INDIAN INSTITUTE OF TECHNOLOGY, HYDERABAD

Using Reviewers' Expertise to Rank Product Reviews

by

Suryabhan Rathaur

A thesis submitted in partial fulfillment for the
degree of Master of Technology

in
Computer Science Engineering

under guidance of
Dr. Manish Singh

June 2016

Declaration

I, Suryabhan Rathaur, declare that this thesis titled, "**Using Reviewers' Expertise to Rank Product Reviews**" and the work presented in it are written by my own. I confirm that this work was done wholly or mainly while in candidature for a thesis research. Any part of this thesis has not previously been submitted for any degree or any other qualification at this University or any other institution, this has been clearly stated. I have consulted the published work of others, this is always clearly attributed. I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work. I have acknowledged all main sources of help where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signature

Suryabhan Rathaur

Student Name

CS14MTECH11015

Roll Number

Approval

This Thesis entitled "**Using Reviewers' Expertise to Rank Product Reviews**" by Suryabhan Rathaur is approved for the degree of Master of Technology from IIT Hyderabad

C. Krishna Mohan

Examiner
Dept. of CSE
IITH

L.B Vaneeli

Examiner
Dept. of CSE
IITH

MSingh

(Dr. Manish Singh) Advisor
Dept. of CSE
IITH

U. Ramakrishna
U.RAMAKRISHNA

Chairman
Dept. of CSE
IITH

Abstract

E-commerce websites are increasingly using user-supplied reviews to make the decision making experience better for the users. However increasing online presence of users has resulted in numerous amount of reviews in these online portals. Reading all these reviews is not expected of users and online portals do provide ranking of reviews based on helpfulness as a solution. However these methodologies are not perfect and so new reviews even if they might be of good quality do get ignored. Also experience of user is not taken into account. Here we propose an approach to improve the ranking mechanism by including the experience of a user as a factor and also incorporating the concept of a domain expert. Further on, we examine the idea of related categories when finding domain experts. We use Amazon dataset to implement our methodology and report our findings.

Acknowledgements

I express my deepest gratitude to all the people who have supported me and encouraged me during the course of this project. I am very grateful to my thesis advisor Dr. Manish Singh for giving me the opportunity of carrying out this research under their guidance. I am very deeply indebted to him for his support, advice and encouragement without which this research could not have proceeded smoothly. He had encouraged me to think in a scientific and methodical manner.

Finally, I thank to my family members and my friends for their continuous motivation and encouragement.

Contents

Chapter 1

Introduction

Over the last decade, the growth of e-commerce website has been at exponential rate. Similarly, consumer presence on online platform has increased drastically. Presence of large number of shopping portals gives a lot of options to choose from and enables competition among e-commerce player to provide consumer with better shopping experience. But unlike brick-and-mortar shops, the consumer cannot get the physical feel of product required to make the purchasing decision. Only the specifications of product are not enough for customers. Due to the lack of this physical experience of the product, consumers want to know the experience of other consumers who have used the product. This feedback of past users is provided in the format of consumer reviews. Consumer reviews can heavily influence the opinion of a prospective buyer and hence, reviews have a huge impact on the sales of e-commerce websites [1]. Consumers spend a lot of time doing their own version of research and comparison before actually buying a product. They try to find about the quality and usefulness of product and what recommendation other users provide, thus making the role of consumer reviews very important for a product [2].

With the rise of consumers on online platform, more and more users are contributing their experience. This means a single product can have thousands of reviews. Even in India, a relatively new market for e-commerce, products have up to 17k reviews. More number of reviews means more and more information. But it is very difficult for a customer to read all those reviews. Also, not all reviews are helpful and relevant. So, only the relevant information needs to be extracted and presented to the consumer that user can easily access them rather than going through the plethora of reviews. Extracting useful information form consumer reviews either by relevance ranking or by summarization has been a very active area of research in the past decade. Various approaches include providing statistical summary of all the reviews combined [3], applying feature extraction to inform consumers about the quality of different features of a product, using sentiment analysis to find the best subset of reviews to be displayed [4], and to rank the reviews using different parameters [5] [6] [7] [8].

