Feature Based Review Categorization

Jenil Shah - 69945843
Devansh Soni - 40036529
Naga Samantha Davuluri - 66501858
Arpitha Rao H S - 45426383
Ashwin Viswanathan - 44198875
Rithi Ramji - 13917106

University of Florida

Abstract

Today, most of the e-commerce websites ask customers to review their products. As e-commerce is becoming more and more popular, the number of customer reviews that a product receives grows rapidly. It becomes tough for a potential customer to make informed decision because the count of the reviews is in hundreds and thousands. In this project, we aim to summarize all the customer reviews of a product. Our system outputs two final products. A classified detailed summary and a consolidated summary. In first product, we are only interested in the specific features of the product that customers have opinions on and also whether the opinions are positive or negative. In second product, we summarize the reviews by selecting subset of the original sentences from the reviews which provides overview of main points.

a. Introduction

With the advent of e-commerce, product sellers have their customers write reviews about their product. Customers often express their satisfaction or dissatisfaction about products through positive or negative reviews. These reviews serve as a reference or basis for people buying the same product in the future. However, with the rapid growth of e-commerce, there is an inadvertent increase in the number of reviews for products. It can therefore be extremely time consuming for the customers to go through every review and understand the common consensus among people who have reviewed the product. In such a scenario, our project Feature Based Review Categorization proves to be useful. It works in three stages as i) Mining the product features which are most often commented on by customers. This is done using frequent itemset generation using Apriori algorithm. ii) Deciding the opinion for every feature as positive or negative. This helps in understanding the number of positive and negative opinions for each feature of the product. iii) Generating a consolidated text summary which is evaluated with a framework called ROUGE or Recall Oriented Understudy for Gisting Evaluation which automatically determines the quality of a machine generated summary by comparing it to summaries written by humans. The Feature Based Review Categorization has been implemented on Best Buy and Amazon datasets which are cleaned and integrated using Amazon Elastic Map Reduce. Prediction is a combination of Natural Language Processing and Data Mining techniques which uses Python and the NLTK library. Finally text summarization is done using Rule Based methods. In addition to above, a User Interface is developed using Sencha Ext JS JavaScript Framework which takes in the product name as search item and generates the important features, number of positive and negative reviews for important features and the consolidated summary as the result. The summary is evaluated using ROUGE for a few products and the results are tabulated. Also, Tableau is used for visualization of the integrated dataset.

The responsibility has been divided among the members as follows - Cleaning and Integration of Amazon and Best Buy datasets have been handled by Ashwin Viswanathan, Naga Samantha Davuluri and Rithi Ramji. Prediction and UI by Jenil Shah and Devansh Soni. Visualization by Arpitha Rao H S.

b. Background

We can see numerous reviews from customers on business website, e.g., amazon, bestbuy etc. Many of them are very lengthy and tedious or verbose and sometimes even unnecessary. Going through all these will be tiresome effort and often not very fruitful. Hence we are summarizing these reviews so that the customer's time and effort is minimized. Most often customer reviews will be in the form of rating and paragraph of opinion about the product (partial/complete).

Related Work

Current works (Pang and Lee, 2002) assume that rating is binary – good or bad, this polarity ease the process, but might not always hold good. For example, amazon provides five-star rating, which means that customers could have five choices for rating instead of two. Besides that rating in itself is very vague. For example, does a three-star rating mean that the product's quality is not that good in customer's opinion or not that bad. There are problems associated with processing textual opinion too. In the syntactic level, sentences in review will contain English sentences and/or snippets and therefore cannot always be successfully parsed.

In the semantic level, sentences in reviews might not even relate to the product. There can be problems with the classic sentimental classification too. It generalizes the opinion of the customer to polarized ones – Good or bad, but it does not provide what reviewer actually liked or disliked. In fact negative polarity does not mean that the customer did not like anything about the product and positive polarity does not mean that the customer liked everything about the product. Many of the related work on sentiment classification is actually only partially knowledge-based. Few of them just focus on classifying semantic orientation of individual words or phrases, using linguistic heuristics or a pre-selected set of seed words (Hatzivassiloglou and McKeown, 1997; Turney and Littman, 2002; Turney and Littman, 2002). There are arguments that human beings might not always choose discriminating words than some statistic methods do (Pang and Lee, 2002).

