IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Estimating Amazon Product Ratings Based On Customer Reviews Using NLP

[1] Shaik Thaseen Taj, [2] A. Mary Sowjanya

[1]M.tech [2] Assistant professor

[1]Department of IT&CA , Andhra University College of Engineering(A)

[2]Department of CS&SE , Andhra University College of Engineering(A)

Visakhapatnam, AP, India

ABSTRACT:

As online shopping becomes increasingly more popular, many shopping web sites encourage existing customers to add reviews of products purchased. These reviews make an impact on the purchasing decisions of potential customers. At Amazon.com for instance, some products receive hundreds of reviews. It is overwhelming and time restrictive for most customers to read, com-pretend and make decisions based on all of these re-views. Customer's most likely end up reading only a small fraction of the reviews usually in the order which they are presented on the product page. The fundamental objective of this paper is to give nearly full picture of Classifying Amazon Reviews, its sorts and characterization. It also includes the complex orders of late articles and the outline of the ongoing pattern of research in the Classifying Amazon Reviews and its related territories Hence, our application eases this task by analysing and summarizing all reviews and predicting rankings for reviews .reviews provide objective feedback to a product and are therefore inherently useful for consumers. These ratings are often summarized by a numerical rating, or the number of stars. Of course there is more value in the actual text itself than the quantified stars. And at times, the given rating does not truly convey the experience of the product – the heart of the feedback is actually in the text itself. The goal therefore is to build a classifier that would understand the essence of a piece of review and assign it the most appropriate rating based on the meaning of the text, which will help the user decide what other buyers have experienced on buying this product. We carry out this process by a number of modules that include feature extraction and opinion extraction which improves the process of analysis and helps in the formation of an efficient summary.

KEYWORDS: Classifying Amazon Reviews, NLP, Word Embedding Topic Modeling, Machine learning,

I. INTRODUCTION

The rapid growth in volume of product reviews for online shopping web sites drives us to analyze and mine the detain these reviews to help potential customers make informed purchase decisions. It is almost impossible for a customer to read all reviews. For instance, there are66SLR camerasand85TVs on Amazon.com. These SLR cameras and TVseach have more than100reviews. Some of the popular models (e.g. "Canon Digital Rebel XSi 12.2 MP Digital SLR Camera with EF-S 18-55mm/3.5-5.6 IS Lens (Black)")have more than700reviews. We observed that the average number of reviews for SLR cameras and TVs is15:24and10:79, respectively. The average review length of products in these two categories is approximately11sentences. One of the challenges in analyzing these reviews is that Reviews contain complicated opinions on the quality of products, quality of customer services related to the sale and seller credibility. In this paper, we propose a computational model to mine data from these reviews that will construct a justified ranking system to help future customers make better-informed decisions. In addition to the opinions about the product's features, reviews often include comments

unrelated to the product itself. Distinguishing the content focus of these sentences is an important component in the analysis of the reviews.

Fig 1 Example amazon review

The first sentence shows that the camera is recommended based on previous reviews. The second sentence expresses a positive opinion on the camera's quality. The sentence underscored in blue shows negative opinion on the lens. The review also complains about the customer service of the seller. Although there view title expresses a strong negative opinion to the customer support of the seller, a potential buyer of the product cannot make a conclusion about the quality of the product from this review. Sentences/comments unrelated to product quality, such as customer service should be filtered out when measuring the product quality, otherwise it leads to a biased ranking system. The review for a product on the Amazon.in website consists of an overall rating of the product which is obtained from a statistics of each individual's customer rating out of 5 stars and a customer review section where customers drop in their experience on buying that particular product. While the overall rating gives a vague idea of the product's genuineness, the customer review section gives a potentially elaborated idea. Although reading through the customer reviews gives a comprehensible picture, it might be very time consuming in some cases where the product has thousands of reviews listed. We propose to analyze and summarize these customer reviews which are unstructured using sophisticated NLP toolkits. we used natural language processing approach. We propose a dynamic system for feature based comment summarization based on the corresponding domains of products. In this process, the reviews are extracted by web crawling. And preprocessing like lemmatization , extracting the root word, removing accents, removing punctuations, converting to lower case, removing stop words , removing extra spaces are performed These compound sentences are broken down into individual sentences and further into words by sentence-tokenization and word-tokenization respectively. Now, identification and extraction of the features of a product is done first.. Once this is done, phrase modeling (bigrams ,trigrams)will be done. then we will apply bag of words model and TF-IDF model applied to our data . the after that NLP techniques word embedding (word2vec),t-sne, topic modelling are performed on the data. Then random forest ,cross validation, XG boost are used.

