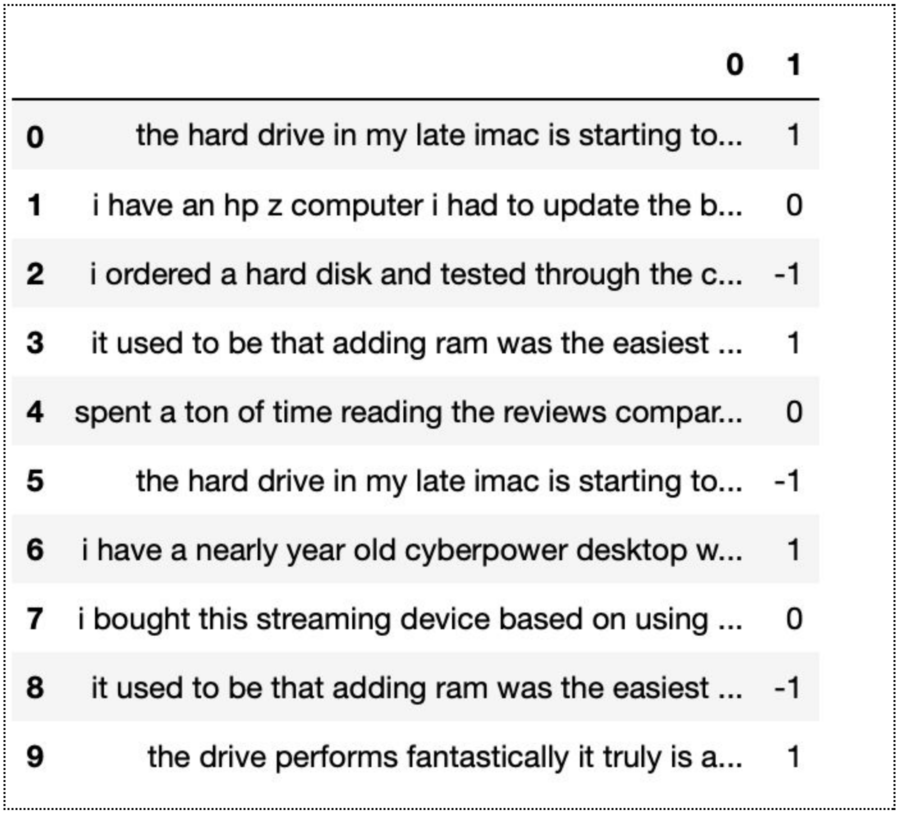
It took 0.05s to classify 4200 records. So, it can classify 84000 records per second.

4.2 Classification of Reviews

Reviews will be classified as either good review (1), average review (0) and bad review (-1). For now, we are displaying the review summary. However, in the future, after classifying the reviews of each product into good reviews, average review, and bad review, we can use the same method as the text-summarization to select the most relevant good review, average review and bad review to display them.

4.2.1 Data Preprocessing

All the reviews made on the products of computers will be used as a training set. Dataset will be prepared in a way that if a user gives a rating of 5, his/her review sentences will be labeled as a good review, if a user gives a rating of 4, it is labeled as average review and for the rest rating, they are labeled as a bad review. In addition, all the review sentences are converted into lowercase. Numbers and other types of characters are removed from the reviewed. An equal amount of reviews from three categories, each has 3160 reviews, totaling in 9,480 sentences in the dataset.

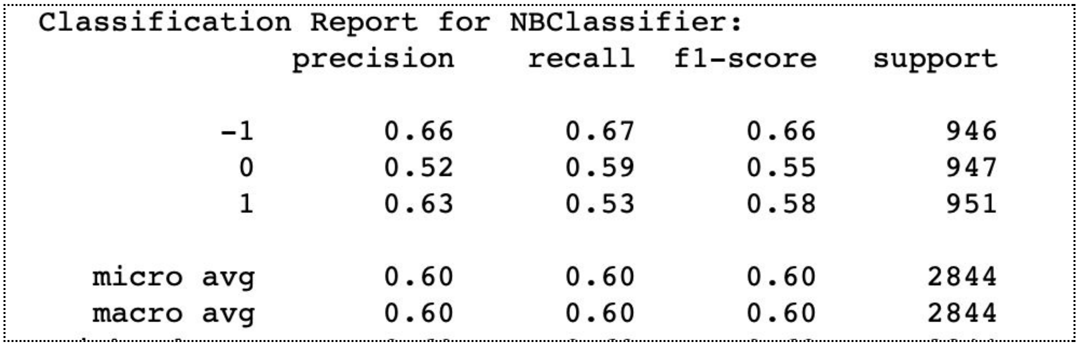


	0	1
0	the hard drive in my late imac is starting to...	1
1	i have an hp z computer i had to update the b...	0
2	i ordered a hard disk and tested through the c...	-1
3	it used to be that adding ram was the easiest ...	1
4	spent a ton of time reading the reviews compar...	0
5	the hard drive in my late imac is starting to...	-1
6	i have a nearly year old cyberpower desktop w...	1
7	i bought this streaming device based on using ...	0
8	it used to be that adding ram was the easiest ...	-1
9	the drive performs fantastically it truly is a...	1

Figure 4.4. Dataset classified based on reviews

There will be two additional steps here before they are converted into word vectors. Since all the reviews are very long texts, stopwords will be removed from it first. Stopwords from NLTK are used to extract the stopwords from review sentences. Then, they will also be stemmed to make it simpler. SnowBall Stemmer is used for this case. After that, they are split into train set and test set in 70:30 ratio, since the dataset is quite big and a little bit more portion is given to test set. After that, the same steps are performed as the first classification before they are put inside classifier.