Pang, et. al experimented if it is sufficient to treat sentiment classification simply as a special case of topic-based categorization (with the two "topics" being positive sentiment and negative sentiment). They have used three standard algorithms: Naive Bayes classification, maximum entropy classification, and support vector machines. Results from these machine learning techniques are good when compared to human-generated baselines. But relative performance from Naive Bayes is worst and SVMs is the best, even though the differences are subtle. Also their approach is based on the assumption that customers' opinions are polarized.

A generalization of overall rating and user comments on several features for each product is provided in (Wang and Ren). They calculate an overall rating of the product based on PIM-IR algorithm and generalizes these comments on features using feature-based classification. Feature Extraction is a difficult problem and that is due to classifying and extracting quotes done by the human beings – website editors and customers, which is a tedious job. Therefore

implementing automation to extract potential features and let the human beings check out whether they are really useful is a very valid option.

Novelty

The uniqueness of our approach is based on an idea provided in (Hu and Liu)**[1]** to extract the features from customer reviews and find the opinion words for these features. We prune the feature set by the relationship between the features and opinion words. Here we use the feature based sentiment classification to automatically extract features and customers' opinions toward these features out of the sentences in the reviews and give customers an overall concise review of positives and negatives of various features of the product.

c. Algorithm and System Description

Our system has two final products:

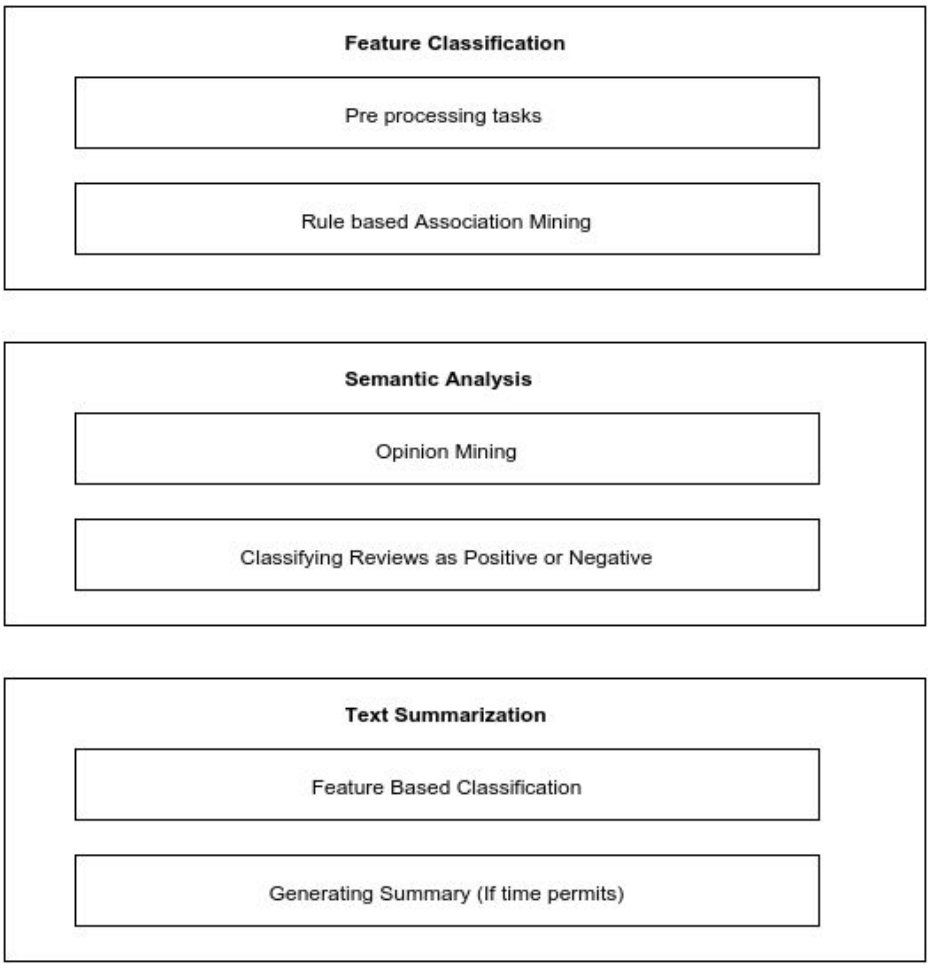
- Sentence classification according to polarity of reviews.
- Text summarization for a particular product.