2. RELATED WORK

In [1] Fang and Zhan(2015)has done the sentimental analysis on product review data which uses a Naive Bayes classifier for extracting subjective content and tackling polarity categorization problem. It concentrates on both sentence-level and document level. A general process is proposed with adequate process descriptions to categories sentiment polarity.

In paper [2] by Patil and Mane(2016), shows identifying ranked aspects as per their importance. Free reviews are parsed using NLP which identifies the aspects of the particular product. For classifying sentiments supervised classifier SVM is used, then probabilistic.

In [3]Goyal and Parulaker (2015)used movie review on the basis of text using random forest classifier by counting the number of times the word repeated.

In [4] Rahul Wadbude and his team (Wadbude et al., 2016) used Fine-grained sentiment analysis of text reviews has recently gained much attention to the natural language processing community.

3 .METHODOLOGY

Datasets

The Amazon dataset contains the customer reviews for all listed Electronics products spanning from May 1996 up to July 2014. There are a total of 1,689,188 reviews by a total of 192,403 customers on 63,001 unique products. The data dictionary is as follows:

- **asin**- Unique ID of the product being reviewed, string
- **helpful**- A list with two elements: the number of users that voted helpful, and the total number of users that voted on the review (including the not helpfulvotes), list
- overall- The reviewer's rating of the product, int64
- **reviewText**- The review text itself, string
- **reviewerID**- Unique ID of the reviewer, string
- **reviewerName**- Specified name of the reviewer, string
- **summary**- Headline summary of the review, string
- **unixReviewTime**- Unix Time of when the review was posted, string

Data Wrangling

The df is created from the Amazon dataset. If the file has been downloaded then the dataset is loaded from the local file. Otherwise the file is accessed and extracted directly from the repository.

NLP Pre-Processing

In preprocessing we extract the root word therefore it is important to reduce words to their root form.. We'll be using Lemmatization to reduce tokens to their base word. WordNetLemmatizer used from the Natural Language Toolkit (or NLTK). Lemmatization only applies to each word but it is dependent on sentence structure to understand context. therefore part-of-speech tags associated with each word. accent, punctuations, stop words, extra spaces will be removed,and the words are converted into to lowercase, Then Tokenization will be done..after that Phrase Modelling will done. The higher the threshold, the more often two words must appear adjacent to be grouped into a phrase .Bigrams are generated from using the gensim phraser. Only those that pass the bi_gram criteria are considered. Trigrams are generated by applying another gensim phraser on top of a bigram phrase.

TF-IDF Model

The Term Frequency-Inverse Document Frequency (TF-IDF) approach assigns continuous values instead of simple integers for the token frequency. Words that appear frequently overall tend to not establish saliency in a document, and are thus weighted lower. Words that are unique to some documents tend to help distinguish it from the rest and are thus weighted higher. The tfidf weighting is based on our bow variable.

```
from gensim.models.tfidfmodel import TfidfModel

tfidf = TfidfModel(bow)

for idx, weight in tfidf[bow[0]]:
    print(f"Word: {vocabulary.get(idx)}, Weight: {weight:.3f}")
```

```
Word: address, Weight: 0.113
Word: around, Weight: 0.060
Word: arrive, Weight: 0.093
Word: back, Weight: 0.051
Word: bad, Weight: 0.068
Word: big, Weight: 0.126
Word: come, Weight: 0.046
Word: contact, Weight: 0.103
Word: could, Weight: 0.054
Word: day, Weight: 0.061
Word: earlier, Weight: 0.141
Word: ease, Weight: 0.220
Word: ect, Weight: 0.181
Word: email, Weight: 0.213
Word: exception, Weight: 0.131
Word: exchange, Weight: 0.132
Word: expect, Weight: 0.067
Word: freeze, Weight: 0.259
```

Fig-2 TF-IDF

Word Embedding for Feature Engineering

The downside of count-based techniques is that without regard to word sequence and sentence structure, the semantics get lost. The Word2Vec technique, on the other hand, actually embeds meaning in vectors by quantifying how often a word appears within the vicinity of a given set of other words.

Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique that we can use on our modeled to reduce its 100 dimensions to just two dimensions.

