

Information Retrieval

Group 31

Team Members

- Bhargav Singapuri (U1920026B)
- Luo Minjuan (N2202356J)
- Samarth Agarwal (U2020109G)
- Wu Jingyuan (U1920617K)
- Toh Si Min Jaclyn (U1922432H)
- Kenn Lim Zheng Jie (U1921807J)

Links

- Youtube: <https://youtu.be/EPTur0ypL6k>
- Code and Data: <https://github.com/majulahsingapuri/info-retrieval-project/>

Question 1

Crawling and Storage

There are 2 sources of data for this project. The first is an existing dataset on car reviews that we found from an online source¹ and the other that we scraped from a public website². The former was formatted as XML files that had to be extracted and then was subsequently stored to the Solr server that we instantiated. The latter is scraped when the user requests for data on a particular vehicle and the data is not available or up to date in our Solr server.

Application and sample queries

This search engine is designed for persons that want to quickly get some information about a particular vehicle that they are interested in or look for certain qualities that they want from a car but are unsure of the exact vehicle that they want to buy. Since our system allows the user to filter by the vehicle make, model and year in addition to full text search for key terms, it is an easy and helpful way to quickly get information on cars.

Some sample queries are as follows:

1. Reviews for a Nissan Altima 2022: /select?
q.op=AND&q=MANUFACTURER%3Anissan%0AMODEL%3Aaltima%0AYEAR%3A2022
2. Reviews for a car with good mileage and easy to park: /select?
q=TEXT:mileage%0ATEXT:great%0ATEXT:%20easy%0ATEXT:%20Park&q.op=AND&indent=true

Records, words, and types in the corpus

In the first dataset, there are 3,591,342 total tokens and 106,959 unique tokens.

Code Breakdown for cars.com Scraper

The code starts by importing the required modules, which are requests, BeautifulSoup, dateutil, pandas, and re. These modules provide functionalities to send HTTP requests, parse HTML content, convert dates, work with data in a structured manner, and use regular expressions, respectively.

The function `scrape_car_reviews` take three arguments: manufacturer, model, and year. These arguments are used to construct the URL used to send the request to the website using the requests module. But first, the code needs to replace ‘’ and ‘-’ in the model and manufacturer strings with ‘_’. This is done to construct a URL that is compatible with the website's URL structure. The URL is then constructed using the formatted string method.

The code then searches for all review containers using the class "consumer-review-container". For each container, the code extracts the title of the review, author data, date, author, and content of the review.

¹ <https://archive.ics.uci.edu/ml/datasets/opinrank+review+dataset>

² <https://www.cars.com/research/>

The author data is found using the "review-byline" class and then parsing the HTML to extract the name. The date is parsed using the dateutil module, which automatically detects the date format and converts it to a datetime object. The author is extracted using a regular expression pattern and searched for a match string. The title and content are found using the "h3" and "p" HTML tags. All the extracted data is stored in a list of dictionaries.

Finally, the data is converted to a Pandas DataFrame using the from_records() method and returned.

Comment: By using various modules such as requests, BeautifulSoup, and Pandas, the code can extract the relevant information and store it in a structured way. The code is flexible enough to be used for other car models and years by modifying the input arguments of the function.

Indexing and Querying

Web Interface

The web interface which we have designed utilises the React JavaScript library. The final design of the Web interface consists of 4 main components, namely the Search bar, the filter, the pie chart as well as the post area.

The Search bar as shown in Fig 2a. consists of the search area and the search button whereby the user keys in their desired search input within the search area and clicks on the search button to generate the desired response.



Fig2a. Search Bar

The filter button opens a Snack bar Widget as shown in Fig 2b. whereby the user can filter details of their desired search result such as Year, Manufacturer and Model. The user is also able to sort the display of the results by order of Most Helpful or Least Helpful which are represented by the number of votes the posts receives.

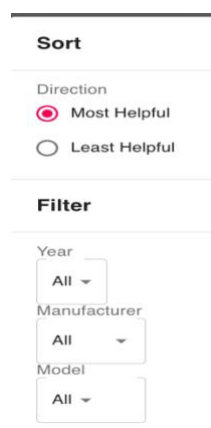


Fig 2b. Snack bar Widget

The pie chart in Fig 2c. below gives users a visual representation of the sentiment of search results obtained from the user’s queries.