Final System Architecture

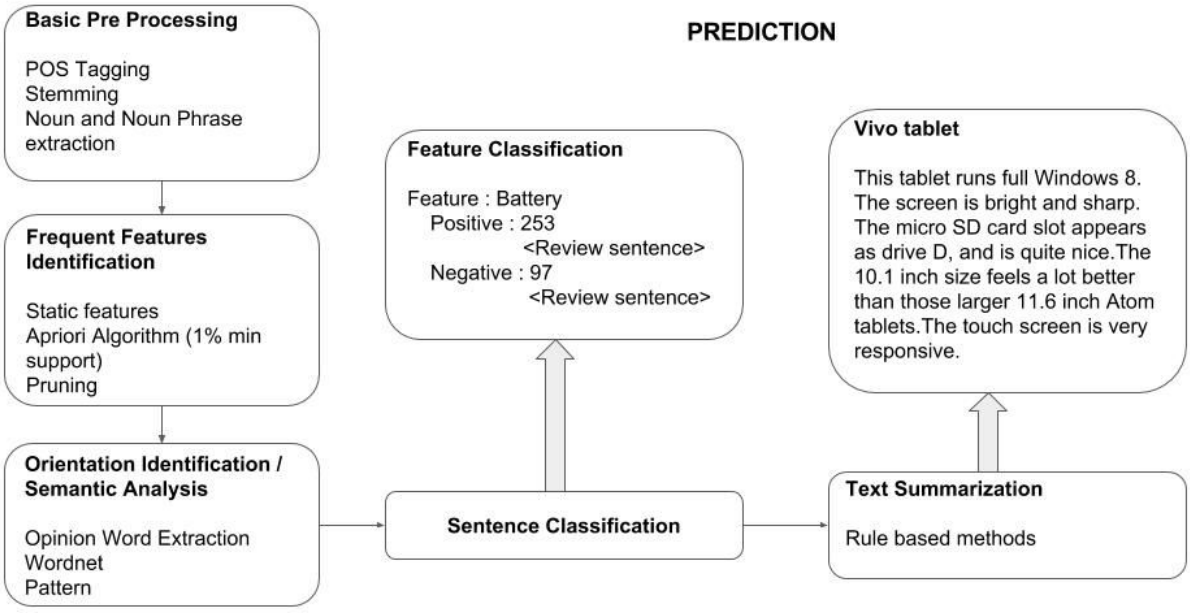
The system does summarization in following three steps:

- Dynamic Identification of the features and review classification
- Determining the polarity of the reviews
- Text classification and text summarization

The prediction pipeline is shown below.



Complete System Architecture for prediction is shown below.



Prediction is combination of Natural Language Processing and Data Mining techniques.

Each system component is explained in detail.

Basic Pre Processing

- **Part of Speech Tagging**

The part of speech tagging is crucial component of the system pipeline. Product features are usually nouns or noun phrases in review sentences. We use NLTK for POS tagging. After tagging, we extract the noun and noun-phrase tagged words. Other components of the sentence are unlikely to be product features

- **Stemming**

In linguistic morphology and information retrieval, stemming is the process for reducing inflected words to their stem, base or root form—generally a written word form. We used Porter stemmer from varied choices of stemmer available. Stemming is used to reduce the occurrence of similar word in different formats.

Frequent Feature Identification

This sub-step identifies product features on which many people have expressed their opinions. Initially we started to build system with Static features. Our system will focus on features that most of the people like or dislike. Currently our system deals with explicit features. Explicit features are those in sentences which give direct indication of noun or noun phrases. We use below mentioned methods for frequent feature finding.

- **Association mining**

Static features have limitations. They are fixated for the given product or similar types of product. Static classification model fails when we deal with varied products. To dynamically extract features from reviews, we use association mining.

Our system uses association rule mining for the following reason. Many things present in customer reviews are not directly related to product features. When customers comment on product features, the words that they use converge. Using association mining to find frequent itemsets is appropriate because those frequent itemsets are likely to be product features.

Algorithm used : Apriori algorithm
Minimum support : 1%

- **Pruning**

Results of association mining do contains many features that are not genuine. Infact, some detected words as features are completely inappropriate. We use pruning methods to eliminate such non-genuine feature items. One such scenario is the plural set. Many features appear along with their plural words.

Ex. For camera features, features identified were picture and pictures. One of the above detected features is redundant. Pruning takes care of such scenarios and remove non-genuine features.

Orientation Identification / Semantic Analysis

After feature extraction, following methods are performed.

