DOI: 10.21917/ijsc.2017.0193

AN APPROACH FOR REVIEWING AND RANKING THE CUSTOMERS' REVIEWS THROUGH QUALITY OF REVIEW (QoR)

Sumit Kawate[1] and Kailas Patil[2]
Department of Computer Engineering, Vishwakarma Institute of Information Technology, India
E-mail: [1]sumitkawate@gmail.com, [2]kailas.patil@viit.ac.in

Abstract

Quality is referred as the degree of excellence that means the expected product or service being considered to achieve desired requirements. Whereas, Quality of Reviews (QoR) relates to the task of determining the quality, efficiency, suitability, or utility of each review by addressing Quality of Product (QoP) and Quality of Service (QoS). It is an essential task of ranking, the reviews based on the quality and efficiency of the reviews given by the users. Whatever the reviews are provided to the particular product or services are from user experiences. The Quality of Reviews (QoR) is one of a kind method that defines how the customer's standpoint for the service or product that he/she experienced. The main issue while reviewing any product, the reviewer provides his/her opinion in the form of reviews and might be a few of those reviews are malicious spam entries to skew the rating of the product. Also in another case, many times customers provide the reviews which are quite common and that won't helpful for the buyer to interpret the helpful feedback on their products because of too many formal reviews from distinct customers. Hence, we proposed novel approaches: i) to statistical analyzes the customer reviews on products by Amazon to identify top most useful or helpful reviewers; ii) to analyze the products and its reviews associated for malicious reviews ratings that skewed the overall product ranking. As this is one of the efficient approaches to avoid spam reviewers somehow from reviewing the products. With this, we can use this method for distinguishing between nominal users and spammers. This method helps to quest for helpful reviewers not only to make the product better from best quality reviewers, but also these quality reviewers themselves can able to review future products.

Keywords:

Quality of Review, Opinion Mining, Sentiment Analysis, Quality of Product, Quality of Experience

1. INTRODUCTION

An increase in internet network allows supporting to the users to express their opinions on products for public evaluation to get in focus to understand them. There is also increasing in an E-business website for shopping and services for customers [10]. QoS and QoE are different to each other. They mean at various sources as they come from it. QoE is a reflection of the experiences of the user with feedback (review) while provider of service directly provides QoS. They are highly co-related to each other [1].

The Quality of Reviews (QoR) refers to the task of determining the quality, efficiency, suitability, or utility of each review by addressing Quality of Product (QoP) and Quality of Service (QoS). It is an essential task of ranking, the reviews based on the quality and efficiency of the reviews that is provided by the users of the products [6]. Whatever the users express their point of view on the product is mean to positive or negative opinion. Consider one scenario where Nibsy Garcia purchased a Samsung mobile from Amazon.com and after the purchase she gave her review of her product as "this product is not original, you send me a cell phone with parts made in china, i bought a Samsung with original parts, and you send me a Chinese version". Well, as above review from Nibsy, we can say that she wasn't happy with the product she purchased and experienced from Amazon.

If any reviewer provides a review on the product and many people agree with the review because of his/her review helps people for their purchasing decisions, then their review will get a 1+ helpful rate from people satisfied from his/her review [15]. Just like, "like" on status or photos on Facebook, "love" on tweets on the Twitter.

Fig.1. An Example of Customer's Review on Product of Amazon

The Fig.1 shows a case of customer review on the product of Amazon. As D. Stringer (reviewer) rates the product 5/5. And According to their review on the product, 27 people (customers/visitors) found it as helpful for them out of 29 people. So this reviewer somehow stands well regarding reviewer quality [12], [14].

Fig.2. Example rating inconsistency for a sample product

For instance, see Fig.2 is a screenshot of reviewers' reviews on a sample product. The review from reviewers Leslie S., Charlie B. Huntley and Sarenity 3 seems to be fine. As they reviewed the product based on what they experienced and thus they even rate it. But if we consider, reviews of reviewers Thomas Magnum and Ana M. are the inconsistent on what they reviewed on the same product. They reviewed 3 out of 5 and this express neutral to the product. But the overall rating of the product seems to be positive and also they express positive experience towards the reviews. Their reviews could be malicious or inconsistent. And thus removing this review will give us our objective. This is surely going to help the product ratings to find out inconsistencies.

Finding out QoR is devised as a regression problem. As for review ranking, the model allocated a particular score to each review and based on that "review recommendation" is possible. The score allocates to each review helps for ranking reviewers [13].

So, we target one of the biggest vendors in the shopping category named as "Amazon" for our analysis and research [3]. There were few reasons to choose Amazon for our research. We wanted to do an analysis on big and precise data. Here, big data means the number of products to be more. As, more product availability tends to more and more reviews provided by reviewers.

Usually, sellers sell their product to earn money and buyers wish to purchase the product that they desired concerning the good quality and lesser price. Our focus on the set of reviewers where they used to give reviews on the product either they wish to purchase it or they have already purchased and provided the feedback on the product. Whatever, the feedback is regarding favorable or unfavorable to the Quality of Product (QoP) [1], [4].

The proposed model helps in reviewing and ranking the reviewers' review to help to avoid the ratings of spam reviewers. The approach we are providing almost solve the trustworthy reviewers to rate the particular product. The choice of reviewers for the product recommendation is to be done by the reviewer score [13]. The reviewer score depends on not only the total number of helpfulness counts, but also the total number of unhelpfulness count. As unhelpfulness play a crucial role for rating the reviewers.

Consider the sample review data:

{"reviewerID": "A012668725TCXOBEMGHBA", "asin": "B00DPV1RSA", "reviewerName": "Saralynn", "helpful": [21, 29], "reviewText": "It was exactly what I was looking for. It is nice quality and soft. For being one big clip, it is perfect for a layer of color that isn't permanent (: I highly recommend this product. Plus it's not extremely expensive so it is well worth it! You won't be disappointed (:", "overall": 5.0, "summary": "Perfect", "unixReviewTime": 1376611200, "reviewTime": "08 16, 2013"}

where,

reviewerID - ID of the reviewer,

e.g. A012668725TCXOBEMGHBA

asin - ID of the product, e.g. B00DPV1RSA

reviewerName - name of the reviewer

helpful - helpfulness rating of the review, e.g. 21/29

reviewText - text of the review

overall - rating of the product

summary - summary of the review

unixReviewTime - time of the review (unix time)

reviewTime - time of the review (raw)

In above sample review data, Saralynn purchased a product and according to her review, she wanted to recommend this product to other people as she was highly satisfied with this quality product. But if you look at the above data, even though she rated the product 5 out of 5, we can't judge the reviewer's quality from her ratings to the product. Well, the number of user rates on the distinct products based on their experiences, but few of them are informative-user, or we can say, useful user [5]. But we are interested in how many other people/customers on Amazon that agreed and satisfied with the review given by the reviewer.

Apparently to say, if the reviews follow "Creditability", "Informativeness", "Readability" quality attributes, then the customers found this review a "Trustworthy" for them and they mark the review as helpful [2]. And this enhances in reviewer's quality. In above sample review data, out of 29 people, 21 people found saralynn's review helpful either for their purchasing decisions or their recommendations.

Thus, there might be a case is that many Amazon reviews are dishonest spam entries that were written to skew the product rating. In another case, there are too many reviews from distinct reviewers on distinct product from the distinct product category and most of their reviews are un-necessary or simply to say un-useful. So, we decided to build one model to evaluate users' opinions on the product by analyzing and reviewing the reviews of the reviewers to derive conclusions that make the product better for further customers to take purchasing decisions.

2. LITERATURE SURVEY

Bipin Upadhyaya, et al. [1] proposed the work for a service using identification and aggregation of Quality attributes by showing recall and precision. As they have shown most of QoE attributes for selection of services to be used if QoS attributes aren't available.