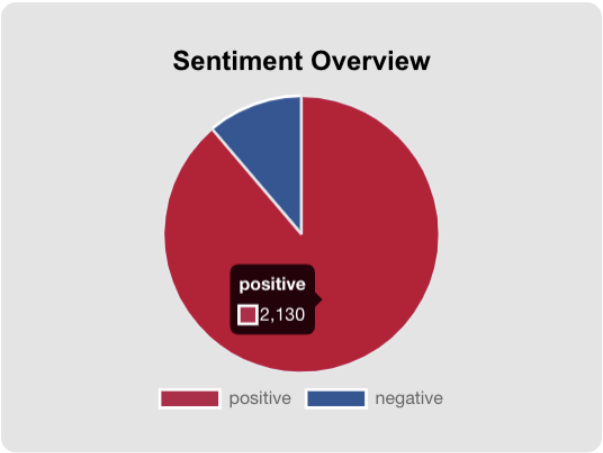


Fig2c. Pie chart

The post area displays the reviews which are relevant to the user’s search query as shown in Fig 2d. Each post corresponds to a single review and contains Year, Text, Manufacturer, Model, Label, Votes as well as two buttons “Useful” and “Not Useful”.

Year	2007
Comment	The Yaris has the looks and then some. For 106 hp, this car has pick up. I have gone 100 mph on it, and it feels very stable (except when there's a lot of wind the car feels like it's floating and not on the ground). Do not be scared to drive it on the highways. I have gone on multiple road trips with it and it is so much fun, especially driving through windy roads. I could not imagine driving any other car on highway 1 (of course except a Ferrari).
Manufacturer	toyota
Model	yaris
Label	Postive
Votes	123

Comment by zippyaris on 21/08/2007

Useful

Not Useful

Fig 2d. A post in the post area

With the different elements of the search engine pieced together, the final UI for our search engine is as shown in Fig 2e. below.

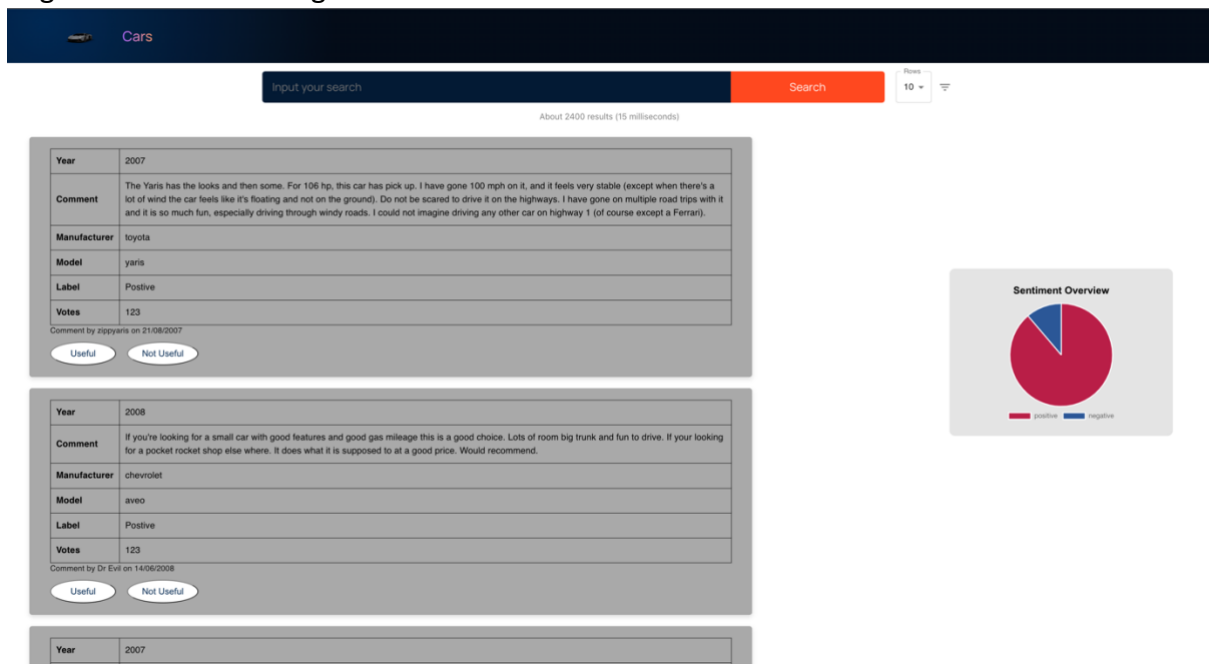


Fig2e. Search Engine UI

Query Flow

When using the webpage, the user can type in the keywords within the search bar. Upon entering the search button, the web-application processes the User's search input and sends a search request to our Solr Server. The search results are then fetched and displayed in the post area. When filters are applied, the search request to the Solr Server is altered accordingly to retrieve the filtered results. The search results fetched correspond to reviews which text are relevant to the User's Search input. By default, the posts displayed are in descending order of helpfulness, meaning that posts with the highest number of votes are displayed first.