- Classification of sentences on basis of features
- Extracting Opinion words from classified sentences.
- Detecting the polarity of the classified sentences using Opinion words.

Classified sentences are grouped sentences according to frequent features. Opinion words are basically the strong and related adjectives present in classified sentences. We use Wordnet dictionary to detect the polarity of the opinion words. To make implementations accurate, we used Pattern[4] to detect the polarity of the sentences.

Sentence Classification

Once polarity of classified sentences are detected, our first product is ready. Sample of our first product is shown below.

Digital Camera:

Feature **Picture Quality**

Positive: 253

<individual review sentences>

Negative: 6

<individual review sentences>

Feature **Size**

Positive: 134

<individual review sentences>

Negative: 10

<individual review sentences>

Text Summarization

For our second product, we need to summarize all reviews of the product in 5-6 sentences. We used rule based methods to generate text summary. From final product one, we picked top 5 features who has most number of reviews. Consolidated text summary contained each sentence of each of top five feature. The polarity of summary sentence of that particular feature is detected by the ratio of positive to negative reviews. Example of text summary is shown below.

Product : ASUS VivoTab Smart ME400 ME400C-C1-WH 10.1-Inch 64GB Tablet
Review Summary : This tablet runs full Windows 8.The screen is bright and sharp.The micro SD card slot appears as drive D, and is quite nice.The 10.1 inch size feels a lot better than those larger 11.6 inch Atom tablets.The touch screen is very responsive.

Visualization of Data Sources

Data Sources : Amazon and Best Buy reviews

In our project, Data sources from Amazon and Best Buy (in CSV format) are visualized using Tableau. As can be seen in Figure 1: Source_Percentage, we show here the percentage of both the data sources we have

used in terms of records. Source is shown by distinct colors and number of records by the size. Here the data is filtered on categories, which keeps actually 2187 of 2187 members.

Figure 1

In Figure 2: Broad Categories, we show the various categories of Amazon and Best_Buy data in terms of both its records(Size of the circles) and Number of reviews per category(as a measure in Tableau). We have used distinct colors for every category. Here Over all categories is broken down by Source. Marks are labeled by over all categories. Also the data is filtered on categories, which keeps actually 2187 of 2187 members.

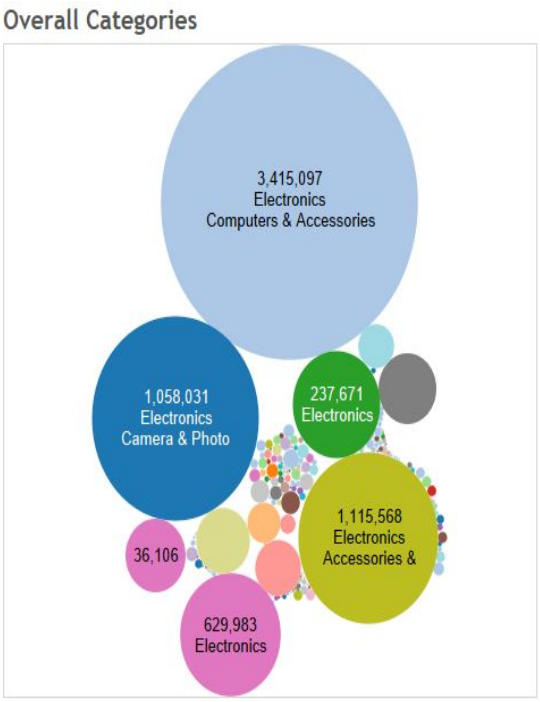


Figure 2

In Figure 3: Overall Categories, we show all categories(Amazon and bestbuy) together and their number of reviews as a single representation. Sum of number of reviews, overall categories and level 1 from the database is depicted in the graph. The marks are labeled by the details of Sum of number of reviews, overall categories and level 1.

In Figure 4: Subcategory_Level1, we show as per our database, the first level of subcategory under the main categories of Amazon and bestbuy datasets, along with their number of reviews. Count of number of reviews and number of records for each source is done here. Color shows details about level 1, count of number of records and number of reviews. Here the view is filtered on exclusions, which therefore keeps 245 members.

In Figure 5: Subcategory_Level2, we show the second level of subcategory under the main categories of Amazon and bestbuy datasets, along with their number of reviews. Here Level 2 and Product_Name from the Database are broken down by the Source. Size shows the sum of number of records. The marks are labeled by level 2 and product name. The view was filtered on product name.