Firstly, NB classifier will be used.

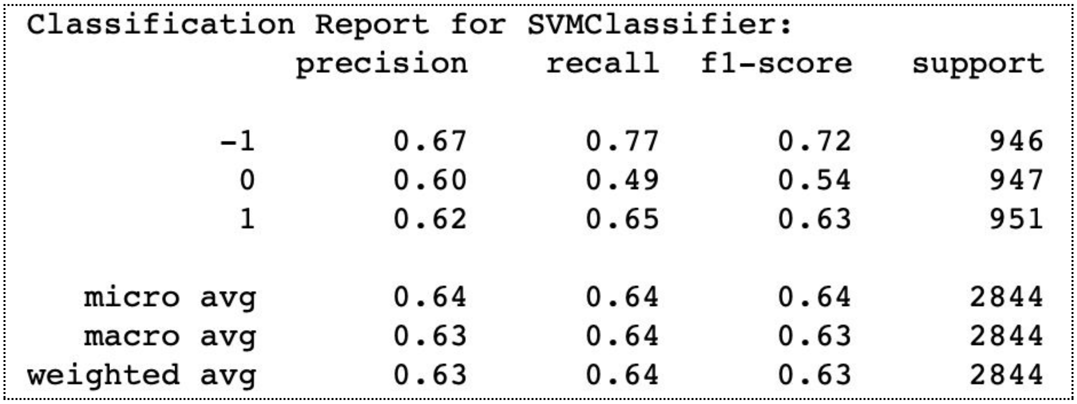


```
Classification Report for NBClassifier:
              precision    recall  f1-score   support

          -1       0.66      0.67      0.66       946
           0       0.52      0.59      0.55       947
           1       0.63      0.53      0.58       951

   micro avg       0.60      0.60      0.60      2844
   macro avg       0.60      0.60      0.60      2844
```

Figure 4.5. Classification Report for Naive Bayes Classifier

With NB classifier, the system is able to detect the bad review more than good and average review. It has the highest precision and recall and f1-score. For an average and good review, performance is quite fair. Average review classification has a higher score in recall where the good review has a higher score in precision and f1-score.

Then, the SVM classifier will be tested.



```
Classification Report for SVMClassifier:
              precision    recall  f1-score   support

          -1       0.67      0.77      0.72       946
           0       0.60      0.49      0.54       947
           1       0.62      0.65      0.63       951

   micro avg       0.64      0.64      0.64      2844
   macro avg       0.63      0.64      0.63      2844
weighted avg       0.63      0.64      0.63      2844
```

Figure 4.6. Classification Report for SVM Classifier

Once again, SVM classifier gives better results. Precision for all three reviews has improved. However, the system is still performing better in classifying bad reviews. It has the highest precision, recall, f1-score among three reviews. Its recall for average review is dropped and f1-score is dropped a little bit. All three score is improved for good review. Therefore, it will be used as the final classifier.

4.2.2 Performance Metric

It took 6 ms to classify 9480 records. So, it can classify 1580000 records per second.

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

4.3.1 Ensemble methods

Ensemble methods is a combination of several base models to improve predictive performance. There are two types of ensemble methods.

The first type of ensemble method is called the averaging methods. This works by building several base estimators and average the predictions. This method is usually better than the single base estimator as it reduces its variance. One of the examples of averaging methods is the bootstrap aggregating (bagging) methods. Bagging methods will take a few random subsets of the original training set and form a final prediction by doing an aggregate on the individual predictions. This method reduces overfitting and works best with strong and complex models.

