Review Ranking System based on Utility

GourabNath
Praxis Business Scool
Bangalore, India
gourab@praxis.ac.in

Veera Vignesh K
Praxis Business Scool
Bangalore, India
veera.vignesh@praxis.ac.in

Vighnesh Tamse
Praxis Business Scool
Bangalore, India
vighnesh.tamse@praxis.ac.in

Kunal Bharadwaj
Praxis Business Scool
Bangalore, India
kunal.bharadwaj@praxis.ac.in

Durjay Das
Praxis Business Scool
Bangalore, India
durjay.das@praxis.ac.in

***Abstract* — In recent years with the advancement of technology the online shopping has become a convenient shopping place because of the cheap price, variety of products, easy comparison and easy understanding about the products. Customers who have bought the product post their experience (good/bad) about the product which helps other buyers to decide whether the item is good for purchase or not. This is also essential for the manufacturer to understand how the product is perceived. But is it feasible to read all reviews? Finding relevant reviews is like finding a needle in a hay stack. In this paper, we propose a novel method to rank relevant and valuable reviews. We believe this research will help customers as well as manufacturers to understand broadly about the product in a better way. This will help the customer to save a lot of time to choose whether to purchase the product or not. Also, it may help the manufacturer to come up with new ideas to meet the customer needs.**

Keywords — Reviews Ranking, Utility of Reviews, Review Analysis.

I. INTRODUCTION

With the advent of ecommerce providers, it has become a one stop shop for the customers from comfort of their home. The demand for online purchase is increasing day by day because people want to shop without much hassle. People can purchase a product anytime from anywhere without facing the crowd, also having an edge by comparing different products. Customers also share their valuable feedback about their purchased products in the form of reviews and ratings. These reviews are also evaluated by other customers by upvoting or downvoting them.

Let's suppose P_i be a product for which people have provided a set of reviews $R = \{R_1, R_2, R_3 \dots . R_n\}$. Apart from the Review (R_i) that a buyer provides there are other metadata like Review Title, User Name, Photos and Date on which the product was reviewed are captured. Based on the usefulness of the review other potential buyers can either Upvote or Downvote it.

For a Review R_i the sum of Upvote and Downvote is the interaction that the review got from other users. This may help in identifying the good reviews and ranking them. Current ecommerce sites just consider the upvote and downvote which we hypothesize can be biased due to number of factors. In our study we try to consider the content of the review before ranking them and not solely depending on the upvote and downvote.

II. LITERATURE REVIEW

Utility of the Review is vastly studied topic and there are lot of related works in the field. [1] Prepossess an method to automatically rank reviews formulated as regression problem using SVM and measuring the correlation data. But this study didn't take into consideration of the number of photos the user has uploaded.[2] Approaches the problem from two different view point one from the customer impact of the review and the other is manufacturer impact of the review considering the actual text of the review rather than just the metadata like time elapsed from the date of the review, Review Rating etc.. Our method can be implemented from a single stand point at yet serve both the group of peoples.

To best of our knowledge, no prior work has taken no of photos, under consideration to find relevant, important reviews. With this extra features our research help to find most important reviews not only based on average ratings and number of (upvotes, downvotes) but also taking consideration of number of photo which is highly correlated with review ratings.

III. METHODOLOGY

Fig. 1 shows the architectural overview of our review ranking system. The input to the system is the product reviews URL given by the customers, whereas the output ranked reviews based on the predicted score that each review got.

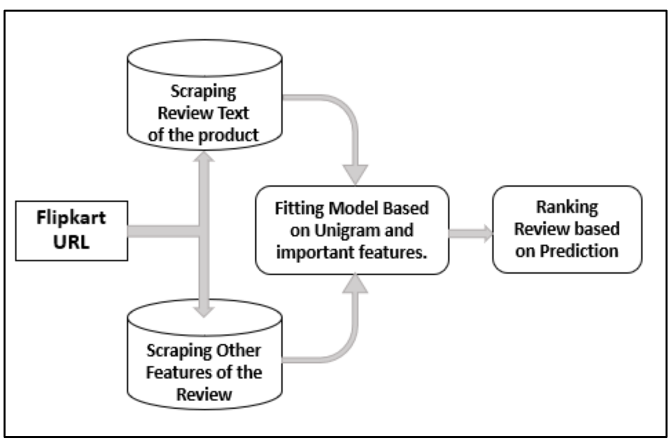

Figure 1. Architecture

Ranking of reviews is done by the following steps

(1) User provides URL of the product.
(2) Review Text are scraped.
(3) Review Metadata are Scraped.
(4) Running a Machine Learning model on the features.
(5). Ranking the reviews based on the scores obtained for each review.

A. URL of the Product

For carrying our experiments, we are considering the products under **Flipkart – Largest ecommerce provider in India.** We hypothesize that review provided by the user doesn't vary across the platforms. It differs only for the product that are provided. Due to this hypothesis the experiments carried out can be expanded to other ecommerce providers.