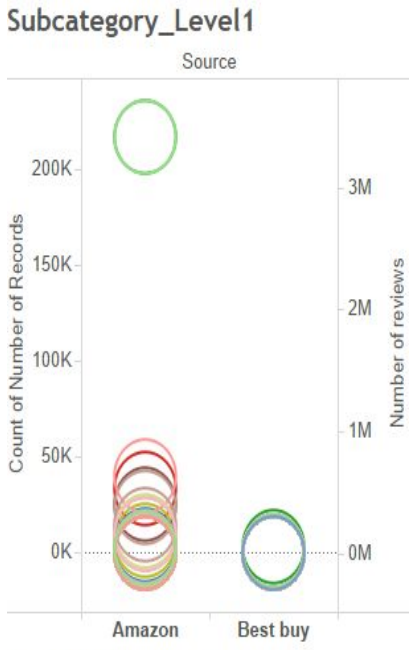


Figure 3

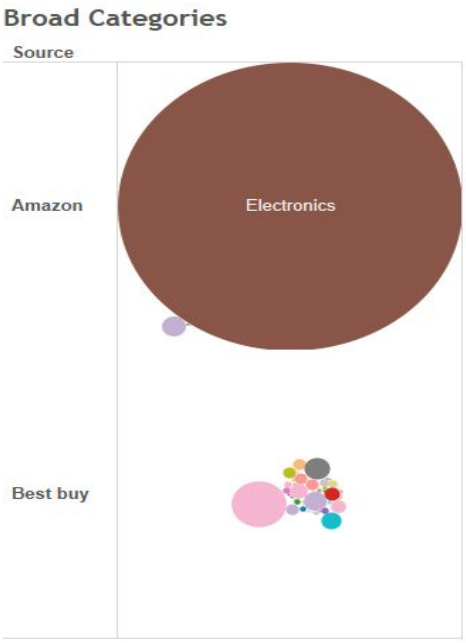


Figure 4

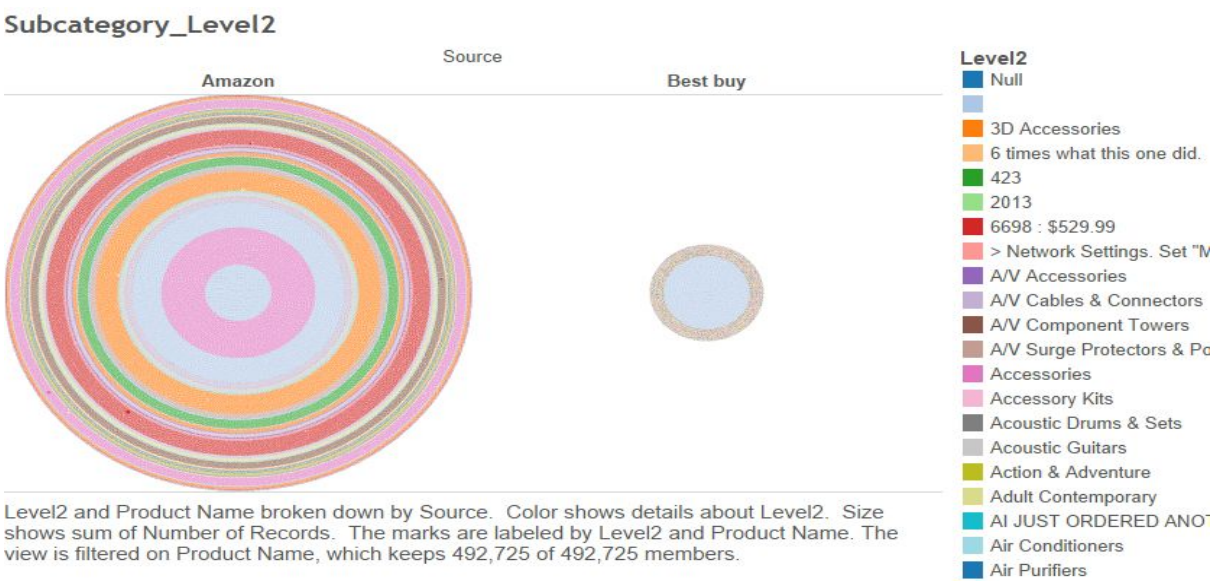


Figure 5

d. Experiments

Datasets

We have used two datasets - Amazon and BestBuy.
Each dataset contains two kinds of files - product information and review information , which are then joined to form one file based on product id. From product information we considered {Product Id, title, List{categories}}. Reviews gave us the {product Id and List{Reviews}}This is done using Map Reduce paradigm. The data in Amazon dataset is in XML format whereas in Best Buy dataset is in JSON format. The joined file from both the datasets in then integrated into a single file based on product title.