Luisa Mich [4] proposed the work on how to eliminate or mitigate the existing quality gaps using meta-model known as 7Loci. For evaluation purpose, characteristics or factors have to evaluate so that to mitigate existing quality gaps.

Noisian Koh, et al. [5] proposed an approach for investigating ratings of online movies like douban.com and imdb.com; to assess the behavior of writing of movies reviews.

Saurabh S. Wani and Y.N. Patil [9] proposed faster and efficient approach to data retrieval for text analysis, which is based on opinion mining of popular social networking websites. They come up with an idea for specifying the significance of IDF (Inverse Document Frequency) and TF (Term Frequency) in scoring formulae of Lucene.

Walisa Romsaiyud and Wichian Premchaiswadi [7] proposed an approach for social tagging, integration along with collaborative web search in order to improve satisfaction for users or group of users in web search results.

Jo Mackiewicz and Dave Yeats [2] proposed a method of the implications on research and theory using creditability, Informativeness, readability. Based on 829 responses, they performed an analysis on responses provided by users by considering the user's perception of review quality.

S. Wang, et al. [20] proposed Cumulative Method, Pearson Co-relation Coefficient, bloom filters in order to reduce the reputation measurement. With this idea they have enhanced their results using the success ratio in recommendation of web services. As they need to investigate feedback ratings for prevention schemes.

S.P. Algur and J.G. Biradar, [21] proposed natural language techniques and toolkit counting method which is quite effective in assessing whether the review is spam or not. It requires human evaluation methods and also they need to provide efficient work on sentiment analysis.

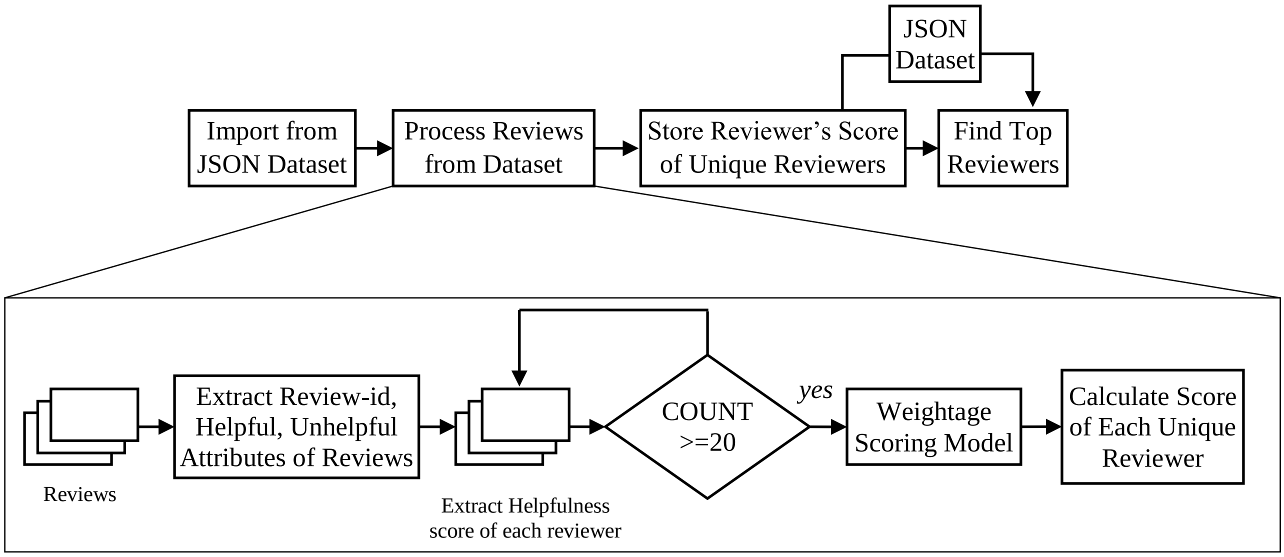

Fig.3. Architecture of the System (based on 1st model)

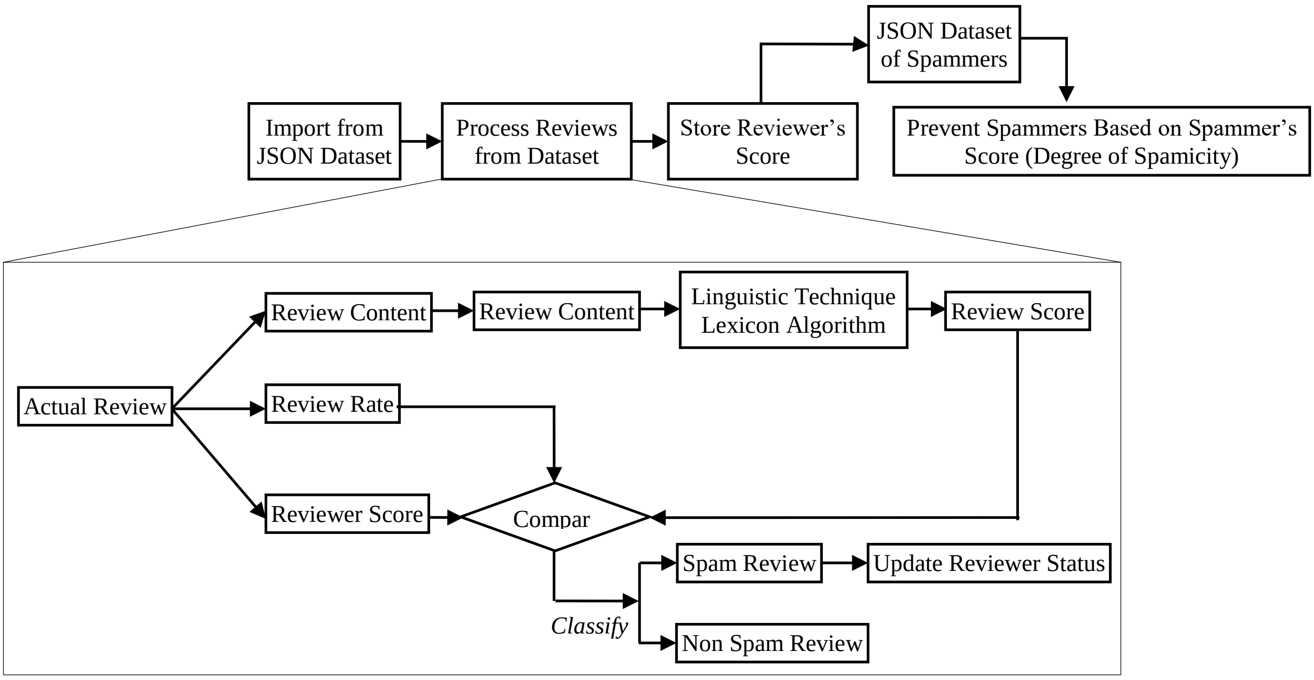

Fig.4. Architecture of the System (based on 2nd model)

Nitin Jindal and Bing Liu [25] proposed an approach that categorizes spam reviews and also they more focused on duplicate detections as near duplicates. They have used feature extraction and also logistic regression model because of highly effectiveness.

Vishal Meshram et al. [16] proposed a survey paper that summarizes ubiquitous computing properties and provides research directions required into Quality of Service assurance.

Another line of research [17-19, 29-33] proposed secured systems to analyze web contents. They use compartment level isolations, policy enforcement and filtering in web browser to prevent content injection attacks.

3. ARCHITECTURE

In Fig.3 shows Architecture of the System. As we took input as a JSON file of Amazon dataset, which is of 54.3 GB. Then we processed the reviews from the dataset by parsing the JSON file and then extracted reviewer id, helpful and unhelpful attributes. Based on reviewer id, we have extracted helpfulness score of each reviewer with the count of the total number of reviewers had given reviews on the products. We sent only those reviewer's reviews to weightage scoring model if the number of reviews per reviewer (count) is greater or equal to 20 reviews. This is required for calculating the score of each unique reviewer. Then store reviewer's score of unique reviewers in JSON file for further analysis. And then select top reviewers as per requirement based on reviewer score. And even these can be useful for selecting top reviewers in each category.

