Indian Journal of Science and Technology, *Vol 8(12), 61768, June 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

An Effective Approach to Rank Reviews Based on Relevance by Weighting Method

J. Meghana Ramya Shri* and V. Subramaniyaswamy

School of Computing, SASTRA University, Thanjavur, Tamilnadu, India; meghanagopal186@gmail.com, vsubramaniyaswamy@gmail.com

Abstract

Mining all the useful reviews from the social networks is the attracting research topic in recent years. It is necessary to analyze the reviews to know about the product or about the topic. Existing works only focus on data extraction and classifying the reviews into positive or negative and spam or not spam. The review relevance, which is very important to rank reviews, has not been considered in the existing work. In this paper, we put forward a method to calculate review relevance. We calculate the review relevance value not only by considering the similarity and correlation, but also the votes for each review. It is proved that our method works well in terms of effectiveness and accuracy. Thus, this work helps in effectively retrieving the useful reviews from social networks.

Keywords: Correlation, Opinion Mining, Review Relevance, Similarity, Votes

1. Introduction

Growth in the web technology has resulted in the drastic increase of user generated content in the form of reviews or comments. Users can liberally post their opinions about the article in any website. They can support or criticize about any article. All these reviews or comments posted by the people are considered to be very important. These reviews are very helpful to analyze about the product or the article. For a product these reviews are helpful for the customers and the merchants. Customers can make purchase decisions and the merchants can make use of the reviews to improve their products.

Opinion mining is the process of analyzing and shortening the user generated content which is helpful to the public. Many researchers have been focused on classifying the content into positive and negative reviews or based on spam or non spam. The extent of relevance of the review with the article was not considered in the past research. The user generated content can contain some non relevant or less relevant content. Users can post their own review, which may be related or not related to the article. They can also post some website links or can reply to anyone else review. All this content can have different importance with the article. Content which is totally not relevant to the article are considered as noisy reviews. Thus it is necessary to sort all the reviews according to the relevance. The reviews with higher relevance are considered to be more important than the less relevant ones. The existing work deals with the review pertinence. Review pertinence is calculated by considering the similarity and correlation.

Natural Language Processing (NLP) is a subfield of manmade brainpower furthermore, computational etymology. It concentrates on the issues of robotized era and comprehension of regular human dialects.

Reviews may be very big and are very difficult to understand the meaning of the sentence. Different people use different words to express the same meaning. Similarity measures like Overlap Coefficient and Jaccard Coefficient is used commonly in the existing systems to measure the relevance between the review and its corresponding article. People who want to purchase a product or who are interested to know about an article mainly

**Author for correspondence*

read the top 20 to 25 reviews. These reviews are very important to the people to make decisions, so they must be very accurate and fully relevant to the article. Our work mainly focuses on ordering the most relevant reviews on the top thus by leaving the less relevant reviews at the bottom. Totally unrelated reviews are considered as unwanted or spam reviews and are discarded.

This paper deals with the process of measuring relevance by considering the similarity, correlation and votes. The data set for the process is obtained from the website dynamically using the web crawler.

2. Related Work

Since the user generated content is very important to make decisions, many researchers in mining the opinions mainly focus on identifying the polarities of user data. The reviews are classified into positive and negative reviews. This classifier draws on data recovery procedures for highlight extraction and scoring, and the outcomes for different measurements and heuristics fluctuate relying upon the testing circumstance. The best routines act and additionally or better than conventional machine learning. At the point when working on individual sentences gathered from web seeks, execution is constrained because of commotion and vagueness[1–3].

Most of the work in opinion mining is domain dependent and focus only on one or two particular domains. Work has been done to automatically obtain the summarization of the reviews. Lu et al. analyzed the process of summarizing the reviews into ratings[4]. Hu and Liu implemented the technique of semantic analysis to summarize all the user reviews for the product[5]. Opinion mining methods were also used to mine the user comments about the election prediction system[6]. Wei et al. proposed a method to label all the attributes of a product by using a sentiment ontology tree[7].

In[8], the preference method was proposed to mine user preferences and map the preferences into numerical rating scale. In[9], the features from the reviews are extracted and automatically find the people who have commented based on the same features.

In[10], the different types of opinions are surveyed and the idea for retrieving the reviews from online using crawler is obtained. The[11] deals with term weighting, where the methods are proposed to weight the terms in the reviews. Term weighting is calculated based on the importance of the term in the review. The review pertinence is calculated by considering the similarity and the correlation[12]. It is clear that considering correlation does not perform well in terms of computational efficiency.