B. Review Text

Review Text is the voice of a verified buyer on the platform. Hence, the bias of a non-buyer giving comments on the product can be eliminated

The reviewer of the product expresses his feel for the product either good or bad. Mentions what made him satisfied or unhappy.

Review Text is converted into a unigram and their Term Frequency is calculated and multiplied with the Inverted Document frequency to get value of a word in one review. These values are used in the regression model for predicting the interaction score.

C. Other Featuers

Even though review text plays an important role in defining the usefulness of the review, metadata such as Number of Photos that the reviewer has uploaded, Upvote and Downvote and Review Title can also play a significant role in the interaction prediction.

Based on the extracted features we model the interaction as a value between [0,1] and is taken as the target variable **h.**

$$h = \frac{upvotes}{upvotes + downvotes}$$

Higher value of **h** suggests the review is more helpful.

Our System can be modelled as a **regression task** to predict the value of h.

D. Modelling

Extracted Tf-Idf matrix and other features are appended into the dataframe and passed into a regression model to predict the value of h. During testing we calculated the MAPE (Mean Absolute Percentage Error) value of the model and recorded it.

$$MAPE = \frac{\sum \frac{|A - F|}{A} x\, 100}{N}$$

E. Ranking

We took the best performing Model across different segment of products and selected the best features that gave minimum MAPE and used it for deployment.

IV. EXPERIMENTS

In this section, various experiments conducted with the features, selection of model and Selection of feature is discussed. During training phase, we split the reviews into test and train to fit the model on train and test it on the testing data.

A. Data Collection

When the user enters the URL of the product a headless chrome browser instance is fired up and the URL of the review page is loaded using a redirect. To facilitate this, we use Selenium package, which is primarily used for unit testing.

Since the review page contains 10 different reviews but structured in HTML with same class. We can use this to our advantage. Using selenium, we are first finding the read more button in the current page and click on it using inbuild functions in selenium.

Once the entirety of the review is loaded, we can then locate individual element division using its xpath and extract the data from it.

The result review from a page is appended to a pandas [8] dataframe. Similarly, we run a loop over the number of pages for which we need review and extract the reviews and append it to the same dataframe

B. Data Processing and Feature Generation

Before processing the review text important features were created using the following techniques

1) Lemmatization:

We use lemmatization technique from Spacy package [12] to identify the word's lemma or dictionary form. This will be useful while creating Tf-Idf matrix of the words.

2) Remove Stop Words:

Repeated words like conjunction, do not add much value to the content of the review. So, we remove them to get exact ideas out of review. For this spacy's default English stop words are used.

2 New Features are created for further experimentation.

a) Len Before which is a length of review before removing the words, emoji and punctuations.

b) Len After which is a length of review after removing the stop words, emoji and punctuations.

3) POS Tagging:

POS tagger from Spacy [12] library in python was used to produce parts of speech tags for every word in the review across the corpus. Then percentage of noun, verb, adjective, adverb for each review is added to the dataframe as a feature.

Features like the number of emojis per review, number of exclamation mark, number of question mark are derived from the Review text are appended to the dataframe.

A Tf-Idf matrix is created using Scikit-learn's [9] TfidfVectorizor defined under text feature extraction module. Minimum document frequency is kept as 1% to remove the words that are rare.

It was interesting to find that there were very a small number of reviews that used exclamation or question marks. Also, not many reviewers have used Proper Case in their reviews..

The TF-IDF matrix thus created consisted of all the unique words found in all the reviews as columns plus the columns we created during feature engineering stage.

We also used Vader Sentiment Analysis for getting the review sentiments such as positive, negative or neutral and found that a few of the users who have rated the product very low (less than 2 ratings) have positive sentiments. On checking the reviews of these users manually we found that the reviews were actually positive but the ratings given were less. This can be misleading. Thus, Vader sentiment analysis helps in identifying the sentiment based on the content of the reviews provided by the customers.

C. Modelling

Since the target value is modelled as a continuous value between 0 and 1 we looked out for regression techniques, we experimented with various machine learning regression models like SVR [9], XG Boost [10] and Random Forest [9]

We experimented with the available models. On various product groups like laptop, camera, mobile phones. And also, on various parameter pairs created in the above section.

Based on our experiment we found out that Random Forest is the one that has lowest generalization error across product groups.

Random forest gave lowest generalization error on the model with following feature set.

- Review Text
- Number of Photos
- Num of Sentences
- Percentage of Adjective in the sentence.

For further deployment of the model following set of features were used.

User Interface:

To make it more user friendly we have provided a user interface which is a flask app hosted on Heroku. This would help the user to pass the Flipkart product URL and on submit the program at the backend can run the algorithm which separates the positive and negative reviews and rank them as per their predicted helpfulness. Since this is hosted online the time to get the useful reviews will be greatly reduced which well be helpful for both the customers and manufacturers

This website is designed to be responsive and mobile friendly.