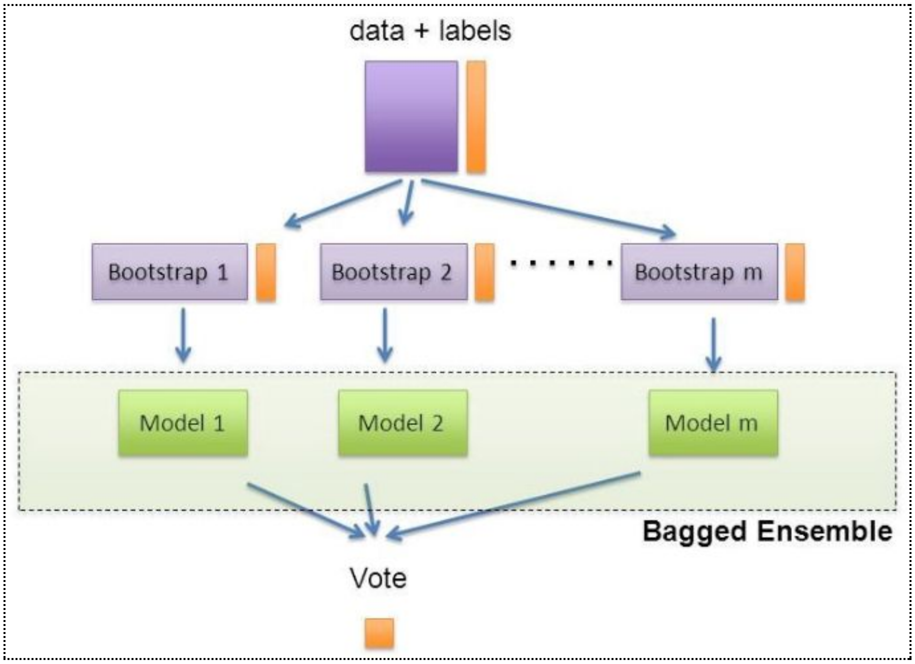


Figure 5.1. Bagging method

Bagging works effectively when there are limited data and can get an estimate by aggregating the scores over many samples. There are two types of voting. The first type of voting is hard voting, where a majority of the classifiers will determine what the result to be. In Figure 5.2 below, it shows that H_i are the bagged models, and based on the prediction, it produced the final prediction by voting.

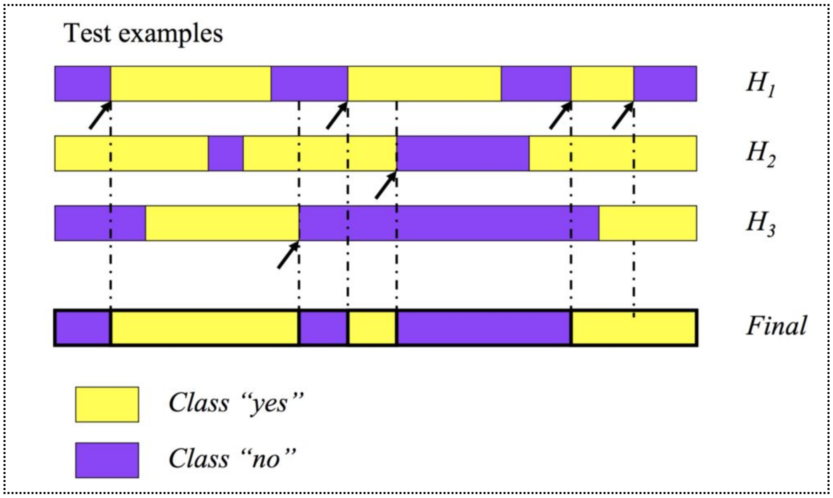


Figure 5.2. Voting in the bagging method

The second type of voting is soft (weighted) voting. It computes a percentage weight with each classifier. A predicted class probability from each model for each record is collected and multiplied by the classifier weight and finally, average it. The final prediction is then derived from the prediction with the highest average probability. However, in reality, it is difficult to get the weights. Therefore, to counter this process, a linear optimization equation or neural net could be constructed to find the correct weighting for each of the models for better accuracy.

Another example of averaging methods is the random forests. The sample drawn with replacement from the training set will build each tree in the ensemble. A random subset of the feature will be selected and further randomizing the tree. Due to the randomness, the bias of the forest will slightly increase. However, due to the averaging, it decreases the variance as well which result in an overall better model.

The second type of ensemble method is called the boosting methods. This works by building the base estimators sequentially which reduce the bias of the combined estimators. The overall performance will be improved by combining the weak models into a strong ensemble. One of the examples of boosting methods is called AdaBoost. By fitting a sequence of weak learners such small decision trees or models that are only slightly better than random guessing, on weighted versions of the data. The predictions are combined through a weighted majority vote or weighted sum to produce the final prediction. The data modifications at each boosting iteration consist of applying weights to each of the training samples. Initially, the weights are all equal. After each successive iteration, the sample weights are modified individually and the learning algorithm is reapplied to the reweighted data. Those training examples that were incorrectly predicted by the boosted model induced at the previous step have their weights increased while those that were correctly predicted will have their weights decreased.

After understanding the different type of ensemble methods, the average method (bagging method) could be applied to our system. This is because the accuracy of both NB and SVM are quite similar. By creating multiple NB and SVM classifiers as seen earlier in Section 4.1 and Section 4.2, it could increase the accuracy for both categories of products and reviews for the user.

5. Links to Files

1. Youtube link to the presentation
   https://www.youtube.com/watch?v=0AJvQt3BLok&t=4s
2. Google Drive link for queries and classification result
   https://drive.google.com/open?id=1BRoj-twP3Trn4j7Tok6M6jEhy0B8Lao7
3. Google Drive link for Source Codes
   https://drive.google.com/open?id=1i6shx9jtfuyiB6ZjejVgSF2tp1vh0RvT

6. References

[1] G. Gayatree, E. Noemie, and M. Amélie, "Beyond the stars: improving rating predictions using review text content," in *WebDB*, 2009, vol. 9, pp. 1-6: Citeseer.