The unrelated data which are not useful for the user are removed and are considered as spam. Spammers include the spam data which takes the user to some other unrelated websites. Some include the spam data to promote or to degrade the product. The[13] explains the trustworthiness of the online reviews. Existing researches mainly focus on simply classifying the reviews into spam and non-spam. Work has been proposed to simply rank the comments which are non-spam. This method does not consider the degree of relevance of the review with the article.

In the proposed work, we rank reviews based on their relevance with the accordance with the article. The existing system deals with the similarity and the correlation for finding the review relevance. The above work has not considered the votes of the review which are also very helpful in ranking the reviews. As in[14], not all the reviewers vote for the comments provided by the user. This occurrence is called as the "words of few mouths". These votes also provide additional information about the importance of the review. Sapna Zol et al. explain the applications, process and challenges[17] in sentiment analysis. In[18], quantified summary is obtained based on preference in four dimensions such as class, polarity, frequency wise and review contrast summarization.

We propose a method to rank the reviews based on relevance by considering similarity, correlation and votes. We integrate the results of similarity, correlation and votes to sort the reviews based on the relevance.

3. Problem Outline

We consider an article A, which contains many review named as r_1, r_2, r_3, ….r_n. Each review may have up votes or down votes. We used the following methods to sort the reviews based on the relevance. The following is the summary of the methodology which is implemented. Initially data from the website is crawled dynamically and are stored in the database.

- Since the crawled content consists of unrelated data, it is necessary to preprocess the content.
- The similarity between the each review and the article is calculated.
- The correlation between reviews is calculated.

- Number of votes for each review is obtained.
- All the values are integrated and finally sorted to find the review relevance with the article. In the following section, we first analyze the types and arrangement of reviews.

3.1 Types of Reviews

Reviews are classified into different types[12] accordingly

Common Review

This type of review contains the general opinion about the product. The opinion may be a positive comment or a negative comment.

Irrelevant Review

These reviews are not relevant or less relevant to the article or to the product. These reviews contain some random text which is unrelated to the article.

Comment Review

These reviews comment on other reviews. These reviews may be supporting or opposing the other reviews.

Link Review

In some articles reviewers post some other website links so that other reviewers use that link. Mostly all these link reviews are not useful to the reviewers.

3.2 Review Structure

Each website has its own review structure. Each part in the review has its own role[12] which is described as follows

Reviewer Details

Reviewer details contain the name of the reviewer or the IP address of the reviewer. This reviewer details helps to know about the valid reviewer.

Review Content

This part contains the valuable information about the product or the article. The reviewer uses natural language to express their opinion.

Review Vote

Users can vote for the reviews to express their opinion. The reviewer can support for the review by choosing thumps up symbol or can oppose for a review by choosing thumps down symbol.

4. Proposed System

This section is arranged in the following manner. Initially, the data set is preprocessed and then the similarity rank in calculated. Thirdly, correlation rank is calculated which is followed by vote rank calculation. Finally, the review relevance value is calculated. The flow diagram of our proposed work is as follows (Figure 1).

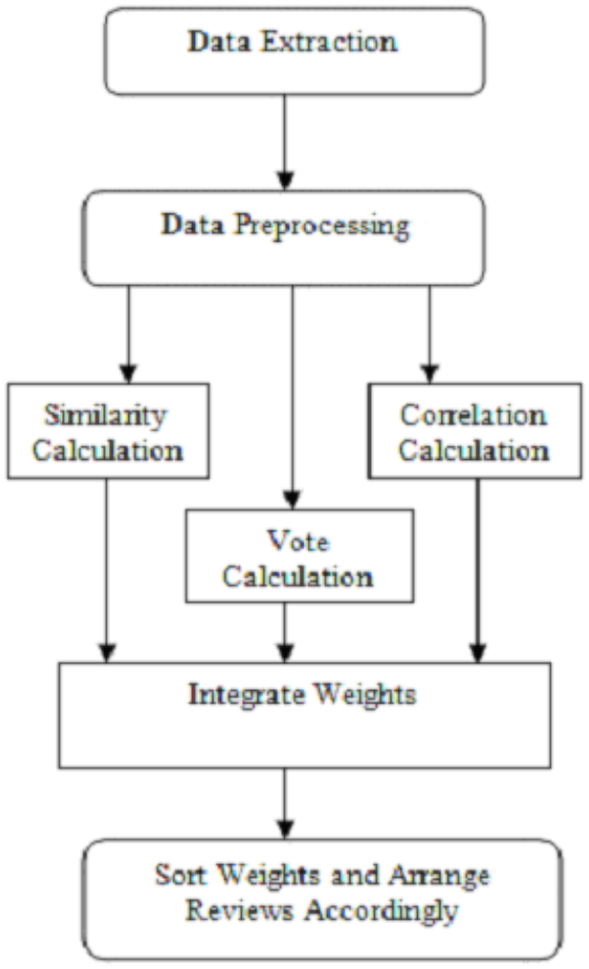

Figure 1. Flow diagram for calculating review relevance.