The Fig.4 shows the architecture of the proposed system, where the set of reviews is stored in review database file. From that we extract the textual reviews, field in order to classify them and in many cases, where the reviews of the reviewers having good or bad ratings. As these ratings are given through the helpfulness mechanism to the customer review, but many of time the reviewer may review to the product that contain spam contents. Here spam contents refer to the loss of quality where reviewer intended to disturb the quality of product. As shown in our architecture, we first extract the data from dataset in json file format and then consider overall rating given by the reviewer to product. And then our job is to find out whether that review is spam or not by using lexicon mechanism. We calculate the score of that review and then adjust that score in a Likert scale with the reviewer rate to the product. Based on this score, our approach will find whether that review is spam or not based on rating consistency.

4. STATISTICAL ANALYSIS OF REVIEWS OF AMAZON

As the number of product and services offered by Amazon has increased. [3], [11], [14] And also the customer's attentions towards Amazon has also increased due to their need. Reviews of customers who have purchased and experienced the product provide more attention on the product itself. Customers provide reviews by rating the product that they experienced based on the product quality. The customer rates the product from 1 to 5-star rating. But many times they offered to write reviews on the product they want to purchase or they wish to purchase, but sometimes it is not a much more favorable option to judge the reviewer's quality by reviewer's rating score out of 5. We are considering Amazon as for reviewing the reviews of the product so to achieve the top reviewers in each product category.

4.1 DATA COLLECTION

We have worked on the Amazon reviews dataset and we obtained it by contacting Julian McAuley [3]. The dataset we are analyzed is in the JSON file format, and it is of size 54.3 GB. We found the following statistical result:

Number of Reviews = 82.6 Million

Number of unique reviewers = 21.1 Million

Number of unique products = 9.8 Million

Average number of reviews per reviewer = 3.904188

Average number of reviews per product = 8.372951

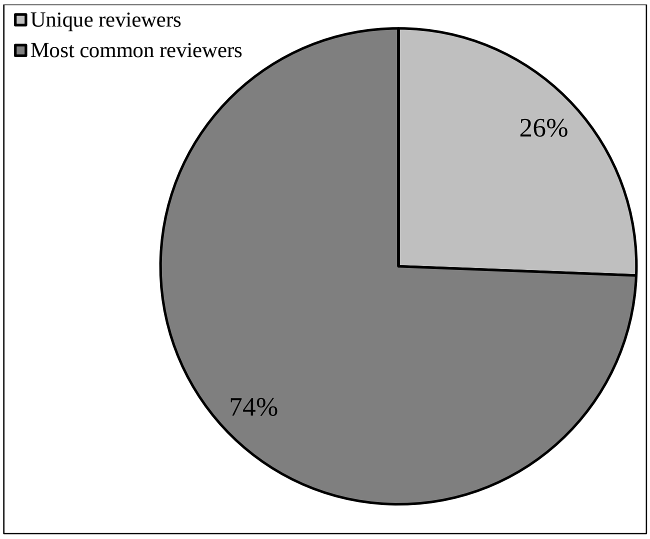

Fig.5. Ratio of unique reviewers to all common reviewers in Amazon

The Fig.5 clearly states that the number of reviews per reviewer. The reason is for greater reviews per reviewer as there are many more reviewers; those gave reviews more than one time. Due to this, the common reviews are more significant than that of unique reviewers.

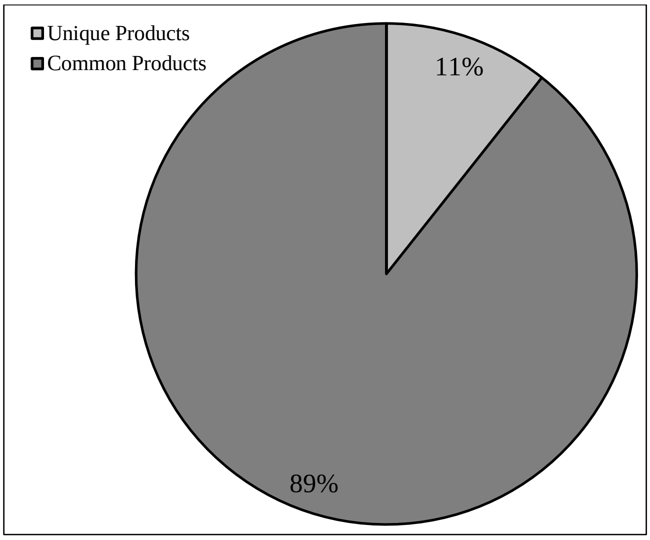

Fig.6. Ratio of unique products to all common products in Amazon

In Fig.6 states that the number of reviews for the product that are 11% more and there are cases that the reviewers provided more than one reviews to products.

If we look at both Fig.5 and Fig.6, we can come to know that reviewers and products are available, but the rate of reviews for products from reviewers is more. Yes, here we come up that if the number of such reviews is greater in availability that this may tend to increase in malicious spam reviewers involved in the reviewers for rating the products.

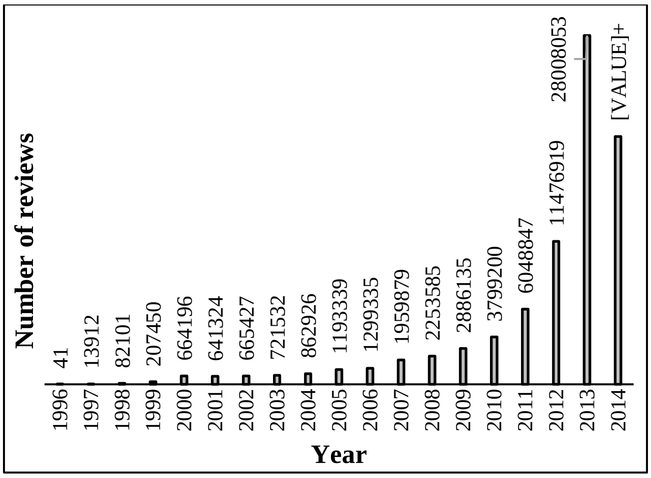

Fig.7. Total number of Amazon reviews per year

As above, Fig.7 shows that the per-year analysis of the number of reviews provided by unique reviewers. We have collected the data from May 1996 to June 2014. If you look at above Fig.7, you came to know that there is a curve formed from the year 1996 to 2013 (Note: As in 2014, we have reviews up to the month of June). The curve states that the number of reviews from each year increased drastically. It means each year; the numbers of reviewers are involved. And the reason behind it, they experienced high availability of productive growth in shopping market.

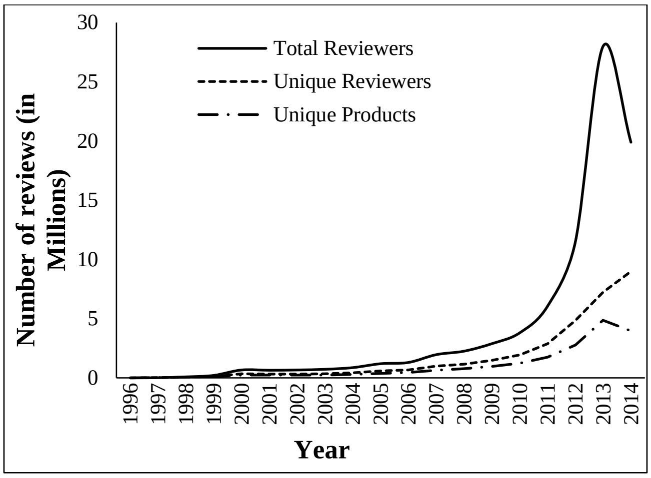

Fig.8. Rate of reviews on Amazon products by unique reviewer per year

You can see a better analysis of this in shown in Fig.8, our focus is on the reviewer's review and we must consider the product category to select those reviewers those had given most possible reviews for different products in the category.

