Numeric Rating of Apps on Google Play Store by Sentiment Analysis on User Reviews

Mir Riyanul Islam

Department of Computer Science
American International University - Bangladesh
Banani, Dhaka-1213, Bangladesh
mirriyan41@gmail.com

***Abstract*—The sudden eruption of sentiment analysis and opinion mining has opened new possibilities to improve our information gathering interests. We are always keen to know what others say about the devices or applications we are going to use. Its observed that sometimes the numeric rating has vast difference than the reviews given by the users. To remove this ambiguity a unified rating system has been proposed here. The starred rating and a generated numeric polarity of the reviews are combined to generate the final rating. The proposition is based on sentiment analysis and an optimized probabilistic approach described by a group of researchers. The approach is proved for its efficiency in a diverse corpus of writings where the targets are of different categories.**

Keywords*—*Sentiment Analysis, Polarity Extraction, Numeric Rating, Apps Review.

I. Introduction

OTHERS opinions are always important for us during any decision making process. Before the wide spread of the World Wide Web we used to ask our friends about anything we were going to use. But now a days we need to ask someone for the suggestions. All the reviews and recommendations are available on online review sites and blogs. To be more specific, the users of devices with Android IOS generally choose their required apps from the Google Play Store. Generally it is seen that people makes the decision for any app on the basis of the numeric rating of that. The rating is the average of all the ratings given by other users by stars. Moreover, the users have to include a comment as well. It is observed that ambiguity lies between the star rating and the comments of the users. It creates confusion to the new user who is going to download and use the app. Here another problem evolves, people always go for a summary rather than an elaborative statement. So this problem rises with ambiguity of reviews and the biasness of users to summarized option.

The problem described above has two sub-problems. Firstly the ambiguity. This can be easily removed by using anyone of the two review types, star rating and comment. Another sub-problem is the biasness to the summarized rating of the users. Earlier most of the approaches were only to find out the polarity of the comments bound within negative, neutral and positive. Moreover once the problem was treated with the help of opinion mining with support vector machine [1]. As a part of the same approach later it was shown with the well-known Naive-Bayes model [2]. All of the earlier approaches were only to extract a rating from the comments leaving the star rating part aside.

The problem considered here some challenges such as, the user reviews are qualitative and the rating should be done in a quantitative scale. Moreover, users are not always formal while writing the reviews. The sentiment expressions in user reviews varies diversely. This diverseness can be in single words like as awesome, worthy, etc., as well as in case of multi-word phrases like time wasting, great to use etc. Moreover, there can be mixture of formal and slang expressions too with abbreviations and spelling fluctuations. On a research done on 3000 tweets, it is seen that 15.25% expressions contains slangs [3]. The process have to be developed such that it can differentiate and handle with this diverse domain of sentiment expressions.

Extracting sentiment expressions was done earlier from formal corpus. Generally the user reviews are not written following the formal rules. So, the exact sentiment expression cant be extracted from the user reviews with some predefined patterns. In a word, the writing style of the user reviews cant be easily parsed with various parsers and parts-of-speech taggers. These tools are typically made on the basis of standard spelling and grammar.

Here, the solution to the problem is proposed using sentiment analysis on the user reviews and considering the starred rating. Sentiment analysis will be conducted on the reviews of the users and a numeric rating will be generated from the polarity of the content of the reviews. This step is conducted with the procedure shown by Chen et al. [3]. Their approach was exactly defined for diverse sentiment expressions of the type discussed above and for a definite domain as there are apps of various categories in Google Play Store. The final rating will be the average of the rating done by sentiment analysis and the star rating given by the users. This is will solve the ambiguity and the users do not have to go for reading an elaborative comment. They will get the recommendation from a unique rating done on both types of review.

II. RELATED WORKS

THE sudden eruption of sentiment analysis as well as opinion mining has opened new possibilities to improve our information gathering interests. Scholars throughout the world have worked a lot to understand the opinion of the general people on various topics using the natural language processing and semantic analysis. Most of them were done for the formal writings.

A research of Dave et al. produced a method of extracting the polarity of the user reviews for products [4]. The process returns the polarity of the comments as poor, mixed and good. Again Hu and Liu proposed Feature Based Summarization (FBS), an opinion summarization of products which is categorized by opinion polarity [5][6]. Cheng then also demonstrated an opinion summarization, categorized by product features, in bar graph style [7].