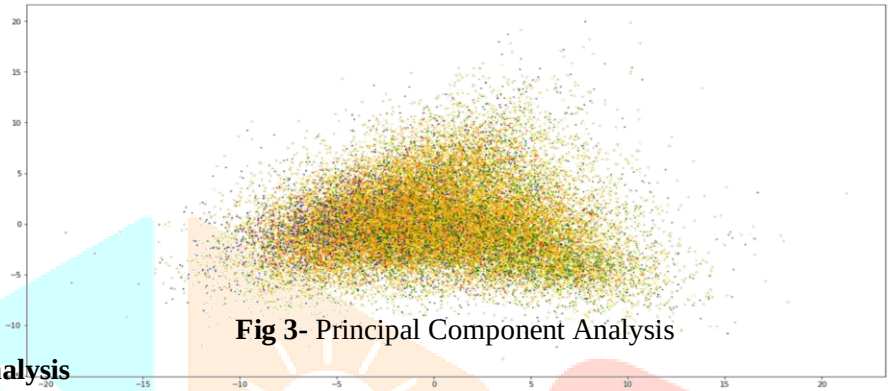

Fig 3- Principal Component Analysis

Exploratory Data Analysis

In this paper several interesting Natural Language Processing techniques are implemented in order to explore our Amazon dataset.

t-SNE

Like PCA, the t-Distributed Stochastic Neighbor embedding (t-SNE) is another dimensionality reduction technique to assist in visualizing high-dimensional datasets. To perceive the similarity between the related words in terms of spatial distance, t-SNE provided the coordinates of each word in a 2D scatterplot plane.

```
tsne = TSNE(n_components=2, perplexity=5, n_iter=1000, random_state=42)

sample_vecs = word_vec.wv[set(word_bank)]
sample_tsne = tsne.fit_transform(sample_vecs)
tsne_x = sample_tsne[:, 0]
tsne_y = sample_tsne[:, 1]

f, axes = plt.subplots(figsize=(20,7))
ax = plt.scatter(x=tsne_x, y=tsne_y)

for label, x, y in zip(word_bank, tsne_x, tsne_y):
    plt.annotate(label, xy=(x+3, y+3))

plt.show()
```

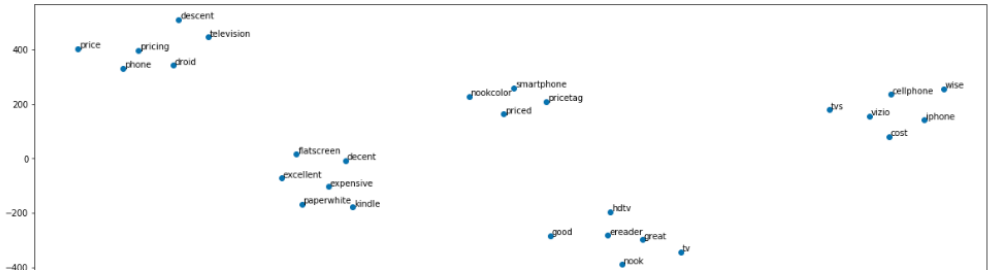

Fig-4 t-sne

Word Algebra

Since Word2Vec characterizes words into quantified tokens, we can consequently add or subtract word vectors together. To add is to combine the meaning of the components and to subtract is to take out the context of one token from another.

Named Entity Recognition

Now we use spaCy to go further and identify what nouns in the documents refer to. So me Named-Entity Recognition (NER) classification tags include distinguishing persons, organizations, products, places, dates, etc.

Dependency Tree

The capability of spaCy's NER is based on deciphering the structure of the sentence by breaking down how tokens interact with and influence each other. Below is the dependency trees of the first two sentences of the most_helpful_text.

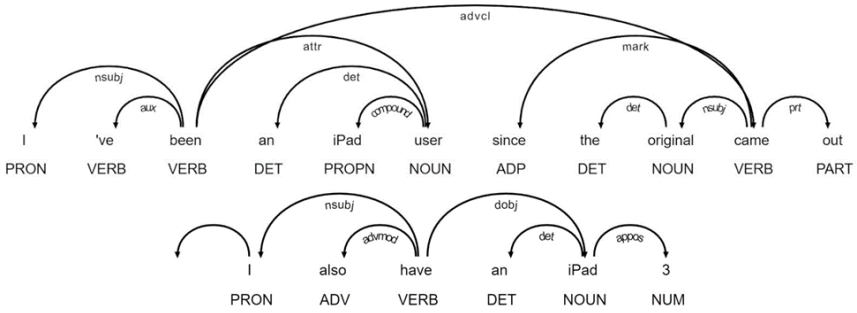

Fig-5 Dependency Tree

Topic Modelling

Because Latent Dirichlet Allocation (LDA) can cluster documents together according to topic, the reviews can be classified and grouped according to the type of electronics product they correspond to. The product reviews will have weights assigned to each of the topic and the topics themselves will have weights on every token. As it is a clustering-based model, LDA is unsupervised and only the num_ topics is configurable.