4.2 DATA PROCESSING

For analyzing and ranking the reviewers we need to focus on how many people found the review by are viewer as "helpful". [14] As how many people found the review by are viewer as "unhelpful"? The choice of unhelpful is also important because there might be some case where the review for the product is ineffective, unfavorable, not genuine, false or not helpful at all. This attribute helps to rate reviewer for ranking purpose. It is clear that the reviewer may provide some bogus reviews that is not helpful for the product seekers (customers). And for this, we are selecting only those reviewers, who have at least the minimum number of reviews matched with our threshold in the account.

Usually, people judge the product by specifications, appearance, services, and genuineness, but after all these many customers carefully watch the reviews given by other reviewers and if they agreed or satisfied with it, then customers mark that review as helpful or unhelpful. We are focusing on this helpfulness and unhelpfulness attributes to identify the reviewer score.

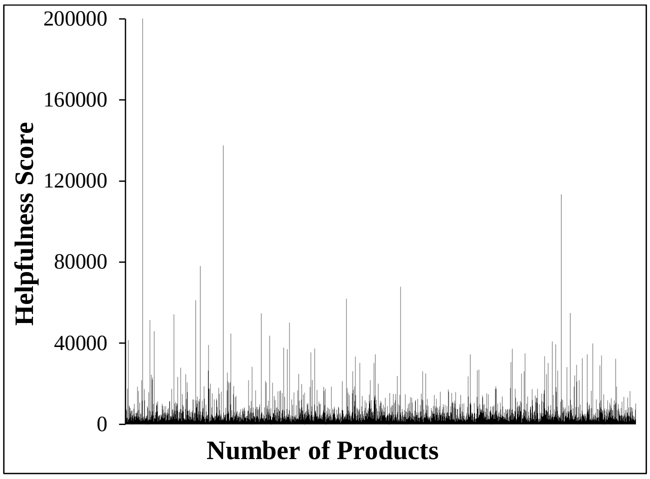

Fig.9. Overall Helpfulness score of all reviewers

The Fig.9 shows the overall helpfulness score of the individual reviewers. We then considered those reviewers that they reviewed the product more than or equal to 20 times. Indirectly, it means we consider only those reviewers that they have at least 20 reviews for the products in the dataset.

Well, the case like this is if reviewer got the number of helpfulness score as high as compared to other reviewers, then it doesn't mean that he is a trustworthy reviewer. As already discussed we even need to focus on the unhelpfulness count by other reviewers for the reviewer's review.

In the above example, Saralynn gave the review on the product and 21 people found that her review is helpful for their purchasing decisions or their expectations/requirements or the reality talk about it. But even we consider we need not forget those 8 peoples, and they found her review unhelpful at all.

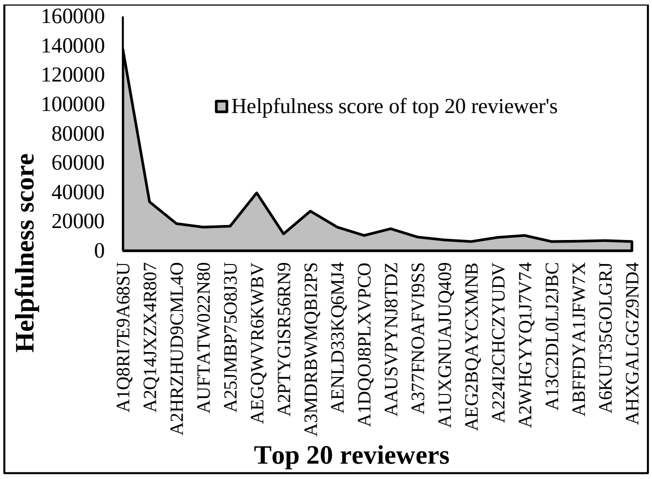

Fig.10. Overall helpfulness score of top 20 reviewers

Consider the above Fig.10 that illustrates the statistical analysis of the top 20 reviewers. We calculate the helpfulness score and upon it, we received the above reviewers (reviewer id on their X-Axis) have a more helpfulness rating (score) than another. Both helpfulness and unhelpfulness count helps to build a model for calculating the reviewer's score.

As reviewer id: "A14OJS0VWMOSWO" gave 44557 reviews on the product and had helpful rate of 207789 which is largest in our dataset. But this reviewer had found helpful by 207789 people out of 249904.

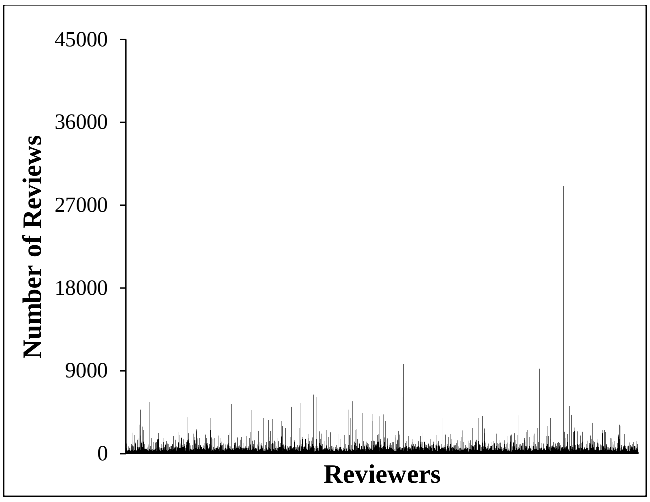

Fig.11. Total number of reviews per reviewer

We considered those reviewers who reviewed the products and they provided at least 20 reviews from their side. The reason behind selecting 20 reviews is, we considered a threshold value 20 for reviews so that minimum 20 reviews from the reviewer should be considered. It is not possible to judge the trustworthiness of reviewers through one or fewer reviews. We need some set of reviews for the reviews to judge them, to rate them and to rank them.

In Fig.11 reviewers who, having their id: A14OJS0VWMOSWO, AFVQZQ8PW0L, A328S9RN3U5M68, and A9Q28YTLYREO7 gave the maximum number of reviews to different products.

4.2.1 Weightage Scoring Model:

To rank reviewers, we need to assign a weightage score for those unique reviewers whose value for minimum reviews are 20 as we set it as a threshold value. And then, we considered the total number of helpful ratings for a reviewer and also average helpfulness and unhelpfulness score. The formula for calculating the weightage score of a reviewer is as in Eq.(1):

$$RWH = ((c/c + 20) \times avgH) - ((c/c + 20) \times avgUH) \quad (1)$$

where,

RWH = Reviewer's Weightage score

c = total count of the number of helpful rating for a reviewer

$avgH$ = average helpfulness of a reviewer

$avgUH$ = average unhelpfulness of a reviewer

20 = minimum reviews required (threshold value)

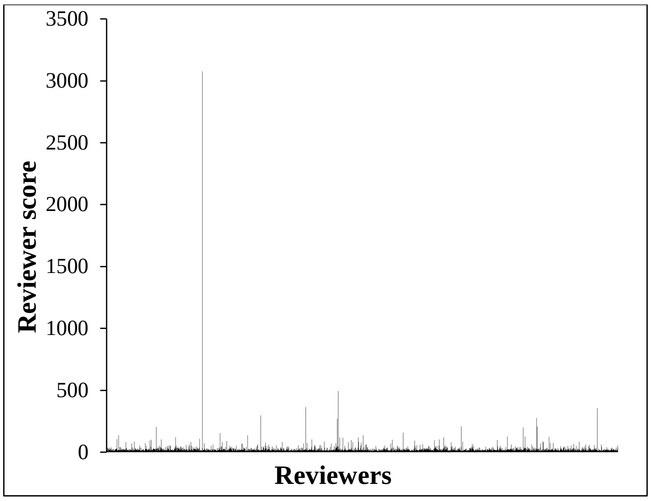

Fig.12. Reviewer score of all reviewer

From our scoring model, we calculate the weightage score of those reviewers whose minimum reviews are 20. And from these, we plot the graph as shown in Fig.12. From weightage scoring model, we have retrieved total 582080 reviewers along with their scores, helpfulness and unhelpfulness count.