Sample Queries


Query 1. Keyword: mileage

Filters: None

Number of results: 497 Posts returned.

Query Content: All posts in containing the word 'mileage' in the review text.

Query Speed: 15 ms

 Cars

Rows 10

About 497 results (15 milliseconds)


Year	2008
Comment	If you're looking for a small car with good features and good gas mileage this is a good choice. Lots of room big trunk and fun to drive. If your looking for a pocket rocket shop else where. It does what it is supposed to at a good price. Would recommend.
Manufacturer	chevrolet
Model	aveo
Label	Positive
Votes	123

Comment by Dr Evil on 14/06/2008

Year	2008
Comment	I bought an HHR because I need a practical vehicle with good gas mileage. For those who like the '40s chopped top hot rod look, the HHR hits the mark and surely is set apart from its Asian counterparts. I see the hard surfaces in the interior as a virtue, since my kids always come back muddy from bicycle sports events and its durable as easy to clean for this purpose. I opted for the SS version that was engineered by GM Performance Division. It handles like a sports car and has the acceleration of a rocket. My experience with actual fuel mileage is about 27.5mpg for combined 70/30 hwy/city with the 2.0L Turbo. I'm very happy with my HHR SS and it fits my specific transportation needs well.
Manufacturer	chevrolet
Model	hhr
Label	Positive
Votes	123

Comment by 87silver on 28/03/2009

Sentiment Overview



positive negative


Query 2. Keyword: mileage good

Filters: None

Number of results: 834 Posts returned.

Query Content: All posts containing either 'mileage' or 'good' in the review text.

Query Speed: 28 ms

 Cars

Rows 10

About 834 results (28 milliseconds)


Year	2008
Comment	If you're looking for a small car with good features and good gas mileage this is a good choice. Lots of room big trunk and fun to drive. If your looking for a pocket rocket shop else where. It does what it is supposed to at a good price. Would recommend.
Manufacturer	chevrolet
Model	aveo
Label	Positive
Votes	123

Comment by Dr Evil on 14/06/2008

Year	2008
Comment	I bought an HHR because I need a practical vehicle with good gas mileage. For those who like the '40s chopped top hot rod look, the HHR hits the mark and surely is set apart from its Asian counterparts. I see the hard surfaces in the interior as a virtue, since my kids always come back muddy from bicycle sports events and its durable as easy to clean for this purpose. I opted for the SS version that was engineered by GM Performance Division. It handles like a sports car and has the acceleration of a rocket. My experience with actual fuel mileage is about 27.5mpg for combined 70/30 hwy/city with the 2.0L Turbo. I'm very happy with my HHR SS and it fits my specific transportation needs well.
Manufacturer	chevrolet
Model	hhr
Label	Positive
Votes	123

Comment by 87silver on 28/03/2009

Sentiment Overview



positive negative

Query 3. Keyword: * Empty Input *

Filters: Manufacturer: BMW

Number of results: 87 Posts returned.

Query Content: All posts containing with the Manufacturer BMW

Query Speed: 15 ms

The screenshot shows a web interface for car reviews. At the top, there's a search bar with the text "input your search" and a "Search" button. Below the search bar, it says "About 67 results (15 milliseconds)". A modal window is open in the center, showing "Sort" and "Filter" options. The "Sort" section has two radio buttons: "Most Helpful" (selected) and "Least Helpful". The "Filter" section has three dropdown menus: "Manufacturer" (set to "bmw"), "Model" (set to "All"), and "Year" (set to "All"). To the right of the modal, there's a "Sentiment Overview" pie chart showing a large red slice for "positive" and a small blue slice for "negative". Below the modal, there are two review cards. The first card is for a 2008 BMW x5, with a positive label and 40 votes. The second card is for a 2008 BMW 5 series, also with a positive label and 40 votes. Each card has a "Useful" and "Not Useful" button at the bottom.