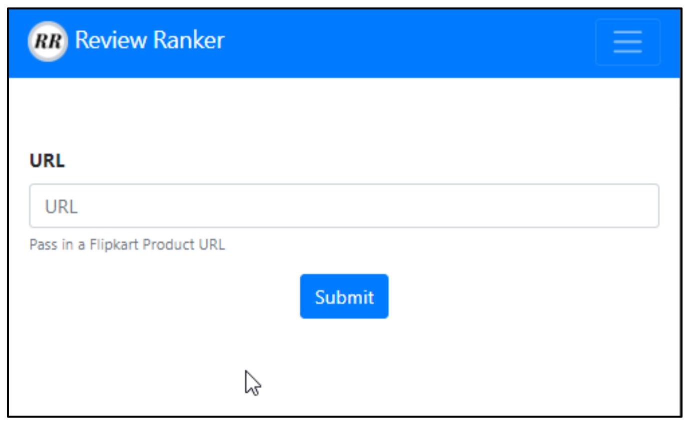

Figure 2 Flask app – Input

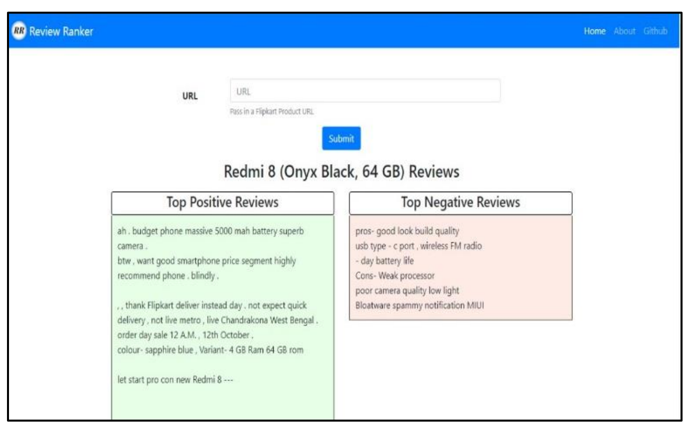

Figure 3 Flask App – Output

https://reviews-ranker.herokuapp.com/

V. FINDINGS

Based on Experimentation we found out the following

- Model
- Feature Set

Experiments showed that Random Forest is best performing across categories with lowest MAPE. We tested our model across mobile phone, Camera and Laptop.

Experimentation with the selected Random Forest Model over the different set of Features gave us the following result.

Feature Set	Mobile Phones	Laptop	Camera	Average Across Product Groups	Span
[RT,NPh,NS,PAdj,h]	13.27	14.22	13.05	13.51	1.17
[RT,NPh,NS,h]	13.62	14.72	12.6	13.65	2.12
[RT,RR,LA,h]	14.02	14.69	12.43	13.71	2.26
[RT,NPh,NS,PV,h]	13.65	14.81	12.7	13.72	2.11
[RT,RR,NPh,h]	14.08	14.88	12.51	13.82	2.37

Figure 4 Test MAPE for Random Forest Model – Across categories

VI. CONCLUSION

In online shopping, reviews play a vital role. Reviews are the opinions of the individual customer. So, when a customer wants to purchase a new product, he/she always consider other people's opinion before investing in it. So, in this paper, we proposed a technique to rank the most suitable reviews about the product based on content of the review itself and not just on features extracted. We generalized our ranking method across different most sold products like electronic gadgets. A time-saving approach is always helping mankind. We hope our method will help millions of customers and manufacturers and save their time to understand the product.

VII. REFERENCES

[1] S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, "Automatically assessing review helpfulness," Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 423–430, Sydney, Australia, July 2006.

[2] A. Ghose and P. G. Ipeirotis, "Designing novel review ranking systems: Predicting usefulness and impact of reviews," Proceedings of the International Conference on Electronic Commerce (ICEC),2007.

[3] A. Ghose, P. G. Ipeirotis, and A. Sundararajan, "Opinion mining using econometrics: A case study on reputation systems," Proceedings of the Association for Computational Linguistics (ACL),2007.

[4] J. Liu, Y. Cao, C.-Y. Lin, Y. Huang, and M. Zhou. "Low-quality product review detection inopinion summarization," Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 334–342, 2007. (Poster paper).

[5] Z. Zhang and B. Varadarajan, "Utility scoring of product reviews," Proceedings of the ACM Conference on Information and Knowledge Management (CIKM), pp. 51–57, 2006.

[6] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[7] Grinberg M. Flask web development: developing web applications with python. " O-Reilly Media, Inc." 2018.

[8] McKinney W, others. Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference. 2010. p. 51–6.

[9] Pedregosa F, Varoquaux, Ga"el, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. Journal of machine learning research. 2011;12(Oct):2825–30.

[10] Chen, Tianqi and Guestrin, Carlos XGBoost: A Scalable Tree Boosting System Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, 785—794

[12] Honnibal, Matthew and Montani, Ines, spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing, 2017