For Example, Reviewer ID: ANYEL2T9NDED7, Total number of reviews: 35, Overall rate: 202, Helpful rate: 166, Unhelpful rate: 36.

So from our weightage scoring model, we found that Reviewer's weightage score (RWH of ANYEL2T9NDED7) = 2.363636.

There is a case that the reviewer score may be negative if the unhelpful rate is more than helpful rate.

Consider the reviewers, those having reviewer id like A1Q8RI7E9A68SU, A2Q14JXZX4R807, A2HRZHUD9CML4O and AUFTATW022N80 have a maximum reviewer score. As, maximum reviewer score 3076.227 scored by reviewer id A1Q8RI7E9A68SU while minimum reviewer score -82.0952 scored by reviewer id A33YHYE1QJ47K8.

It means many people found their reviews as helpful. Might be there to be a case where the above reviewers would have given the bad opinion on the specific product and people satisfy with the review given by the reviewer and found their review as helpful.

As, reviewers provide reviews or comments on products along with rating associated.

4.2.2 Malicious Reviews Rating Prevention (MRRP) Model:

Another model for QoR is Malicious Review Rating Prevention Model based on Rating Consistency Check [26]. As, we ranked the reviewers based on helpfulness criteria by considering trustworthiness property [23], [25]. But, we found that, the quality is not in control because of existing malicious ratings that damaging overall product reputation. [20]The main reason is detecting malicious ratings and predicting the helpful reviews are the two distinct issues when we consider Quality of Review [20].

Hence, our motivation is to repair the overall product rating that is skewed from the malicious reviews ratings. This can be achieved using the Rating Consistency check for Malicious Reviews Rating Prevention (MRRP). We propose a novel approach to statistical analyzes the customers' reviews on the products to identify malicious reviews rating in the dataset. Because of such technique will help the customers to find accurate product ratings and also to find best probable reviewers.

4.2.3 Workflow of MRRP Model:

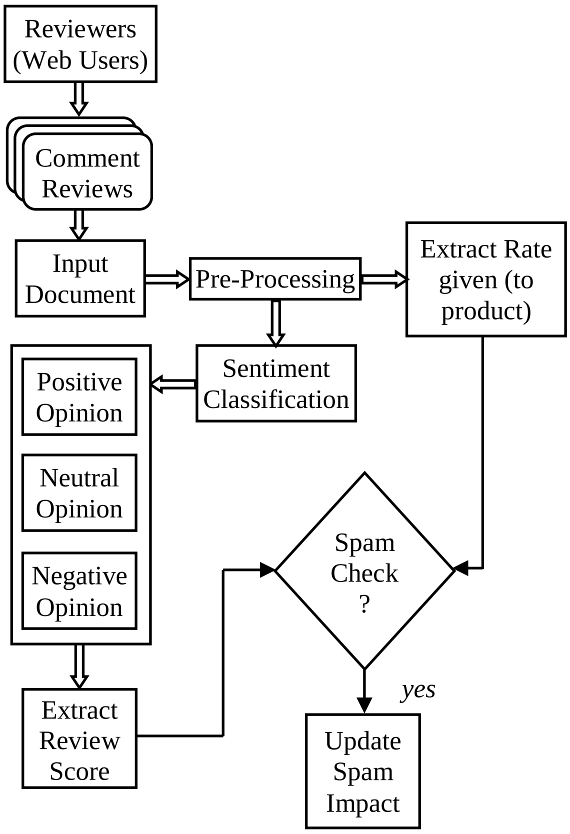

Fig.13. Workflow of MRRP model

The Fig.13 shows the workflow of MRRP model that is based on sentiment analysis. We have extracted and preprocessed reviews. For sentiment classification we are using "sentimentr" package for text polarities. The reason behind choosing "sentimentr" package over "syuzhet", "RSentiment" and "Stanford" package is that, since it balances accuracy and speed. And then we extracted spam score and reflect changes in product rankings [27].

We have even used the SentiWordNet3.0, Stanford Core NLP and Word Counting Method [21].

$$\text{Total Words} = NPW + NNW$$

$$PScore = (TPW) / (\text{Total Words})$$

$$NScore = (TNW) / (\text{Total Words})$$

where,

NPW = Number of Positive Words

NNW = Number of Negative Words

PScore is Positive Score

NScore is Negative Score

Word Counting Discriminative Algorithm

```
if(totalScore >= 0.50)
{sent = "strong_positive"; s_score=5;}
elseif(totalScore>= 0.25 &&totalScore<0.50)
{sent = "positive"; s_score=4;}
else if(totalScore> 0 &&totalScore<0.25)
{sent = "weak_positive"; s_score=3;}
else if(totalScore<= 0 &&totalScore>-0.25)
{sent = "weak_negative"; s_score=3;}
elseif(totalScore<= -0.25 &&totalScore>-0.50)
{sent = "negative"; s_score=2;}
elseif(totalScore<=-0.50)
{sent = "strong_negative"; s_score=1;}
```

We have calculated s_score is the score of the content of the review. Based on this, we can able to draw the sentiment of the review. For this, initially, we used SentiWordNet 3.0 Dictionary to devise the overall positive and negative score based on WordNet. As, it includes POS (Parts of Speech), unique ID, PosScore, NegScore, SynsetTerms and Gloss. As, pair of <POS, ID> uniquely identifies synset where positive score and negative score is assigned to it. We have drawn those scores to calculate the number of positive and negative words. It is kind of word counting algorithm [24].

The problem with this algorithm is to word count itself. As we can define how much the negativeness or positiveness of the word and it is not the case for accurate sentiment analysis. As per our example in Fig.11 the review of Charlie gives strong negative because of a maximum number of negative words so it draws -2.02 and on average it is 1. So this model based on word count is not good enough or accurate as for our sentiment purpose.

4.2.4 Stanford CoreNLP Algorithm:

We have used Stanford CoreNLP algorithm. The problem we faced while designing this algorithm is the packages were called for each time as we pass the reviews from our dataset. So depend on the size of the text of the review, it took an average (1.7+2.3) seconds for every review. The reason we found that we were calling initialize Core NLP packages each time, so we rather did with a separate initializecorenlp() and getsentiment() function and pass the text to preprocess.

Well, for Adding annotator tokenize, TokenizerAnnotator, Adding annotator "ssplit", "Adding_annotator_parse", "ParserGrammar" took 1.7 second for only one time and then we have made the things ready for each and every review. So saved 1.7 seconds for each review and made algorithm efficient

We have used edu.stanford.nlp library to draw the sentiment score.

```
String[] Sent_Text = {"Very Negative","Negative", "Neutral", "Positive", "Very Positive"};
intSent_Score = RNNCoreAnnotations.getPredictedClass(tree);
Sent_Text[Sent_Score];
```

As, from above function, we found sentiment scoreSent_Score asper Likert scale and used as indexing in the declared inSent_Textstring to find actual sentiment.