Pang, Lee and Vaithyanathan showed that machine learning approaches are better on traditional topic-based categorization but not fruitful on sentiment classification [8]. Information extraction technologies [9] have also been explored. A statistical model named as Spin model is used for sentiment words but the experiments are discussed in brief by Takamura et al. [10]. Popescu and Etzioni proposed the OPINE system [11] for summarized feature-based sentiment extraction.

A very recent work [12] by Arti et al. has revealed various sentiment analysis methods to summarize a number of comments and reviews. Those methods are based on mathematics and statistics specially Gaussian Distributions. To overcome the problems during sentiment analysis Liu have given showed useful approaches [13]. A great step was taken by Chen et al. which considered the informal writings in this domain of sentiment analysis [3]. They extracted the sentiment expressions from diverse writing corpus and ranked them with polarities.

Much progress has been made in this field of sentiment analysis. However, most of the earlier works are based on summarizing a number of comments or reviews. Some of the important works are done using statistical techniques, based on machine learning or word-correlation algorithms. But combining with separate star rating is not considered widely yet.

III. THE PROPOSED APPROACH

LET us consider **R** is a user review of an app of category **C**. Here the category is needed so that the sentiment expression extraction can be efficient. The parameters of our proposal exactly matches with the approach of Chen et al. [3] as their sentiment expression extracting procedure takes two parameters as this scenario. Here their approach is followed to get the numeric polarity of the review **R** of category **C**. They considered approach is chosen because of the result of various scales like, Precision, Recall, F-measure etc. shows better result in case of diverse sentiment analysis with target dependent polarity.

A. Preprocessing the User Review

At the very beginning of the process a set of root words are to be collected basing on **C**. The root words will be collected from sentiment-lexicons such as SentiWordNet[1], MPQA[2] and General Inquirer[3]. These will be used for the formal statements. Furthermore, for the informal and slang statements we will fetch words from Urban Dictionary[4]. For the extraction of root words, only the words with high scores will be taken from the knowledge bases.

In the next step, we will generate some candidate expressions. To generate the candidate expressions we have to split the review into sentences and words subsequently. The user reviews are not always a full paragraph and sometimes they are not even a complete sentence. So, in this procedure the review will be split in sentences by SentParBreaker[5] and then with the help of Stanford Parser[6] the dependency relations of the words will be extracted from the sentences.

In the user reviews there can be more than one expressions. It may happen that, there lies one positive and one negative expression. In that case it will be considered that the review is inconsistent. Otherwise, for containing maximum number of positive or negative expression determines a review consistence. Let c_1 and c_2 are two candidate expressions in a review. This relation can be found by the following algorithms:

1) Identifying Inconsistency Relations: We know that, a sentiment expression is inconsistent with its negation; two sentiment expression linked by contrasting conjunctions are likely to be inconsistent. By this ideas c_1 and c_2 are identified inconsistent with each other if (1) c_1 is a part of c_2 (but not equal), and c_2 starts with a negation and ends with c_1; or (2) c_1 appears before c_2 (without overlapping), where there is no extra negation applied to them, and they are connected by contrasting conjunctions like as but, although etc. Here the extra negation means that the negation part of c_1 or c_2.

2) Identifying Consistency Relations: c_1 and c_2 are identified consistent with each other if c_1 appears before c_2 (without overlapping between them), and there is no extra negation applied to them or no contrasting conjunction connecting them.

After that two networks are formed where consistent and inconsistent expressions are connected by their respective relations. The consistency network is denoted by $N^{cons}(P, R^{cons})$ and the inconsistency network is denoted

[1]http://sentiwordnet.isti.cnr.it/
[2]http://mpqa.cs.pitt.edu/
[3]http://www.wjh.harvard.edu/ inquirer/
[4]http://www.urbandictionary.com/
[5]http://nlp.stanford.edu/software/lex-parser.shtml
[6]http://text0.mib.man.ac.uk:8080/scottpiao/sent_detector

by $N^{incons}(P, R^{incons})$. Here, P denotes a node set consisting the candidate expressions and R is a set of weighted edges that denotes the respective relations between the candidate expressions.

B. Extracting Numeric Polarity

Instead of extracting polarity directly a *polarity probability* for each candidate expression is determined first. To do this, for each candidate expression c_i, two types of polarity probability will be calculated: P-Probability $Pr^P(c_i)$ that indicates the positive sensitivity and N-Probability $Pr^N(c_i)$ that indicates the negative sensitivity of the candidate expressions. These polarity probability will retain the characteristics such as $Pr^P(c_i) + Pr^N(c_i) = 1$, that is the more like c_i is positive or negative, the less likely it is negative or positive.