4.1 Preprocessing

The data is dynamically crawled from the website using the crawler. The HTML parser is used to analyze the dataset. The crawled data needs to be preprocessed for effective results. The first step in preprocessing is tokenization, where all the words in the dataset are divided into tokens. The second step is Parts of Speech tagging (POS) and finally stemming is performed. The preprocessed dataset is used for further relevance calculation.

4.2 Similarity Calculation

Similarity can be measured as the degree of relevance between the review and its corresponding article type. Similarity can also be measured by considering the word overlap which is also a very important factor for review pertinence. Similarity here is measured by considering the relevance between the review and its article. The Vector Space Model (VSM)[15] is the best way to compute the similarity between the review and the article is used here. The cosine formula for finding the similarity is as follows:

$$\text{SimRank (r,A)} = \frac{\sum_{i}^{n} c(w_i, r)c(w_i, A)}{\sqrt{\sum_{i}^{n} c(w_i, r)^2}\sqrt{\sum_{i}^{n} c(w_i, A)^2}} \quad \ldots(1)$$

Where C(wi, r) is the number of times the word w appears in the review r, c(w, A) is the times the word w appears in the article A. Vector Space Model has proved to be the best method for finding the similarity between the two documents. But it has some shortcoming of finding the similarity between the review and the article.

The position where the word is placed in the sentence gets its own importance. The words which are placed in the first line of the sentence or at the starting of the sentence get more importance than other words. The most commonly used words in the document get some importance. Thus the weight of each word in the article is calculated as

$$\text{Weight (w, A)} = C(w, A) * M * \text{Pos}(w), \quad (2)$$

Where, weight (w, A) is the word weight w in the article *A*. (w, A) is the number of times the word w that appears in A and M is the total number of reviews that contain the word w. Pos (w) is the position of the words in the review. The weights are assigned accordingly,

$$\text{Pos}(w) = \begin{cases} 1.5, \text{ if the word is present in the title.} \\ 1.3, \text{ if the word is in the first or the last line but not in the title.} \\ 1, \text{ others.} \end{cases}$$

The same words can be expressed in different forms. These semantically related concepts cannot be identified using VSM. Thus the WordNet ontology is used to identify the related meanings of all the words. It is a lexical database which groups English vocabulary into a set of synonyms and provides a brief description about the synonym relation. Thus the formula for calculating the semantic similarity between two words is calculated using WordNet ontology is as follows,

$$\text{Semantic}(w_1, w_2) = \max\text{Semantic}(C_{1i}, C_{2j}), \quad (4)$$

Thus the similarity is finally calculated by integrating all the above formulae as follows.

$$Sim(r, A) = \frac{\sum_{i}^{n}\sum_{j}^{n} c(w_i, r)\,weight(w_j, A)\,Semantic(w_i, w_j)}{\sqrt{\sum_{i}^{n} c(w_i, r)}\sqrt{\sum_{i}^{n} c(w_i, A)^2}} \quad (5)$$

4.3 Correlation between Reviews

Some reviews which have high pertinence need not be very similar to the article. Thus, calculating review relevance only based on the similarity is not very efficient. To overcome the above disadvantage, correlation between the reviews is identified. An undirected graph is drawn based on the cosine similarity, where each node is a review and the value inside each node is the review's relevance with the article. The edge weight between the two nodes is the value of cosine similarity between two reviews. If the similarities between two reviews exist, then the corresponding nodes are linked as neighbors. From the above graph, the correlation between the reviews is estimated using the random walk algorithm[16] as follows,

$$Per(r_i, A) = \sum_{r_j \in adj[r_i]} \frac{w(r_j, r_i)}{\sum_{r_k \in adj[r_j]} w(r_j, r_k)} Per(r_j, A) \quad (6)$$

Where, w (r_j,r_i) is the cosine similarity of two reviews, adj[r_i] is the neighbors of the review ri. The above formula is an iteration process and a static value for correlation is obtained only after several iterations.