4.2.5 MRRP Algorithm:

Algorithm1: //to get sentiment score

```
floatgetSentiment(String text)
{
```

```
Process p = Runtime.getRuntime().exec("java -jar \"F:\\sentimental_analysis.jar\" \""+review+"\"");
synchronized (p);
/*we have drawn review sentiment score by using discriminative rules.*/
return score;
}
```

Algorithm2: //for malicious review check

```
Double rate = user.getOverall(); //actual rating given by reviewer
Double sentscore= getSentiment(review); //rate score calculated
if(((rate==5.0 || rate==4.0) &&type.equals("Positive")) || ((rate==1.0 || rate == 2.0) &&type.equals("Negative")) || (rate == 3.0 &&type.equals("Neutral")))
{
spam="No";
}
else if((rate==3.0) && (-0.25<=sentscore&&sentscore<=0.25))
{
spam="No";
 }
else
{
spamrate+=rate;
spam="Yes";  spamcount++;
}
```

Algorithm 3: //update product overall ratings

```
Double orate=0;
for(inti=0;i<=n;i++) // n is no of review for the product
{
nrate +=nrate+rate; //adding total rate
}
Orate=nrate/(n-spamcount); /* calculate overall new rate after removing spam */
```

For sentiment analysis we have used "sentimentr" package, which is developed by trinker. It provides quick calculation of text polarities based on sentence level or even a group of sentences by aggregating. We have used "sentimentr" in the R language. As we have connected R language with Java language. The reason behind to use with R is because the R provides analytics front which Java lacking. So, we have decided to integrate both technologies for high end data analytics [27].

We have used two main packages to integrate R with Java, i.e. Rserve and Rjava. The main purpose of using both the library files in the Java language is easy to use and operating in server mode. Here, the library runs as a server to which client process or program is able to connect and perform the task. For server mode operation, Rserve works on TCP/IP communication by starting an instance of Rserve and the client communicate to Rserve.

Installing Packages on R

```
install.packages("sentimentr")
install.packages("Rserve")
```

Starting Rserve server on R

```
library(Rserve)
Rserve();
```

Creating Java client

```
//import REngine.jar and Rserve.jar in referenced library
```

Java client for Rserve

```
RConnection connection = null;
try {
connection = new RConnection();
connection.eval("source('E:/SentiScript.R')"); //call script
sentscore=connection.eval("myAdd(\""+textual+"\")").asDouble(); //passing arguments to myAdd() function and receiving score
} catch (RserveException e) {
        }
catch (REXPMismatchException e) {
e.printStackTrace();
        }
finally{
connection.close();
        }
}
```

As for malicious review check, we consider the sentiment analysis in order to take ratings from the contents of the review. The reason behind to use sentiment analysis is to check for the rating quality and we have set the Likert scale of 5 to rate the review. We then compare the review rate or score out of 5 with the actual rating given by the reviewer to the product. If their difference is more than 1 rating, then we can say that the review stands as a spam or at least inconsistent [21].

The solution to remove the ratings are spam or inconsistent. This makes the overall product rating to be refined as enhanced by removing malicious review ratings and keeping the genuine product ratings. So, indirectly this helps the seller of the product to get their overall product ratings in purified form as it is refined by our model.

4.3 RESULTS OF OUR STUDY

Our QoR approach is based on two models: Weightage Scoring Model and Malicious Reviews Rating Prevention (MRRP). As per our evaluation, we have seen that Weightage Scoring model helps for customers in order to review the product and this helps to customers (buyers) to make their purchasing decisions based on the enhanced results. The MRRP model helps the seller to preserve the overall quality ratings on the product. We are filtering inconsistent or malicious reviews by not considering their ratings and such consideration helps to overall product ratings.

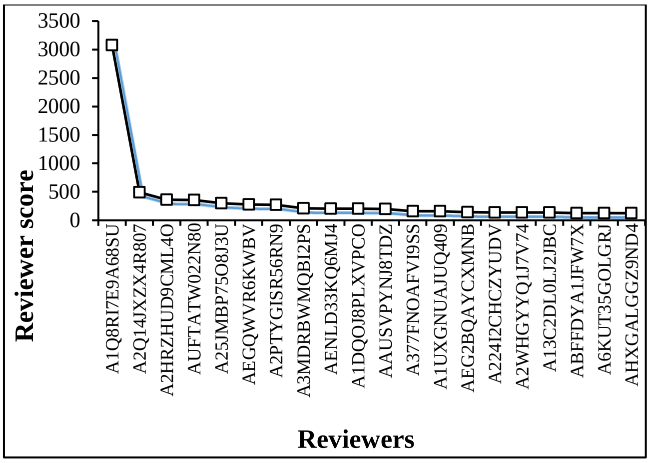

Fig.14. Reviewer's score of top 20 reviewers

From our data collection and processing, we developed a weightage scoring model so that to rate each reviewer based on their count of reviews. Now to rank the reviewers, we considered the weightage score of each reviewer, and we sorted them in descending order intending to get top reviewers from our analysis.

As from analysis, if a reviewer scores good points in helpfulness count due to the maximum number of people considered his/her reviews as helpful, and also having none or very few unhelpful score; then the reviewer will stand in the top ranking category for reviewing the products.

As shown in Fig.14, let's take first (topmost) reviewer having the reviewer id: "A1Q8RI7E9A68SU" has an overall reviewer score of 3076.2273 based on the total 24 reviews. Out of which overall 137585 people found this reviewer as helpful for their purchasing decisions or their personal satisfaction. Similarly, 2231 people found his reviews are unhelpful for them. But overall most of the people found his reviews or himself as helpful [12].

As per our analysis, the top reviewers will help for reviewing more products, and this will help to people for their purchasing decisions. However, the people even recommend the product based on their requirements have been fulfilled by learning the high-quality reviews from the top reviewers for the specified product from each category. It depends on peoples' choices, how they select the reviews as their expectations.

If their expectations, meet with the reviews then it becomes the fair decisions for those peoples. It is quite sure if the product is worst and the reviewer is claiming the same that the product is not good and also they formed the review based on the "informative" and "readability". Then the people will find the reviewer's review as helpful and it is an obvious case that any product can get positive or/and negative set of reviews [12].