Now we can define the consistent and inconsistent relations of two candidate expressions from the P-Probability and N-Probability of the candidate expressions. Let c_i and c_j are two candidate expressions whose *polarity probability* will be independent, their *consistency probability* will be $Pr^P(c_i)Pr^P(c_j) + Pr^N(c_i)Pr^N(c_j)$ and the *inconsistency probability* will be $Pr^P(c_i)Pr^N(c_j) + Pr^N(c_i)Pr^P(c_j)$.

The consistent relation between c_i and c_j indicates that the review contains consistent sentiment, i. e. the expectation of consistent probability is 1. The difference between consistency probability and its expectation can be measured by squared error: $(1 - Pr^P(c_i)Pr^P(c_j) - Pr^N(c_i)Pr^N(c_j))^2$ in the network N^{cons}. Similarly, for the inconsistency network N^{incons}, the difference between inconsistency probability and its expectation is measured by $(1 - Pr^P(c_i)Pr^N(c_j) - Pr^N(c_i)Pr^P(c_j))^2$. Therefore we get the sum of square errors (SSE) in both the networks as:

$$SSE = \sum_{i=1}^{n-1}\sum_{j>i}^{n} (w_{ij}^{cons}(1 - Pr^P(c_i)Pr^P(c_j) - Pr^N(c_i)Pr^N(c_j))^2 + w_{ij}^{incons}(1 - Pr^P(c_i)Pr^N(c_j) - Pr^N(c_i)Pr^P(c_j))^2)$$

where, w_{ij}^{cons} and w_{ij}^{incons} represents the weights of relation between c_i and c_j in the networks N^{cons} and N^{incons} respectively and n is the total number of candidate expressions. Here the SSE is used instead of absolute error so that the two types of relations can not cancel each other.

It is expected that the P-Probabilities and N-Probabilities will minimize the SSE in such a way that the corresponding consistency and inconsistency probabilities will be closest to their expectations suggested by the networks. Now by replacing $Pr^N(c_i)$ with $1 - Pr^P(c_i)$, $Pr^N(c_j)$ with $1 - Pr^P(c_j)$, $Pr^P(c_i)$ with x_i and $Pr^P(c_j)$ with x_j we get the following function:

$$minimize\{\sum_{i=1}^{n-1}\sum_{j>i}^{n}(w_{ij}^{cons}(x_i + x_j - 2x_ix_j)^2 + w_{ij}^{incons}(1 - x_i - x_j + 2x_ix_j)^2)\}$$

where,

$$0 \le x_i, x_j \le 1 \quad for \quad i, j = 1, 2, ..., n$$

Here, the candidate expressions lying in the root set will be assigned with 1 or 0 according to their predefined polarity. The probabilities of other candidate expressions can be found out by the above stated function. For this, the L-BFGS-B[7] algorithm will be used to solve the above problem and the polarities will be generated.

Finally, the candidate expressions will be assigned with a P-Probability and N-Probability. From them only the candidates having probability below the threshold value for both positive and negative will be considered as the result and others will be removed. This P-Probability and N-Probability will determine the positive or negative sentiment of the user reviews. As the polarity gained by this will be in the range between 0 and 1, it will be converted to a rating on a scale between 0 to 5 as the prevailing starred rating on Google Play Store has it already established.

C. Combining the ratings

In the Google Play Store site, a numeric rating is generated from the starred rating. Finally the mean of the starred rating and the numeric rating generated by the above procedure will be the final rating for the apps. This will overcome the problem of ambiguity between the starred rating and the user reviews.

IV. Extracting Rating from User Review

In this paper an approach to the solution of the problem stated earlier is discussed. To demonstrate the flow of operation of this process let us consider the following starred rating and the review of an unanimous user for the app Viber under the category Communication.

Rating: 4
Review: Pathetic, the new update of the emoticons is stupid. I hate the emoticon panda icon opening his face that's really weird, replace that with normal icon.

Here it is seen that there lies a vast difference between the starred rating and the review written by the user. As we see the review is written in an informal way, we used the online NLP tools stated in one of the earlier sections to find out the root words, consistency/inconsistency networks. After that we used our proposed probabilistic model on the generated network to find out the polarity of the review. In

[7] http://www.mini.pw.edu.pl/?mkobos/programs/lbfgsbwrapper/index.html

the mentioned review there are some negative informal words used like as ”Pathetic”, ”stupid” and ”weird”. Because of this the probabilistic outcome of the review was 0.135. Finally the user starred rating was normalised to 0.8. Averaging the two numeric values we generate the final rating 0.4675 which is of a below average standard.