4.4 Votes Calculation

Voted rank is the method used to estimate the review pertinence by using the number of votes obtained in the review. Almost in all websites, votes are taken into account and reviews are arranged accordingly. Votes along with similarity and correlation are not used to calculate the review relevance to the best of our knowledge. Votes can be up votes or down votes. If the number of votes for the review is high, then voted rank is high, which gives additional weightage to the relevance value. Most of the people, who read all the comments, may like or dislike the comments. The formula for calculating the voted rank is as follows,

$$VoteWeight = \frac{V(r_i, A)}{\sum_{j=1}^{n} V(r_j, A)} \quad \text{where, } 1 \le i \le n \quad (7)$$

Where, V (r_i, A) is the votes obtained for single review and V (r_j, A) is the total number of votes obtained for the full article.

4.5 Integrated Method

To obtain more improved results we integrate the above methods to calculate the review relevance. We calculate the review relevance by combining both the similarity between the review and the article and correlation of reviews along with votes obtained for each review. The integrated formula is as follows,

Where, r is the review that occurs in the article A. d is the damping coefficient that controls the substitution between two items in the formula. Pertinence value is estimated whenever a new review occurs using the above integrated formula. The above formula is used for repeatedly updating the relevance value for each node. The stationary relevance value is obtained after much iteration. To obtain the stationary relevance value an iterative algorithm known as power method is applied. The algorithm is described as follows,

$$\frac{V(r_i, A)}{\sum_{j=1}^{n} V(r_j, A)} + d \times \frac{\sum_{i}^{n} c(w_i, r) c(w_i, A)}{\sqrt{\sum_{i}^{n} c(w_i, r)^2} \sqrt{\sum_{i}^{n} c(w_i, A)^2}}$$

$$+ [1-d] \left[\sum_{r_j \in adj[r_i]} \frac{w(r_j, r_i)}{\sum_{r_k \in adj[r_j]} w(r_j, r_k)} Per(r_j, A) \right] (8)$$

4.5.1 Algorithm 1:

Input: Similarity rank, correlation rank, vote rank.

Ω : iteration termination value

Output: vector p_k: stationary relevance value for all reviews.

1. Random vector is set for p_0
2. while k=0;
3. repeat
4. k++;
5. calculate relevance value using the integrated formula
6. above relevance value is used to form vector p_k
7. $\Delta = || p_k - p_k\text{-}1||$;
8. Until $\Delta < \Omega$
9. Return p_k

Where, Ω is used to manage iteration termination. Difference between p_k and p_k-1 is denoted using $|| p_k - p_k\text{-}1||$. If Ω value is greater than $|| p_k - p_k\text{-}1||$, then the iteration is automatically terminated.

5. Evaluation Result

The reviews from following websites are used for experimental research, http://www.androidpolice.com/, http://www.phonearena.com/, http://moneycontrol.com/. In these websites people can know the information about the products or article by reading the reviews. They can also post comments and vote for the reviews to express their views. Some of the reviews posted may be relevant or irrelevant to the article, in such a case the reviews with the higher relevance value will be ranked first. The data from the websites are crawled dynamically using web crawler and the reviews are analyzed using the html parser.

NDCG (Normalized Discount Cumulative Gain) method is used for evaluating the review pertinence performance. NDCG method considers two rules while estimating the relevance. First rule that the NDCG

NDCG (Normalized Discount Cumulative Gain) method is used for evaluating the review pertinence performance. NDCG method considers two rules while

$$NDCG@N \equiv Z_N \sum_{j-1}^{N} \frac{2^{rel(j)}-1}{\log(1+j)} (9)$$

Where N is the total number of reviews obtained for the article and Z_N is the normalization value. NDCG@N values are used for retrieving both the top and the bottom N ranked reviews.

In evaluation, the pertinence is first calculated only by considering the VoteRank, and then by the SimRank. Finally, our proposed method is estimated by considering the SimRank, correlation rank and VoteRank.

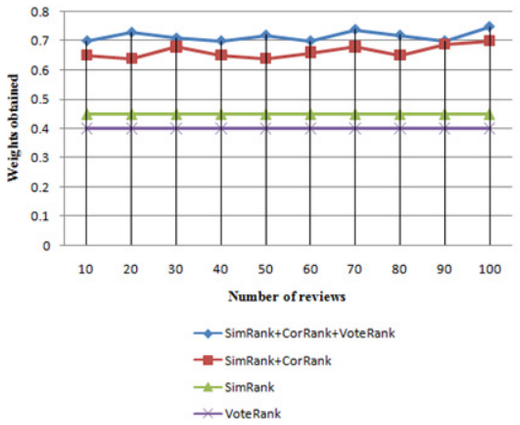

Figure 2. Ranking the reviews based on high relevance.