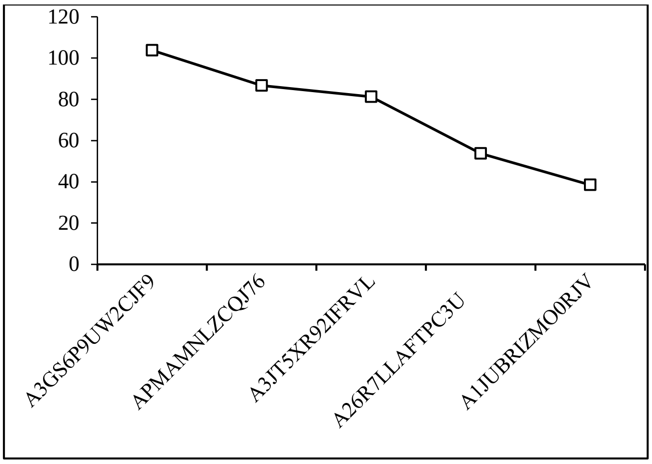

Fig.15. Top Reviewers in Beauty Category

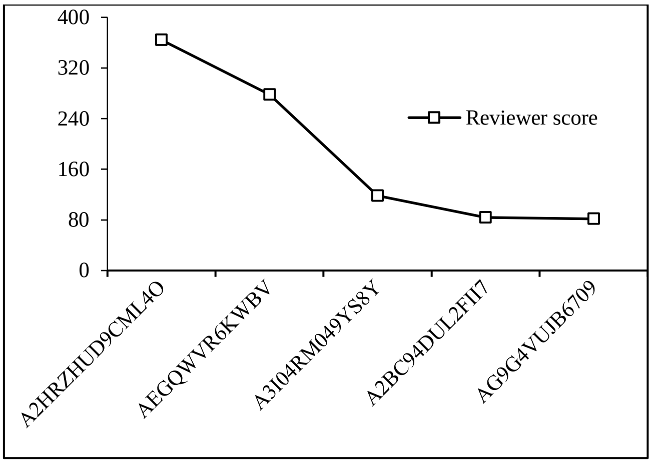

Fig.16. Top Reviewers in Cell Phones and Accessories

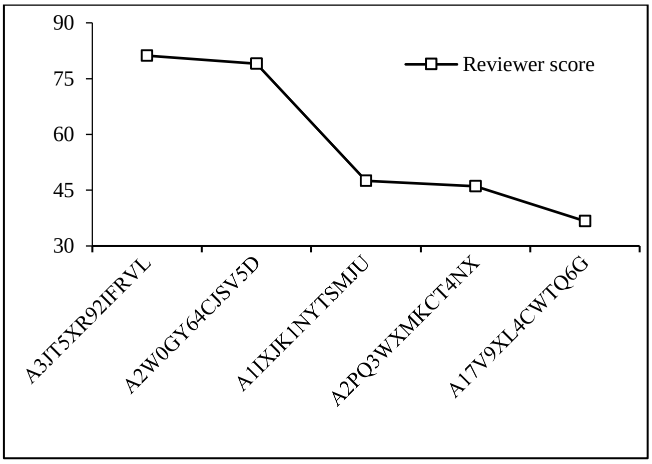

Fig.17. Top Reviewers in Clothing, Shoes and Jewellery Category

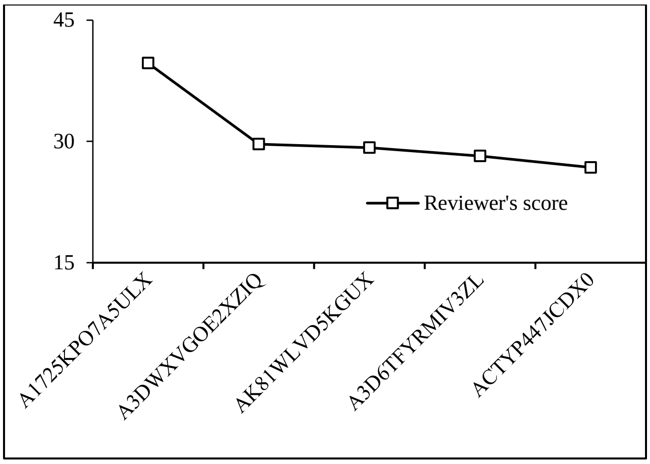

Fig.18. Top Reviewers in Digital Music Category

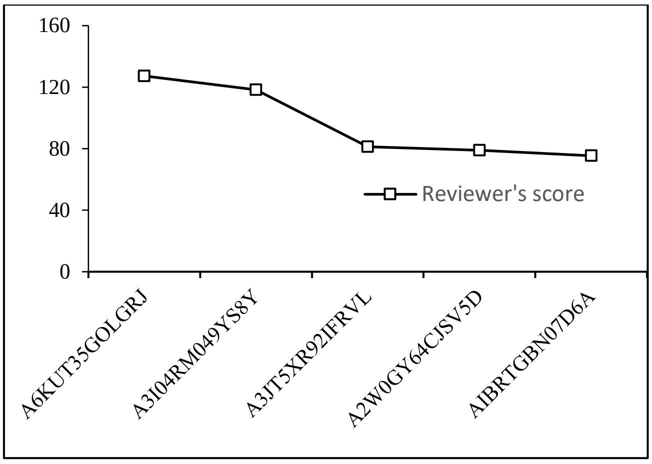

Fig.19. Top Reviewers in Sports and Outdoor Category

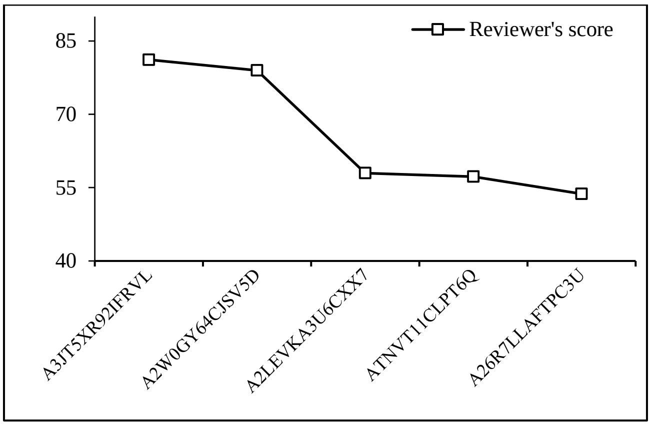

Fig.20. Top Reviewers in Toys and Games Category

From Fig.15 to Fig.20, we have come up with the top reviewers from few, but different categories of products on Amazon. These top reviewers are marked best quality reviewers from our approaches. These help the people or customers for reviewing only top reviewer's reviewer to review the product. We have shown top five reviewers in each category. But according to the product need or even category requirement, there can be able show the quality reviewers. Our proposed approach helps to find out quality reviewers and also to remove malicious spam reviewers. The choice for reviewing the reviewers' review, the QoR stands better rather than some sentiment analysis on the reviews of the reviewers.

We have analyzed the various product categories where the products are well aligned and reviewed by reviewers. As shown in below Table.1, there are many products and reviews from a reviewer on multiple products. As considering MRRP model for malicious ratings, this makes an impact on the products of each category.

Table.1. Number of Products and its reviews from list of product categories

Product Category	Total number of products	Total number of reviews on products
Beauty	12101	198502
Cell Phones and Accessories	10429	194439
Clothing, shoes and Jewellery	23033	278677
Digital Music	3568	64706
Musical Instrumental	900	10261
Sports and Outdoors	18357	296337
Toys and Games	11924	167597

Consider the product id B000MFN8B6 is from Musical Instrumental Category. And there are 10 reviewers who reviewed to this product and the overall product ratings are 4.3 as shown in below Table.2. As for MRRP model, we have used our proposed algorithm to prevent malicious reviews ratings on the product. So we got total 3 malicious review ratings. The overall product rating after filtering spam reviews is 4.7. So we have seen that the product rating changed from 4.3 to 4.7 by filtering malicious review ratings.

Table.2. Impact of malicious review ratings on overall product ratings

Product ID	B003QTM9O2
Number of reviews to the product	5
Overall product ratings before malicious ratings filtering	3.6
Number of malicious or inconsistent reviews	1
Overall product ratings after malicious ratings filtering (new product rating)	4.0

Hence, our MRRP approach helps to get enhanced ratings from skewed ratings and is beneficial to the seller best for getting their overall product ranking.

5. CONCLUSION

Reviewing the reviews through QoR provides a natural way of ranking the top reviewers. Those top reviewers are trustworthy based on the user's perceptions on their opinions. Our analysis on Amazon shows how the QoR focuses on the specialized reviews captured by the quality of user's perception, and this leads to rank them in each of the product's categories. It helps for future customers to assure on our ranked reviews of reviewers. It will work for the new customers to reduce the time to find out the best reviews on the product and as per our method, the reviewers shouldn't have to utilize their time to quest for the best possible reviews on the product. Well, QoR helps new customers (buyer) to get help from our ranked reviewers. QoR even helps to seller to get genuine feedback ratings for the product by removing MRRP model. Hence, our MRRP model balances accuracy and speed for malicious reviews rating prevention.

Interestingly, the customers can be able to quest the best possible products available in the market based on the top ranked (rated) reviewers from our reviewer score. These leads to find out quality reviewers from each categories like sports and outdoor, beauty, digital music, book, entertainment, clothing, shoes and jewellery, toys and games, cell phone and accessories, etc. There are many more products on Amazon or even E-Shopping websites, so many more products are available by the same name or same brand and people want to extract best quality product among them. Hence, the best possible ways we have explained through QoR for reviewing and ranking customers' reviews to help both buyers and sellers.