Currently, e-commerce portals rely on ranking of reviews. Majority of platforms employ the concept of up-votes and down-votes for ranking of reviews. Every review can be either up-voted or down-voted. Review having more number of up-votes as well as a good ratio of up-votes to down-votes is assigned a higher rank. Other proposed approaches to optimize ranking of reviews include finding deviation of reviews from the core of a virtual optimal review [7], predicting reviews' helpfulness or impact [5] [6], calculating reviews' expected affect on sales [6] and detecting low quality reviews by applying link analysis based ranking method [8].

Some e-commerce platforms (Amazon, snapdeal etc.) also provide a ranking of customers writing reviews (reviewers). Ranking of a reviewer is affected by multiple factors like total number of reviews, ratio of up-votes to down-votes and activeness of the reviewer. A reviewer with a high ranking is supposedly a good reviewer. Reviewer ranking is a popular feature in Amazon.com. Consumers followed reviews written by a high-ranked reviewer and thus made their purchasing decision based on opinion of such a reviewer. This shows that experienced reviewer can also be a part of decision-making system. But there has been no effort by online portals to take advantage of reviewer's ranking in providing better quality reviews.

Up and down voting system is not a good approach for recent reviews. For a product having high number of reviews, the top subset of reviews gets static after some time. This happens because consumers normally read only the first page of reviews and votes are also assigned to top reviews. There will be many good quality reviews which will be not seen by consumers as they are new and hence quite low on ranking and as no one reads them, there will be no voting for them. Even a high-ranking reviewer will face the same problem. Our work is to provide a solution to this problem by optimizing the top set of reviews by inclusion of recent reviews especially those written by high ranking reviewer.

There are various technical challenges in ranking of reviewers. Our initial approach was to define a ranking function for reviewers by using a scoring system. Ranking function is based on parameters like up-votes, down-votes, number of reviews by reviewer and activeness of reviewer. This ranking score was then to be used to define ranking of a review. But, ranking is done globally i.e. for all the reviews spanning across different categories of products. Most reviewers generally tend to stick with certain categories only. And they consistently provide quality reviews in those categories. These reviewers can be considered as domain experts in respective categories [9] [10] So, the approach is modified to also do the ranking category wise. This involves identifying the domain experts which was done by making use of ranking of reviewers, how diverse they are in writing reviews and their helpfulness.

Next challenge was to measure expertise across multiple domains. Some categories will have higher number of reviews than other categories. Same criteria cannot be used to find domain expert in such different categories. For example, a review written for a popular mobile may have higher votes and if its reviewer gives a review for a totally different domain then he should not be preferred over a reviewer who writes good reviews but with less number of votes in that domain. Thus, expertise should be a local feature of a category. Factors from other categories should not be included to define the expertise. Approaching the problem in this manner also enables the method to be scalable for future as the requirement to redefine parameters when number of reviews increase is eliminated.

Chapter 2

Related Work

Extracting useful information from reviews is an active research area. It has gathered quite a lot traction in last decade or so after review based sites like Amazon.com, tripadvisor, etc became popular. Park and Kim [2] studied the relationship between reviews and different types of consumers and suggested that different strategies should be adopted for different types of consumers. Vermeulen and Seegers [11] investigated the effect of online hotel reviews on customer preference and reported that positive reviews have positive impact on consumer behavior. In case of product reviews, one approach is review summarization which provides an overall summary of the whole corpus of reviews by applying analytics on the whole review set. To produce an accurate summary of reviews, both sentiment analysis and feature scoring methods are equally important. Hu and Liu [12] made use of adjective synonyms for prediction of orientation of sentiments. Various methods used in opinion summarization include basic sentiment summarization [12], text summarization for product reviews [13], visualization of reviews [14]. Another approach includes selecting subset of reviews. Selection of subset can be based on features [4], majority opinion [15] or a combination of both [16]. Third approach is based on ranking the existing set of reviews. Previous research work in this direction include finding deviation from the core of a virtual optimal review [7], estimating reviews' expected effect on sales [1] [5], analyzing the impact of reviews [6] and ranking the reviews based on helpfulness [17] [18]. Our work is focused on the ranking part and enhancing the helpfulness of reviews.

Our work also involves contribution of reviewer in reviews. Work of [19] discusses the importance and impact of online reviewers. The results showed that users not only focus on the rating of the products but they consider the experience of the reviewer. In the study by Ku et al. 20, they report that users rely more on those reviewers who are reputed and have written a higher number of reviews than others. The same study provides a method to identify reputable users in online sharing communities. Trust for reputable reviewers is what makes sites like Amazon.com to display badges on certain reviewers like (e.g., "top-10 reviewer, top-100 reviewer") as well as ranking of the reviewers. An algorithm, ExpertRank [21], was recently proposed to identify subject experts in online knowledge sharing communities.