Query 4. Keyword: Power

Filters: Manufacturer: BMW

Number of results: 6 Posts returned.

Query Content: All posts with the Manufacturer BMW and review text containing the word "power"

Query Speed: 10 ms

The screenshot shows the same web interface as before, but with the search bar containing the word "power". It says "About 6 results (10 milliseconds)". The modal window is not open. The "Sentiment Overview" pie chart shows a large red slice for "positive" and a small blue slice for "negative". Below the modal, there are two review cards. The first card is for a 2008 BMW 7 series, with a positive label and 40 votes. The second card is for a 2008 BMW 3 series, with a positive label and 0 votes. Each card has a "Useful" and "Not Useful" button at the bottom.

Query 5. Keyword: good speed fuel

Filters: Manufacturer - Toyota, Model - corolla

Number of results: 6 Posts returned.

Query Content: All posts with the Manufacturer Toyota, Model Corolla and review text containing either the words 'good', 'speed' or 'fuel'.

Query Speed: 25 ms

The screenshot displays a web application interface for car reviews. At the top, a search bar contains the text 'good speed fuel' and a 'Search' button. Below the search bar, a status bar indicates 'About 6 results (25 milliseconds)'. The main content area shows a list of review cards. The first card is for a 2007 Toyota Corolla, with a positive label and 0 votes. A second card is for a 2008 Toyota Corolla, also with a positive label and 0 votes. A third card is partially visible for a 2007 model. A modal window is open over the first card, showing 'Sort' options (Most Helpful, Least Helpful) and 'Filter' options (Manufacturer: toyota, Model: corolla, Year: All). To the right of the review cards, a 'Sentiment Overview' pie chart is shown, with a legend indicating red for positive and blue for negative sentiment. The chart shows a very small blue slice, indicating a low percentage of negative reviews.

Innovations


In our project, we have two main innovations which are implemented.

Enhanced Search

Considering the use of our search engine, which is to search for car reviews, it is safe to assume that a large proportion of users are likely to be interested in purchasing a car and their ultimate intent would be that after viewing the reviews, they would decide on a car which seems the most desirable. Therefore, we utilise the classification based on sentiment analysis done in the other parts of the project to visualize a pie chart of the sentiments of the user's search results. This would allow the user to get a quick feel of his search results without having to scour the entire list of results returned and would help him to narrow down their choices more quickly.

For example, in searching for Honda Civic Cars we see immediately from the pie chart as shown in the Figure below that almost 20% of the reviews are negative. Hence, a user who is

exploring his options could save time and perhaps search for other vehicles since the sentiment of the reviews of this vehicle model indicate that it is not desirable.

 Cars

Search

10

About 125 results (27 milliseconds)

Year	2008
Comment	We've owned many domestic vehicles and decided to step-up to the "superior" quality of a Honda. What a disappointment! The first year was okay with the vehicle however it's now falling apart. Performs poorly in the cold, if it starts at all. Don't think of driving in winter weather without snow tires. The alarm is toast. The electronics are on the fritz. Seat belts don't retract. Weather stripping is worn through. Loud clunk in the trunk area when making turns. Has a severe shake when approaching highway speeds. Hard to find a comfortable driving position. Overall, the most disappointing vehicle we've ever owned! Will never buy a Honda again.
Manufacturer	honda
Model	civic
Label	Negative
Votes	81

Comment by abandon on 10/02/2009

Useful

Not Useful


Year	2007
Comment	I originally went to the dealership to buy an EX model with the navigation option but as soon as I saw the Si, I had to drive it. I got reeled in by the sound of the exhaust which is mellow but strong and not "boy racerish". The 6 speed is silky smooth. The car is not fast by any stretch of the imagination. I have had cars that made 350hp with plenty of torque and this isn't one of them. I was never under the assumption that I was buying anything more than a Civic to use as my primary work vehicle. It is a nice, sporty looking, decent performing compact sedan but you don't notice the performance until you get above 6000 rpms. That is where the little 2.0L really shines.
Manufacturer	honda
Model	civic
Label	Positive
Votes	40

Comment by Robert T. on 22/06/2007

Useful

Not Useful

Sentiment Overview



positive negative

Interactive Search

To further enhance the querying process of the user, we have included the “Useful” and “Not Useful” buttons. This allows users to increase or decrease the vote count of votes to reflect the relevance of a particular post. The updated vote counts are stored in the Solr server and changes are reflected in the queries. Since users are also able to sort their query results based on votes, this allows users to easily retrieve the most relevant post to arrive at their decision more quickly as shown in the Figure above as well. By allowing for such interactivity, it alters the relevance for a better experience for all users.