According to Figure 2, it is clear that combination of SimRank,VoteRank and correlation rank performs well when compared with SimRank and VoteRank individually.

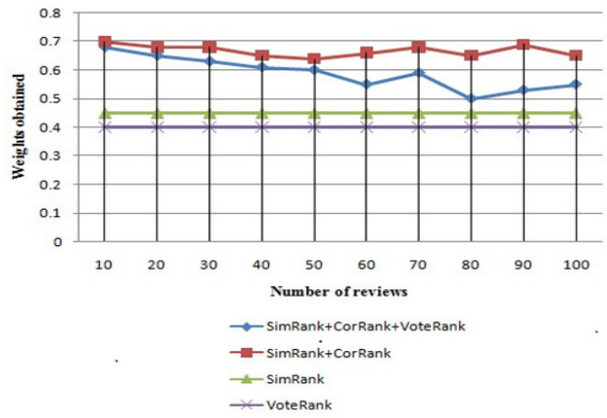

Figure 3. Ranking the reviews based on low relevance.

According to Figure 3, it is clear that this method identifies the reviews with low relevance efficiently. These low relevance values are less important to the article and are considered as spam which can be discarded. Thus this method efficiently removes the spam or less relevant reviews from the social website.

6. Conclusion and Future Work

The reviews from the website are crawled dynamically for the experimental purpose. Thus based on the evaluation, it is clear that combination of SimRank, CorrelationRank and VoteRank performs well by arranging the related reviews in the top by leaving unrelated reviews at the end. Our work is very helpful in organizing relevant reviews of the article first and reviews with low relevance are eliminated by considering those reviews as spam. In future, we will consider the quoted rank along with the similarity, correlation and votes. This method can be improved further by applying parallel programming.

7. References

1. Liu B. Sentiment analysis and subjectivity. Handbook of natural language processing. 2010; 2:627–66.
2. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: Dave K, Lawrence S, Pennock DM, editors. Proceedings of the 12th international conference on World Wide Web; 2003.
3. Tang H, Tan S, Cheng X. A survey on sentiment detection of reviews. Expert Systems with Applications. 2009; 36(7):10760–73.
4. ACM. Rated aspect summarization of short comments. In: Lu Y, Zhai C, Sundaresan N, editors. Proceedings of the 18th international conference on World wide web; 2009.
5. ACM. Mining and summarizing customer reviews. In: Hu M, Liu B, editors. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining; 2004.
6. Citeseer. Crystal: Analyzing Predictive Opinions on the Web. In: Kim S-M, Hovy EH, editors. EMNLP-CoNLL; 2007.
7. Association for Computational Linguistics. Sentiment learning on product reviews via sentiment ontology tree. In: Wei W, Gulla JA, editors. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics; 2010.
8. Leung CW-K, Chan SC-F, Chung F-L, Ngai G. A probabilistic rating inference framework for mining user preferences from reviews. World Wide Web. 2011; 14(2):187–215.
9. Wang H, Liu L, Song W, Lu J. Feature-based sentiment analysis approach for product reviews. Journal of Software. 2014; 9(2):274–9.

10. Sudhakaran P, Hariharan S, Lu J, Gamasu R, Hamid A, Ilyas M, et al. Research directions, challenges and issues in opinion mining. International Journal of Advanced Science and Technology. 2013; 60:1–8.
11. Deng Z-H, Luo K-H, Yu H-L. A study of supervised term weighting scheme for sentiment analysis. Expert Syst Appl. 2014; 41(7):3506–13.
12. Wang J-Z, Yan Z, Yang LT, Huang B-X. An approach to rank reviews by fusing and mining opinions based on review pertinence. Information Fusion. 2015; 23:3–15.
13. ACM. Opinion spam and analysis. In: Jindal N, Liu B, editors. Proceedings of the 2008 International Conference on Web Search and Data Mining; 2008.
14. Zhang R, Tran T, Mao Y. Opinion helpfulness prediction in the presence of 'words of few mouths'. World Wide Web. 2012; 15(2):117–38.
15. Salton G. Automatic text processing: the transformation, analysis, and retrieval of information by computer. Reading: Addison-Wesley. 1989.
16. Page L, Brin S, Motwani R, Winograd T. The PageRank citation ranking: bringing order to the web. 1999.
17. Zol S, Mulay P. Analyzing Sentiments for Generating Opinions (ASGO) - a new approach. Indian Journal of Science and Technology. 2015; 8(S4):206–11.
18. Prakash S, Chakravarthy T, Brindha G. Preference Based Quantified Summarization of on-line reviews. Indian Journal of Science and Technology. 2014; 7(11):1788–97.