Chapter 3

Problem Definition

3.1 Ranking Reviews Independent of Reviewer

As mentioned earlier, e-commerce websites employ the idea of user voting to determine whether a review is helpful or not. A user can mark a review as helpful if he finds it useful or else if he thinks that the review does not provides enough information about the product, then he can mark the review as unhelpful by downvoting the review. Using this information, one can find the helpfulness of a review.

Review Helpfulness : There has been some research on determining the helpfulness of a review and also the factors which make a review helpful [17] [18]. Review helpfulness is intuition that a review can help the consumers know about a product. One way to calculate helpfulness is finding the ratio of number of consumers who voted a review as helpful to the total number of consumers who voted for that review and then use a threshold for helpful reviews. This method is used by Zhang and Tran [17].

3.2 Ranking Reviews Using Reviewer's Domain Expertise

Ranking of reviews on e-commerce sites is generally based on the up-votes and down-votes received by a review and their ratio. One downside of this approach is that it ranks reviews having any number of votes, be it may helpful or unhelpful, higher than those reviews having no votes. This means a review which no one might have read is ranked lower than a review which consumers found unhelpful. Many online portals also provide the option of viewing the reviews in order of most recent reviews first but not all reviews are of good quality and sorting in this order makes customer read a lot of reviews while finding only a few of them actually helpful. Our approach for optimizing ranking of reviews was to incorporate the factor of a reviewer's experience while ranking the reviews. Experience here refers to the past contribution made by the user while writing reviews. However, experience alone is not a sufficient parameter as a user can also write a number of subpar reviews. So, quality of reviews already written by user is also considered. Keeping this in mind we introduce a scoring system for reviewers which is essentially ranking the reviewers. Ranking of reviewers was made popular by Amazon.com. Initially, Amazon.com's primary product were books. A lot of users provided their reviews for books, sometimes even if they have not purchased the book from Amazon.com. Users made their purchasing decision based on these reviews and started following reviewers who wrote quality reviews. Hence, reviewer ranking system was introduced which made finding a quality reviewer easier for other users. Amazon's review ranking system is based on up-votes, down-votes, total number of reviews and recency of reviews.

Domain Expertise : While calculating the ranking of reviewers, all the users throughout all the domains (for example: electronics, clothing, stationary etc.) are considered. But a reviewer who has given good number of reviews in one domain say electronics may not have given good number of reviews in other areas, indirectly telling us that the reviewer does not have much expertise in those other areas. Thus giving importance to such reviewer will overshadow the reviews that are more relevant. The parameter domain expertise estimates how much contribution the reviewer has in the domain under consideration. Considering this factor is important because it will highlight the reviews that were given by reviewers who have good knowledge of the domain. This measure is relative as the calculation is done on the basis of average number of reviews and helpfulness in a category.

We define two terms for calculation of domain expertise, *avgHelp* and *userHelp*. *avgHelp* is the average helpfulness measure of all reviews for a given category and *userHelp* is the average helpfulness measure of all reviews by a particular user in a given category.

3.3 Ranking Reviews Using Reviewers Extended Domain Expertise

Calculating expertise in local manner has a drawback that even if a user is submitting a new review for a related or similar category to the category of which he is an expert, then also his expertise will not be considered. For example, a user having good expertise in category mobile writes a review for a power bank, then there should be consideration of the user's expertise in mobile as both are closely related categories. However, simple domain expertise does not takes this into account. The locally separated expertise works good when a user is submitting reviews in totally different domains like gadgets vs clothing because in such categories it might be better to ignore the other domain expertise, but the same score calculation methodology is not entirely useful in cases like above mentioned example of mobile and power bank.

To consider the similarity of categories, first task is to find how similar any two categories are. This is done by finding the depth of lowest common ancestor(lca) of given two categories. To define lca depth, it should be understood that each product belongs to a category which itself can be a sub-category of a broader category. This hierarchy of categories can be of height greater than 2 and forms a tree. Also, as categories like clothing, electronics, stationery etc are totally different from each other, number of such tress are more than one. If two products belong to related categories, they must have a common ancestor. The factor we are taking in consideration is the distance or depth of this lowest common ancestor to the root node for that category. If this depth is higher, then the both categories in consideration are more likely to be similar.