Question 4 Classification

For our classification task, we chose to use a pre-trained RoBERTa model from HuggingFace. The model was pre-trained on the twitter dataset. When deciding on the model we should use for our sentiment analysis, we took many things into consideration. We looked at performance, accuracy, scalability, and ease of implementation. Then looking at state-of-the-art models, there is no one model that can be considered. There are instances where the Bayesian model would be a better approach than a deep learning model. One example would be when the features are independent of each other and can be considered mutually exclusive we could get high performance with the Bayesian approach, and it can be trained with much lesser data than a deep neural network. Therefore, to circumvent the issue of having to train deep neural networks with an enormous amount of data, we went with a pre-trained model. The RoBERTa model builds on the architecture of the BERT model. BERT is a bi-directional encoder from transformers and RoBERTa builds on it by being pre-trained on a larger dataset, alternative training strategies and using a masked language model for pretraining [3]. Roberta’s performance has been shown to be as good or even better than BERT’s in downstream tasks, this is crucial as we fine-tuned the model further using our

dataset. For our data pre-processing, we used tokenization to convert the sentences into tokens and further used padding and truncation for the sentences to be of equal lengths.

The dataset used was a collection of car reviews, which we had found to be highly opinionated. It was observed during the labelling process (detailed below) that nearly all the reviews were subjective, containing phrases such as “fun to ride. Hence, the subtask of subjectivity detection was ignored. The subtasks chosen were feature extraction, via tokenization, and polarity detection, which was the task the model was trained to perform.

In order to fine-tune the model, 2400 reviews were randomly sampled from the dataset and labelled with 2 different methods. The first method involves labelling each review with either ‘1’ if it is positive, or ‘0’ if it is negative. Mixed reviews where the sentiment of the reviewer was difficult to quantify were excluded, as the purpose was to distinguish solely between positive and negative reviews. The second method involves labelling reviews with scores ranging from 1 to 5, with 1 representing very negative, 3 representing neutral, and 5 representing very positive. This labelling is done to prepare for Question 5, where the model will be enhanced via the implementation of fine-grained classification. For each labelling process, 3 independent set of labels were generated, and an inter-annotator agreement of above 80% was achieved. An analysis was then performed on the labelled data, which showed that most of it was positive. Hence, the positive data was under sampled, while the negative data was augmented using backtranslation, to form the final set of training data to be used, with a total of 1000 samples. As the labels were binary, we were comfortable using backtranslation, as any loss in nuance due to the process is unlikely to cause the overall sentiment to flip.

We used the HuggingFace transformer library to be able to import the model and further fine-tune use it our dataset. For our binary classification task, we got the respective scores during training: {accuracy: 0.960, precision: 0.975 recall: 0.951}. We then predicted the labels of our test set and were able to achieve an accuracy score of 96% and f1 score of 96.25%. For our performance metric we took the average time it took for the model to classify the average number of reviews for the car models. After we found the average number of reviews, we repeated the process of find the performance metric (shuffling the dataset for each iteration) ten times to get an average of the performance. It took our model 18 seconds on average to classify 200 reviews. In the future, we hope to perform better optimization to reduce the time taken even further. Especially noting that consumers are constantly looking to get results faster. For scalability, one way to prevent longer wait times would be to show fewer results at the start and allow the user to click more for additional results.

Question 5

To enhance the classification process, we chose to train a separate model to perform fine-grained classification instead. As the goal of the project is to help users pick a car they want to buy, there is a need to distinguish highly positive/negative reviews from the rest, as those reviews are more likely to swing a user towards a certain decision compared to reviews which are only slightly negative or positive. There is also a need to detect mixed reviews, as

improper classification of a mixed review as a positive or negative review may lead to the user coming to a wrong conclusion.

For this, the fine-grained labels were used, which also had a similar problem of having positive labels oversampled. However, we opted not to perform data augmentation as we were concerned that the nuance may be affected by the process (i.e., augmented text comes out very positive instead of slightly positive like the source text). Hence, we opted to balance out the weights by applying a weighted loss to the model instead. [insert results here]

References

- [1] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019, July 26). *Roberta: A robustly optimized Bert pretraining approach*. arXiv.org. Retrieved April 6, 2023, from <https://arxiv.org/abs/1907.11692>