Integration

Levenshtein distance is used in finding similarity of the product names among Best Buy and Amazon DataSet. And BestBuy data set had extra information like type of product and the company name within the product title. If a product in BestBuy has more than 2 “-” seperated texts,we removed the last split value as it could have type information in it. But if the data is having just 2 “-” seperated values it is a part of product title. After Integrating based on above metrics this integrated file is sent to prediction module in JSON format and to visualization module in CSV format for their respective purposes.

Measure of Success

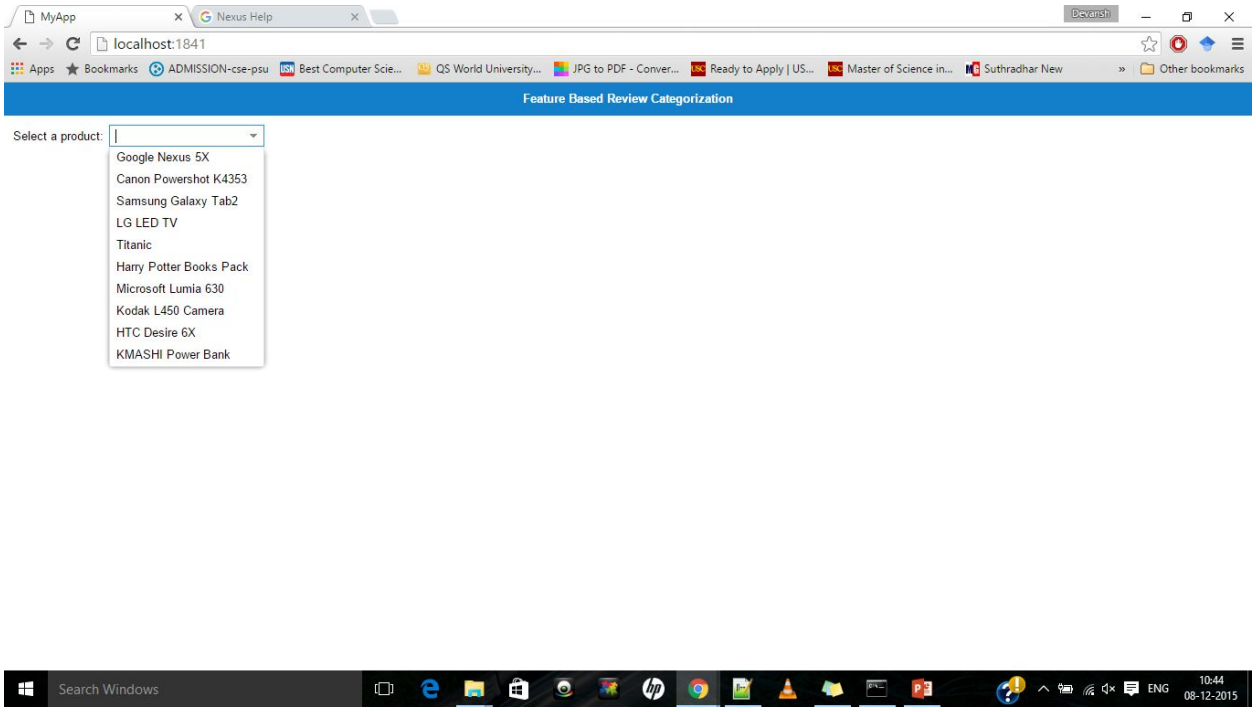
For measuring the efficiency of our system, we have used ROUGE (Recall-Oriented Understudy for Gisting Evaluation) tool. For this purpose, we took 5 products and asked users to summarise the product reviews. We defined framework for manual summarization. For instance, we gave users,the five most popular features of the products based on our system and asked them to write exactly one line about each of these features. The manually generated summary was given to the ROUGE tool, along with the summary generated by our system. The ROUGE tool generated accuracy, precision and F1 measures so that we can evaluate our system. We have used the Java implementation of ROUGE for the evaluation.