Chapter 4

Ranking Functions

4.1 Review Score

First we define helpfulness of a review. For a review we have votes which say that a review is helpful as well as votes which mark the review as unhelpful. Using Zhang et al's[9] approach of ratio of helpful votes to total number of votes, we define our helpfulness measure or review score (denoted by r_score)

$$r\_score = log(U_v) * \frac{U_v}{U_v + D_v + 1}$$

where number of up-votes for a particular review are denoted by U_v and number of down-votes by D_v.

In above equation log of up-votes is multiplied to the ratio of up-votes and total votes so that number of up-votes is also considered as a factor. Only taking the ratio will skew the score in favor of review having few up-votes and no down-votes as compared to reviews having very high number of up-votes and very low number of down-votes. 1 is added in denominator to consider the reviews which haven't received any votes. It avoid division by zero.

Recency of a review is another important factor which needs to be considered. Ranking of those reviews which have not received votes in recent time should go down. We can incorporate this factor by introducing time elapsed. Here time t is considered in days.

$$r\_score = log(U_v) * \frac{\frac{U_v}{U_v + D_v + 1}}{log_2(2 + t)}$$

Log factor is present to reduce the drop rate with respect to time elapsed. Constant 2 is added to log to avoid division by zero and make the denominator positive.

4.2 Reviewer Ranking

To calculate reviewer ranking of a user u, we add up the scores of all reviews written by user u and divide the sum by total number of reviews given by. So, if r_score_u denotes the reviews written by u and $(count_{rev})_u$ denotes the count of reviews written by user u, then reviewer score for u can be given by u_score_u, which is calculated using following function:

$$u\_score_u = \frac{\sum_i \left(r\_score_u\right)_i}{\left(count_{rev}\right)_u}$$

Consider the scenario where a reviewer has written only one review overall and that review belongs to a popular product. Due to any number of reasons, that review might become the top voted review and since the the product is popular, the number of votes for the review might be very high. As the reviewer has written only one review, he can't be considered a domain expert unless he writes more reviews. However, as our reviewer score is indirectly based on number of votes, the reviewer may rank higher in the reviewer ranking. So, we use a modified review score r_score_m in which we divide the original r_score of a review by dividing it by total number of reviews the product has to which the review belongs. We are assuming here that if a review has disproportionate number of votes, then the product is a popular one and will contain more number of reviews. Thus, dividing by total number of reviews will make the high review score to come down. The modified review score r_score_m is given by:

$$r\_score_m = \frac{r\_score}{(count_{rev})_p}$$

where $(count_{rev})_p$ denotes the number of reviews present in the product p. This gives us the modified u_score_u which contains the modified review score by user u given by r_score_{um} and rest of the terms according to previous definitions:

$$u\_score_u = \frac{\sum_i \left(r\_score_{um}\right)_i}{\left(count_{rev}\right)_u}$$

4.2.1 Reviewer Score

See algorithm 1 for calculation of reviewer score

Algorithm 1: Calculate the score of a reviewer

function getReviewerScore (u) :
Input : Reviewer Id $= u$
Output: Reviewer Score u_{score}
$x = 0$ $count = 0$
for i : $each\ review\ by\ u$ **do**
  $x = x + modifiedReviewScore(i)$
  $count = count + 1$
$u_{score} = x/count$

4.3 Domain Expertise

We define two terms for calculation of domain expertise, $avgHelp$ and $userHelp$. $avgHelp$ is the average helpfulness measure of all reviews in a category c and $userHelp$ is the average helpfulness measure of all reviews by a particular user u in category c.

$$avgHelp = \frac{\sum_i r\_score\big(rev_c\big)_i}{\big(count_{rev}\big)_c}$$

$$userHelp = \frac{\sum_i \big(r\_score_{c,u}\big)}{\big(count_{rev}\big)_{c,u}}$$

where $\big(count_{rev}\big)_c$ denotes the total number of reviews present in category c, $\big(count_{rev}\big)_{c,u}$ denotes total number of reviews written by user u in category c and subscript c, u refers that reviews are written in category c by user u.

$$ex_s = \frac{userHelp}{avgHelp} * \frac{(count_{rev})_{c,u}}{\frac{(count_{rev})_c}{(count_{user})_c}}$$

where $(count_{user})_c$ is the total number of users in category.