XG BOOST

Boosting models outperformed Logistic Regression and Random Forest approaches using default parameters ● Tuned, multi-class XG Boost model was ultimately used for the study

Test Set Accuracy: 65.161% Test Set F1 Score: 0.652 Balanced Test Set Accuracy: 53.336% Balanced Test Set F1 Score: 0.533

FIG -6 Confusion Matrix For Test

4. RESULT

Fig-7 : 1 STAR RATING CLOUD

Fig-8 2 STAR RATING

Fig-9 3 STAR RATING

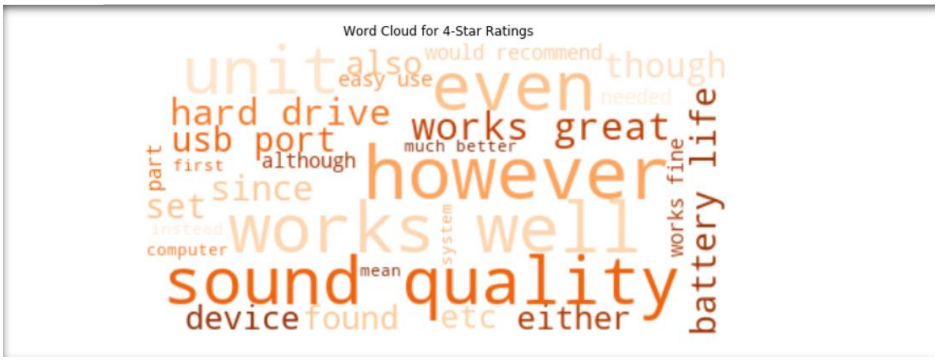

Fig -10 4 STAR RATING

Fig-11 5 STAR RATING

5 .CONCLUSION

A lot of Natural Language Processing techniques were covered in the study. Just some of the concepts explored include topic modelling – where similar texts were clustered together according to topic, named entity recognition (NER) – where nouns were given identifying labels like place or time, and dependency trees – where parts-of-speech tags and sentence structure were discerned. Though the Word2Vec phase was central to our final model, the pre-processing steps were perhaps just as crucial. Prior to tokenization, each document had to be decoded from UTF and encoded to ASCII, and converted to lowercase. The texts were stripped of accents, stop words and punctuation, and multiple whitespaces were dropped. Words were simplified to their root words in order to compact the vocabulary as much as possible. Tokens that were often used together were also singularized through phrase modelling.Beyond word use and word frequency, our model actually extracts and quantifies context. Every token in all the reviews are understood by their neighbouring words and embedded in a given number of dimensions. All the interactions of a word with all the other words it has been associated with are expressed in vectors. And all the words in a given review are averaged according to each of the dimensions to create its 100 features. So the essence of a review by its words make up the final data frame.

REFERENCES

[1].Fang and Zhan(2015) https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2

[2]. Patia and Mane(2016), **https://www.ijsr.net/get_abstract.php?paper_id=ART20161718**

[3]. Goyal, A., & Parulekar, A. (2015). Sentiment analysis for movie reviews. Available at https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/fa15/003.pdf.

[4]. Rahul Wadbude, ., Gupta, V., Mekala, D., Jindal, J., & Karnick, H. (2016). User bias removal in fine grained sentiment analysis. In European Chapter of the Association for Computational Linguistics, arXiv:1612.06821v1 [cs.CL] 20 Dec 2016.

[5]Feature Based Summarization of Customers' Reviews of Online Products. By: Kushal Bafna, Durga Toshniwal M. Tech. Computer Science, Electronics and Computer Engineering Dept., IIT Roorkee-247667, India Associate Professor, Electronics and Computer Engineering Dept., IIT Roorkee-247667, India.

[Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050913008831

[6]Feature-Based Customer Review Summarization. By: Alessandro Maisto, Serena Pelosi. [Online]. Available: http://link.springer.com/chapter/10.1007%2F978-3-662-45550-0_30

[7]Product Review Summarization from a Deeper Perspective. By: Duy Khang Ly, Kazunari Sugiyama, Ziheng Lin, Min-Yen Kan. National University of Singapore Computing 1, 13 Computing Drive Singapore 117417.

[Online]. Available: https://www.comp.nus.edu.sg/~sugiyama/papers/p311.pdf

[8]Comprehensive Review of Opinion Summarization HYUN DUK KIM University of Illinois at Urbana-Champaign KAVITA GANESAN University of Illinois at Urbana-Champaign PARIKSHIT SONDHI University of Illinois at Urbana-Champaign and CHENGXIANG ZHAI University of Illinois at Urbana-Champaign. [Online]. Available: https://s3-us-west-2.amazonaws.com/mlsurveys/134.pdf