User Interface

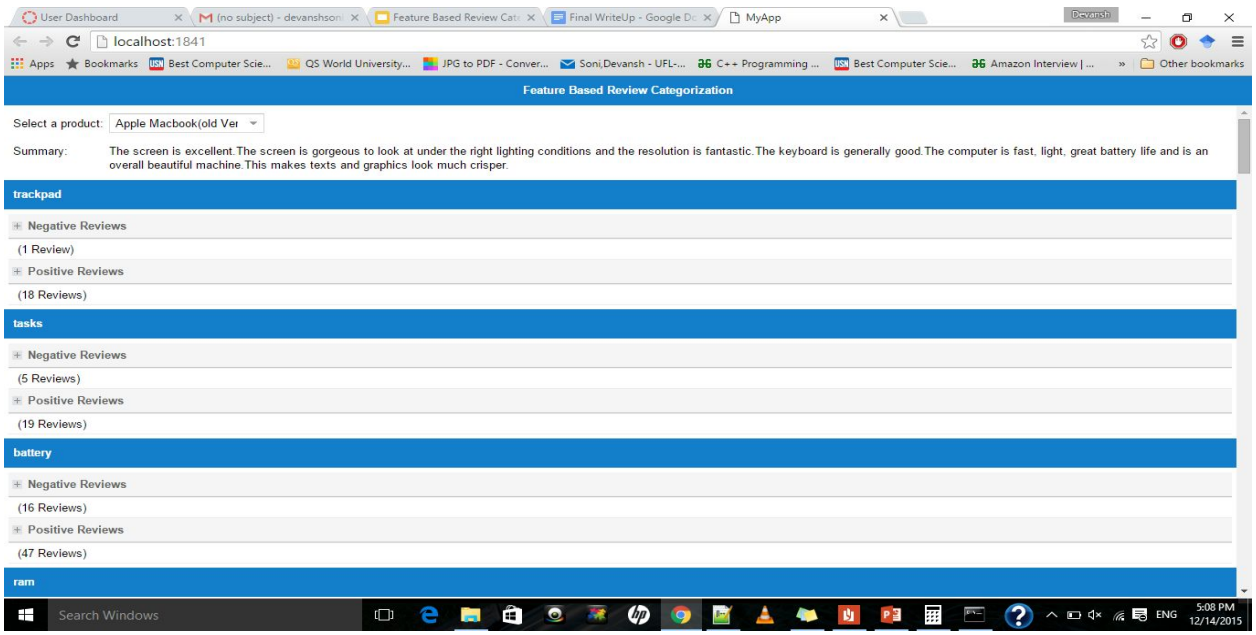
We built a user interface prototype to demonstrate the functions of our system. The home screen has a dropdown containing a list of products. The user needs to select a product from the dropdown to see the review summary and feature based review categorization for the product. Once the user selects a product, a backend API call is made to the server built on top of FLASK framework. Thereafter the code deployed on the server gets executed which sends back a JSON object to the front end. The front end displays the results on the User interface in a presentable form. In addition, the User Interface supports expanded and collapsed views and shows the number of positive and negative reviews for each feature to provide the user with a

gist of reviews.