So, ex_s compares the helpfulness of each user with the average of all users and gives us a score which can be used for ranking of reviewers in a category. Second multiplication term is a measure of number of reviews written by user u when compared to the average of the category. In non-formulaic terms, it can be defined as:

$$\frac{number\ of\ reviews\ written\ by\ user\ u\ in\ category\ c}{average\ number\ of\ reviews\ by\ any\ user\ in\ category\ c}$$

4.3.1 Domain Expertise Score

See algorithm 2 for expertise score

Algorithm 2: Calculate domain expertise score of a reviewer

function getExpertiseScore (u, c) :
Input : Reviewer Id $= u$
Category Id $= c$
Output: Expertise Score in category c, $ex_{s,c}$
$x = 0\ count_{rev} = 0$
for $i : each\ product\ in\ c$ **do**
  for $h : each\ review\ of\ i$ **do**
    $x = x + reviewScore(h)$
    $count_{rev} = count_{rev} + 1$
$avgHelp = x/count_{rev}$
$y = 0\ count_{c,u} = 0$
for $j : each\ review\ by\ u\ in\ c$ **do**
  $y = y + reviewScore(j)$
  $count_{c,u} = count_{c,u} + 1$
$userHelp = y/count_{c,u}$
$count_{user}$
for $k : each\ user\ in\ c$ **do**
  $count_{user} = count_{user} + 1$
$avgRev = count_{rev}/count_{user}$
$ex_{s,c} = (userHelp/avgHelp) * (count_{c,u}/avgRev)$

4.4 Extended Domain Expertise

We incorporate the factor of lowest common category (lca) in our existing expertise score to improve the functionality. Let dp_i be the lca depth of i^{th} category from the category being considered now, dp_{max} be the maximum depth of the category being considered, then we calculate final score f_s using following :

$$f_s = \frac{\sum_i (ex_{(s,c)i} * dp_i)}{dp_{max} * \lambda}$$

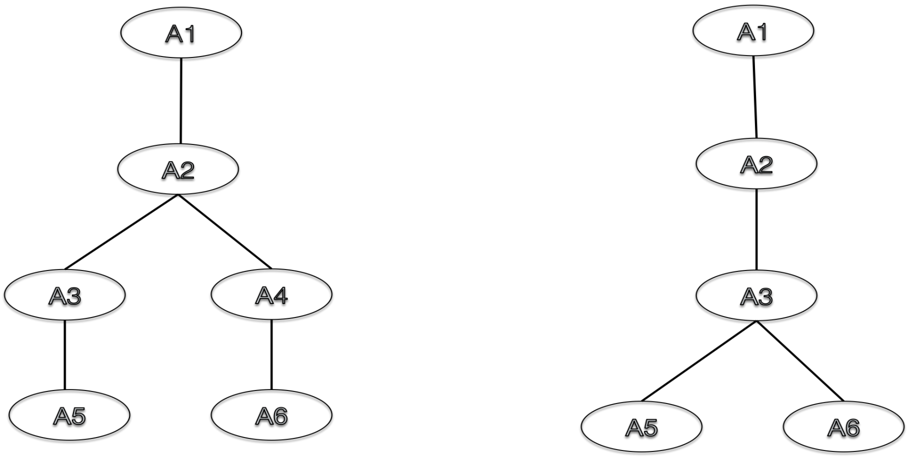

FIGURE 4.1: Fig: In left figure, lca depth of A5 and A6 is 1, so they are less similar. In right figure, lca depth of A5 and A6 is 2, so they are more similar.

where $ex_{s,c}$ is the expertise score of the user in category c and λ term is the count of domains whose lca depth with current category is zero. λ as well as dp_{max} are present in denominator for normalization of the final score.

4.4.1 Final Expertise Score

Algorithm 3: Final expertise score of a reviewer

function getFinalExpertiseScore (u, c) :
Input : Reviewer Id $= u$
Category Id $= c$
Output: Final Expertise Score in category c, $f_{s,c}$
$maxDepth = lca(c, c)$
$tempSum = 0$
$zCount = 0$
for i : *each category* **do**
    $tempSum = tempSum + (lca(c, i) * ex_{(s,c)i})$
    if $lca(c, i) == 0$ **then**
        $zCount = zCount + 1$
$f_{s,c} = tempSum/(maxDepth * zCount)$

Chapter 5

Evaluation

In this chapter, we discuss about dataset we have used and the evaluation process.