ACKNOWLEDGEMENTS

This work is in part supported by an AWS Education Research Grant Award, USA.

REFERENCES

[1] Bipin Upadhyaya, Ying Zou, Iman Keivanloo and Joanna Ng, "Quality of Experience: User's Perception about Web Services", *IEEE Transactions on Services Computing*, Vol. 8, No. 3, pp. 410-421, 2015.

[2] Jo Mackiewicz and Dave Yeats, "Product Review Users Perceptions of Review Quality: The Role of Credibility, Informativeness, and Readability", *IEEE Transactions on Professional Communication*, Vol. 57, No. 4, pp. 309-324, 2014.

[3] Julian McAuley and Jure Leskovec, "Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text", *Proceedings of the 7th ACM Conference on Recommender Systems*, pp. 165-172, 2013.

[4] Luisa Mich, "Evaluating Website Quality by Addressing Quality Gaps: A Modular Process", IEEE International Conference on *Software Science, Technology and* Engineering, pp. 42-49, 2014.

[5] Noi Sian Koh, Nan Hu and Eric. K. Clemons, "Do Online Reviews Reflect a Product's True Perceived Quality?-An Investigation of Online Movie Reviews Across Cultures", *Proceedings of 43rd Hawaii International Conference on System Sciences*, pp. 374-385, 2010.

[6] Oren Tsur and Ari Rappoport. "RevRank: A Fully Unsupervised Algorithm for Selecting the Most Helpful Book Reviews", *Proceedings of 3rd International Conference on Weblogs and Social Media*, pp. 154-161, 2009.

[7] Walisa Romsaiyud and Wichian Premchaiswadi, "Exploring Web Search Behavior Patterns to Personalize the Search Results", *Proceedings of 3rd International Conference on Intelligent Networking and Collaborative Systems*, pp. 313-319, 2011.

[8] A. Malibari Almagrabi and J. McNaught. "A Survey of Quality Prediction of Product Reviews", *International Journal of Advanced Computer Science and Applications*, Vol. 6, No. 11, pp. 49-58, 2015.

[9] S.S. Wani and Y.N. Patil, "Analysis of Data Retrieval and Opinion Mining System", *Proceedings of IEEE Conference Advance Computing Conference*, pp. 681-685, 2015.

[10] W. Wang and Y. Zhou, "E-business Websites Evaluation based on Opinion Mining", *Proceedings of International Conference on Electronic Commerce and Business Intelligence*, pp. 87-90, 2009.

[11] Q. Lu and C.G. Zhang, "The Study and Application of E-Business Website based on the Web Data Mining", Advanced Materials Research, Vol. 846-847, pp. 1431-1434, 2013.

[12] Z. Zhang, J. Qi and G. Zhu, "Mining Customer Requirement from Helpful Online Reviews", *Proceedings of Enterprise Systems Conference*, pp. 249-254, 2014.

[13] P.V. Rajeev and V.S. Rekha, "Recommending Products to Customers using Opinion Mining of Online Product Reviews and Features", *International Conference on Circuit, Power and Computing* Technologies, pp. 1-5, 2015.

[14] Y. Zhang and D. Zhang, "Automatically Predicting the Helpfulness of Online Reviews", *Proceedings of IEEE 15th International Conference on Information Reuse and* Integration, pp. 662-668, 2014.

[15] Yuanlin Chen, Yueting Chai, Yi Liu and Yang Xu, "Analysis of Review Helpfulness based on Consumer Perspective", *Tsinghua Science and Technology*, Vol. 20, No. 3, pp. 293-305, 2015.

[16] Vishal Meshram, Vidula Meshram and Kailas Patil, "A Survey on Ubiquitous Computing", *ICTACT Journal on Soft Computing*, Vol. 6, No. 2, pp. 1130-1135, 2016.

[17] Kailas Patil, Xinshu Dong, Xiaolei Li, Zhenkai Liang and Xuxian Jiang, "Towards Fine-Grained Access Control in Javascript Contexts", *Proceedings of 31st International Conference on Distributed Computing Systems*, pp. 720-729, 2011.

[18] Kailas Patil and Braun Frederik, "A Measurement Study of the Content Security Policy on Real-World Applications", *International Journal of Network Security*, Vol. 18, No.2, pp. 383-392, 2016.

[19] Kailas Patil, Tanvi Vyas, Frederik Braun, Mark Goodwin and Zhenkai Liang, "Poster: Usercsp-User Specified Content Security Policies", *Proceedings of Symposium on Usable Privacy and Security*, pp. 1-2, 2013.

[20] S. Wang, Z. Zheng, Z. Wu, M.R. Lyu and F. Yang, "Reputation Measurement and Malicious Feedback Rating Prevention in Web Service Recommendation Systems", *IEEE Transactions on Services Computing*, Vol. 8, No. 5, pp. 755-767, 2015.

[21] Siddu P. Algur and Jyothi G. Biradar, "Rating Consistency and Review Content Based Multiple Stores Review Spam

Detection", *Proceedings of International Conference on Information Processing*, pp. 685-690, 2015.

[22] Shangguang Wang, Zibin Zheng, Qibo Sun, Hua Zou and Fangchun Yang, "Evaluating Feedback Ratings for Measuring Reputation of Web Services", *Proceedings of IEEE International Conference on Services Computing*, pp. 192-199, 2011.

[23] Meng Qiu, Liang He and Jing Xue, "A Model for Feedback Credibility of Trust Management in Web Services", *Proceedings of International Seminar on Future Information Technology and Management Engineering*, pp. 201-205, 2008.

[24] S. Xie, Q. Hu, J. Zhang and P.S. Yu, "An Effective and Economic Bi-Level Approach to Ranking and Rating Spam Detection", *Proceedings of IEEE International Conference on Data Science and Advanced Analytics*, pp. 1-10, 2015.

[25] Nitin Jindal and Bing Liu, "Analyzing and Detecting Review Spam", *Proceedings of 7th IEEE International Conference on Data Mining*, pp. 547-552, 2007.

[26] Haoran Xu and Yuqing Sun, "Identify User Variants based on User Behavior on Social Media", *Proceedings of IEEE 34th International Conference on Performance Computing and Communications*, pp. 1-8, 2015

[27] Denil D. Kotzias, Misha Denil, Nando De Freitas and Padhraic Smyth, "From Group to Individual Labels using Deep Features", *Proceedings of the 21th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 597-606, 2015.

[28] Abhijeet S. Kurle and Kailas R. Patil, "Survey on Privacy Preserving Mobile Health Monitoring System using Cloud Computing", *International Journal of Electrical, Electronics and Computer Science Engineering*, Vol. 3, No. 4, pp. 31-36, 2015.

[29] Samay S Omanwar, Kailas Patil and Narendra P Pathak, "Flexible and Fine-Grained Optimal Network Bandwidth Utilization using Client Side Policy", *International Journal of Scientific and Engineering Research*, Vol. 6, No. 7, pp. 692-698, 2015.

[30] Dnyaneshwar K Patil and Kailas Patil, "Automated Client-side Sanitizer for Code Injection Attacks", *International Journal of Information Technology and Computer Science*, Vol. 8, No. 4, pp. 86-95, 2016.

[31] Kailas Patil, "Preventing Click Event Hijacking by User Intention Inference", *ICTACT Journal on Communication Technology*, Vol. 7, No. 4, pp. 1408-1416, 2016.

[32] Ronak Shah and Kailas Patil, "Evaluating Effectiveness of Mobile Browser Security Warnings", *ICTACT Journal on Communication Technology*, Vol. 7, No. 3, pp. 1373- 1378, 2016.

[33] Qaidjohar Jawadwala and Kailas Patil, "Design of a Novel Lightweight Key Establishment Mechanism for Smart Home Systems", *Proceedings of 11th IEEE International Conference on Industrial and Information Systems*, 2016.