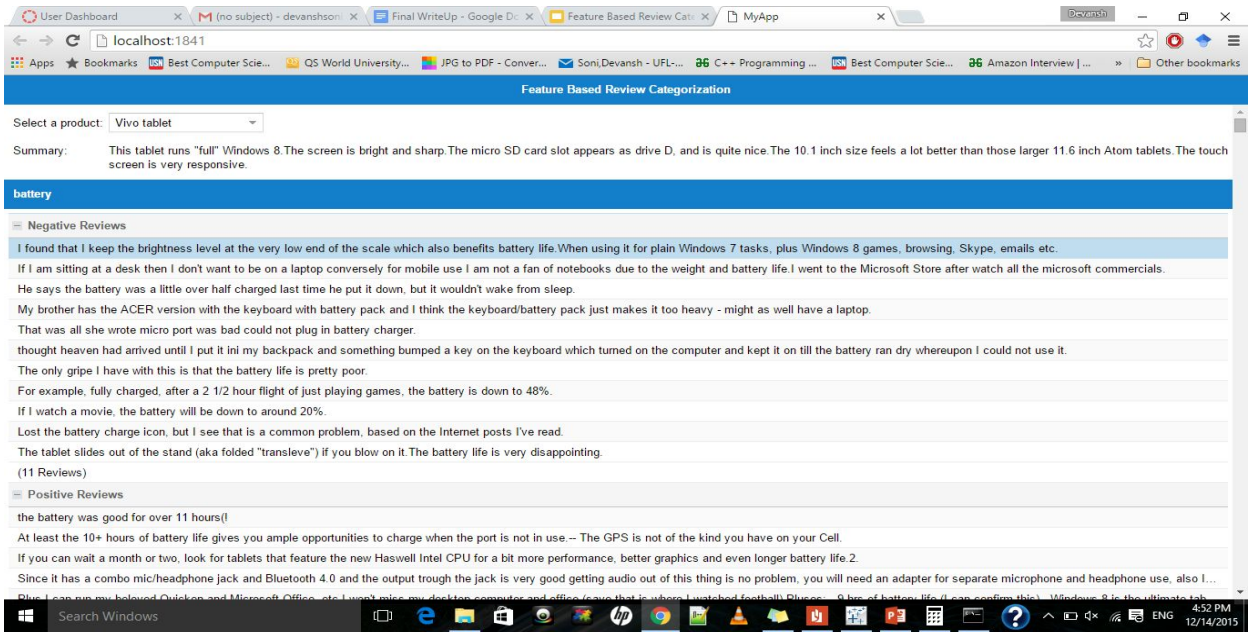
- Home Screen :



- Collapsed View :



- Expanded View :



Experimental Results

The summary evaluation results from the ROUGE tool are as follows :

Product Name	Recall	Precision	F1 - score
Macbook Air	0.57	0.42	0.48
Motorola i730	0.63	0.48	0.54
Acer chromebook	0.69	0.51	0.58
Samsung Chromebook	0.51	0.43	0.46

Effectiveness/Performance

The results displayed above obtained from the ROUGE tool measure the effectiveness of the system. The mean recall is 0.6, the mean precision is 0.46 and the mean F1 is 0.52. Keeping in mind the results are of summary evaluation which does not have predefined measure, our system proves to be effective in review classification and text summarization.

e. Conclusion

Tools used :

- Python (Prediction)
- Python Libraries
    - NLTK (NLP)
    - TextBlob (NLP)
    - Pattern (Sentiment Analysis)
    - Flask (API)
- Hadoop (Cleaning and Integration)
- Ext JS (User Interface)

Challenges:

- **Evaluation**

There is no defined measure to evaluate summarization as there is no data to compare with. Our system summarizes hundreds of reviews into a summary of 5-6 sentences. So we made a framework to evaluate the summary.

For five products we asked 10 humans to review the product in their own words. But humans have to write the summaries in following format :

- Only one sentence per feature.
- Only sentences of top five features classified from final product one.
- No elaborate sentences. Summary should be straight and concise.

We observed that, more human reviews resulted in higher precision and recall. Due to limitations on number of human generated reviews, we cannot quote completely accurate results. The results shown in Evaluation section were obtained by taking into account 10 human reviews per product.

- **Dynamic Feature Extraction**

Obtaining good results with static features, our system aimed to obtain equivalent good results with dynamic features. Extracting dynamic features without compromising accuracy of genuine features extracted was a challenge.

To overcome above challenge, we used Apriori algorithm. Apriori algorithm is followed by pruning to eliminate non-genuine features. Application of Apriori algorithm to extract dynamic features was proposed by (Hu and Liu)[1].

Learnings

- This project provided us with breadth of knowledge in Data Science.
- Covering all pipelines of data science, Cleaning, Integration, Prediction and Visualization, provided us with clear view of data scientist job in industry.
- We obtained another important skill which as per our belief is most important in industry : Implementing research papers
- This project gave us an overview of state of art in domains like Natural language processing and Data Mining.
- We learnt Big data skills of Hadoop and Mapreduce for crunching humongous amount of data.
- We learnt to work in team where we faced situations in which different people have contradictory opinions.

References

[1] Minqing Hu and Bing Liu, Mining Opinion Features in Customer Reviews, Department of Computer Science, University of Illinois Chicago.

[2] Jingye Wang and Heng Ren, Feature-based Customer Review Mining, Department of Computer Science, Stanford University.

[3] Bing Liu. Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers.

[4] De Smedt, T., Daelemans, W. (2012). Pattern for Python. Journal of Machine Learning Research, 13, 2031–2035.