5.1 Data

To perform our experiments we used the online product review dataset of Amazon.com available through SNAP[1]. We downloaded the dataset available from Julian McAuley[2].The whole dataset contained more than 34 million reviews. Our experimentation was done on the reviews of Electronis category which originally contained 1.2 million reviews. However, we worked on 152,076 reviews only. There were total 139,711 reviewers who had written at least one review. Out of all reviews, at least 62% reviews have received any kind of (up or down) votes. Among those reviews which have received votes, 50% were voted perfectly helpful i.e., these reviews haven't received any down votes. Also, only 5% of voted reviews were voted perfectly unhelpful.

5.2 Evaluation and Results

For our work, we have used MySql as the database and python as the programming language. As there has been no work previously which tries to combine reviewer expertise and review helpfulness, there are no standard results for comparison. We created our own evaluation process and it also involves some subjective insights in the results.

For evaluation, initially we calculated domain score for each reviewer for each domain he has written a review. Then 50 highest domain scores were selected. Out

[1]Stanford Network Analysis Project
[2]http://jmcauley.ucsd.edu/data/amazon/

of these, 3 were repeated users i.e., 3 users were having high scores in more than one domains. One user had presence in three different domains. So, we considered a total 46 reviewers. For now, we will call them active reviewers. For these active reviewers, we tried to find reviews having very low review scores and the product of which review is being considered should have at least ten reviews. Eight reviewers failed to satisfy one of those criteria. For the rest of the reviewers, following steps were followed:

- Remove the corresponding low score review from the dataset. Note down the product it belonged to and the category.
- Recalculate the different domain scores ex_s for the reviewer.
- Treat the removed record as a new review and calculate extended domain score f_s using the removed record's category.

Using the new extended domain score, we find the new position or ranking of the given review. Review scores of all the reviews of a product, domain scores ex_s and extended domain scores f_s are normalized for the compatibility of the scores. In many product reviews, sometimes top two or three reviews generate much more upvotes than rest of the reviews of product. To contain such cases while generating normalized score of product reviews, we check if the difference of review score if first and third reviews is greater than that of twice of the fourth review. If yes, then we skip top two reviews and perform rest of the operations without them. On performing above procedure on remaining thirty-eight reviewers, we found that for twenty-four of them, there was no change in the ranking of the reviews. The reason being that most of them had not written a review in similar categories and rest of them had very low domain score in even remotely similar categories. For those reviewers whose reviews had gained some amount in the raking of the reviews, some of the data is presented in table 1.

As can be seen from table 1, for some reviewers, the increase in rank was significant while for others, it was a small gain. Again, we would emphasis on the fact that it depends on the similarity of the categories. A more similar or even same category would cause significant increase in the ranking. Out of 50 considered reviewers, 28% reviewers' low rated reviews' ranking improved through our methodology. On subjective analysis of these improvements, we have found that these originally low-rated reviews by high ranking reviewers are actually much detailed and better insight providing than the actual top reviews. So, our methodology does enhances the ranking of reviews.

Reviewer Id	Product Id	Reviews in Product	Rank Improvement
A30IW59HCR7WUQ	B00005ARK3	808	137
A1DZM5N7UEBV25	B00003006R	104	29
AQCHVXD1R9WYI	B00004T12N	12	4
ANLA598UNJI8A	B00000JDFO	15	4
A231WM2Z2JL0U3	B00005ARK3	808	187
A231WM2Z2JL0U3	B00005NGQF	53	13
A231WM2Z2JL0U3	B00005R098	41	7
A1GPGBHBI6T2HJ	B00004WHF4	47	11
A243HY69GIAHFI	B00004VX3T	193	126
A2B21POKQ3N09H	B00005AW1X	64	28
A14ME4FQBNFYWH	B000062STU	27	10

TABLE 5.1: Ranking gained in reviews after applying the given algorithms

Chapter 6

Conclusion

In this report, ranking of product reviews and ways to optimize the ranking has been discussed. Initially, different types of methods to extract useful information from reviews was discussed. An approach to enhance the ranking system by also incorporating experience of reviewer has been proposed. The approach first started with baseline ranking of reviews, then concept of domain experts in product review system was introduced. An approach to integrate domain expertise into baseline review system was proposed. Then the challenge of taking into account the concept of related categories for expertise was solved by finding similarity between categories. All these approaches are presented in combined formulation to enhance the ranking process of the reviews. Our work has combined different areas and there has been set evaluation methodology to judge our results but on subjective analysis, we find that our methodology does provides positive results. In future, incorporating learning methods like SVM regression and also making use of predicting helpfulness of a review to our work can further improve the work done here.