I we focus only on the mentioned review we can conclude that the particular user is not happy with the app but if we focus only on the starred rating other users may be deceived from the actual sentiment of the user. However, the combined rating minimizes the degree of difference between the two different ratings posted by the user.

V. Future Works

THE whole proposal for the desired result is given on the theoretical and practical works done before. This procedure is not implemented on a noteworthy number of data so the efficiency can not be measured now due to lack of time and resources. A study can be done on the theoretical results of the procedure which will be done on a sound amount of user reviews. Furthermore, the procedure can be modified or changed with other efficient algorithm to get the desired result.

VI. Conclusion

IN this paper, a procedure to get unified user rating from the written reviews and starred rating is described based on another optimization-based approach for extracting diverse sentiment expressions. To the best of my knowledge, this procedure of unifying rating has not been considered for studies earlier. Earlier it was limited to extracting polarity only. Mostly the area of formal writings were taken under considerations. Recently a number of works has been done for the informal reviews and blogs. According to the authors of the approach we followed it is efficient for the diverse sentiment expressions of definite domains. As the user reviews of apps vary from category to category, the proposed procedure is efficient from that point of view.

VII. Acknowledgements

THIS work is done as the final assignment of the Research Methodology course under the guidance of Dr. Khandaker Tabin Hasan.

References

[1] A. Maas, R. Daly, P. Pham, and D. Huang, “Learning word vectors for sentiment analysis,” *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002491

[2] J. Jong, “Predicting Rating with Sentiment Analysis,” pp. 1–5, 2011. [Online]. Available: http://cs229.stanford.edu/proj2011/Jong-PredictingRatingwithSentimentAnalysis.pdf

[3] L. Chen, W. Wang, M. Nagarajan, S. Wang, and A. Sheth, “Extracting Diverse Sentiment Expressions with Target-Dependent Polarity from Twitter.” *ICWSM*, vol. 2, no. 3, pp. 50–57, 2012.

[4] K. Dave, S. Lawrence, and D. Pennock, “Mining the peanut gallery: Opinion extraction and semantic classification of product reviews,” *Proceedings of the 12th international conference on World Wide Web*, pp. 519–528, 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=775226

[5] M. Hu and B. Liu, “Mining and summarizing customer reviews,” *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1014073

[6] ——, “Mining opinion features in customer reviews,” *AAAI*, pp. 755–760, 2004. [Online]. Available: http://www.aaai.org/Papers/AAAI/2004/AAAI04-119.pdf

[7] L. Ku, T. Wu, L. Lee, and H. Chen, “Construction of an evaluation corpus for opinion extraction,” *Proc. of the Fifth NTCIR Wksp. on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access*, pp. 513–520, 2005. [Online]. Available: http://nlg.csie.ntu.edu.tw/conference_papers/ntcir2005a.pdf

[8] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, vol. 10, pp. 79–86, 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=1118704

[9] C. Cardie, J. Wiebe, T. Wilson, and D. Litman, “Combining Low-Level and Summary Representations of Opinions for Multi-Perspective Question Answering.” *New directions in question answering*, pp. 20–27, 2003. [Online]. Available: http://www.aaai.org/Papers/Symposia/Spring/2003/SS-03-07/SS03-07-004.pdf

[10] H. Takamura, T. Inui, and M. Okumura, “Extracting semantic orientations of words using spin model,” *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics*, no. June, pp. 133–140, 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1219857

[11] A. Popescu and O. Etzioni, “Extracting product features and opinions from reviews,” *Natural language processing and text mining*, no. April, pp. 19–38, 2007. [Online]. Available: http://link.springer.com/chapter/10.1007/978-1-84628-754-1_2

[12] A. Buche, D. Chandak, and A. Zadgaonkar, “Opinion Mining and Analysis: A survey,” *International Journal on Natural Language Computing*, vol. 2, no. 3, pp. 39–48, 2013. [Online]. Available: http://arxiv.org/abs/1307.3336

[13] B. Liu, “Sentiment analysis and subjectivity,” *Handbook of natural language processing*, vol. 2, p. 568, 2010. [Online]. Available: http://people.sabanciuniv.edu/berrin/proj102/1-BLiu-Sentiment Analysis and Subjectivity-NLPHandbook-2010.pdf