Bibliography

[1] Chrysanthos Dellarocas, Xiaoquan Michael Zhang, and Neveen F Awad. Exploring the value of online product reviews in forecasting sales: The case of motion pictures. *Journal of Interactive marketing*, 21(4):23–45, 2007.

[2] Do-Hyung Park and Sara Kim. The effects of consumer knowledge on message processing of electronic word-of-mouth via online consumer reviews. *Electronic Commerce Research and Applications*, 7(4):399–410, 2009.

[3] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50. ACM, 2006.

[4] Panayiotis Tsaparas, Alexandros Ntoulas, and Evimaria Terzi. Selecting a comprehensive set of reviews. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–176. ACM, 2011.

[5] Anindya Ghose and Panagiotis G Ipeirotis. Designing ranking systems for consumer reviews: The impact of review subjectivity on product sales and review quality. In *Proceedings of the 16th Annual Workshop on Information Technology and Systems*, pages 303–310, 2006.

[6] Anindya Ghose and Panagiotis G Ipeirotis. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *Proceedings of the ninth international conference on Electronic commerce*, pages 303–310. ACM, 2007.

[7] Oren Tsur and Ari Rappoport. Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews. In *ICWSM*, 2009.

[8] Jianwei Wu, Bing Xu, and Sheng Li. An unsupervised approach to rank product reviews. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on*, volume 3, pages 1769–1772. IEEE, 2011.

[9] Alexandre Klementiev, Dan Roth, Kevin Small, and Ivan Titov. Unsupervised rank aggregation with domain-specific expertise. *Urbana*, 51:61801, 2009.

[10] Michael D Lee, Mark Steyvers, Mindy De Young, and Brent J Miller. A model-based approach to measuring expertise in ranking tasks. In *Proceedings of the 33rd annual conference of the cognitive science society*, 2011.

[11] Ivar E Vermeulen and Daphne Seegers. Tried and tested: The impact of online hotel reviews on consumer consideration. *Tourism management*, 30(1):123–127, 2009.

[12] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[13] Jiaming Zhan, Han Tong Loh, and Ying Liu. Gather customer concerns from online product reviews–a text summarization approach. *Expert Systems with Applications*, 36(2):2107–2115, 2009.

[14] Ming C Hao, Umeshwar Dayal, Daniel Keim, and Daniela Oelke. Generating a visualization of reviews according to distance associations between attributes and opinion words in the reviews, July 30 2009. US Patent App. 12/462,186.

[15] Theodoros Lappas and Dimitrios Gunopulos. Efficient confident search in large review corpora. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 195–210. Springer, 2010.

[16] Theodoros Lappas, Mark Crovella, and Evimaria Terzi. Selecting a characteristic set of reviews. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 832–840. ACM, 2012.

[17] Richong Zhang and Thomas Tran. Helpful or unhelpful: a linear approach for ranking product reviews. *Journal of Electronic Commerce Research*, 11(3):220, 2010.

[18] Susan M Mudambi and David Schuff. What makes a helpful review? a study of customer reviews on amazon. com. *MIS quarterly*, 34(1):185–200, 2010.

[19] Nan Hu, Ling Liu, and Jie Jennifer Zhang. Do online reviews affect product sales? the role of reviewer characteristics and temporal effects. *Information Technology and Management*, 9(3):201–214, 2008.

[20] Yi-Cheng Ku, Chih-Ping Wei, and Han-Wei Hsiao. To whom should i listen? finding reputable reviewers in opinion-sharing communities. *Decision Support Systems*, 53 (3):534–542, 2012.

[21] G Alan Wang, Jian Jiao, Alan S Abrahams, Weiguo Fan, and Zhongju Zhang. Expertrank: A topic-aware expert finding algorithm for online knowledge communities. *Decision Support Systems*, 54(3):1442–1451, 2013.