Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 12 (2017) pp. 146-153
© MIR Labs, www.mirlabs.net/jias/index.html

Opinion Mining to Assist User Acceptance Testing for Open-Beta Versions

Akshi Kumar[1], Ajith Abraham[2]

[1] Dept. of Computer Science & Engineering, Delhi Technological University,
Delhi, India
akshikumar@dce.ac.in

[2] Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence,
WA, USA
ajith.abraham@ieee.org

***Abstract*: The collaborative and participative facets of both Software Development and Web 2.0 has compelled researchers and practitioners to probe the integration among these two distinct, evolving areas of study. Opinion mining as a subtask of text mining, automatically extracts knowledge from loosely unstructured and often ungoverned human-sourced big-data information available through Social Networks. This paper proposed a model for mining the opinion with a deeper emotional implication that can assist as a supporting tool for User Acceptance Testing. The idea is to gauge the acceptance of an open-beta release version of software by initially extracting the opinion from tweets and consequently assessing finer-grained levels of emotions using a hybrid approach (lexicon + machine learning) that can quantify acceptance criteria attributes such as usability. The tool has been implemented & evaluated for supervised machine learning variants, namely Naives Bayesian, Multinomial, Gaussian and Bernoulli Naives Bayesian along with Support Vector Machine. The effectiveness of the proposed tool is presented with a sample set of tweets based on a case study and initial results demonstrate that it is a motivating technique to test the business objective of the system developed.**

Keywords: Opinion Mining; Acceptance Testing; Open-beta software; Tweet, Naïve Bayesian, SVM

I. Introduction

Recent years have seen an escalating attention towards using the social facet of Web as a supporting technology for superior software development practices. Moreover, the SMAC (Social Media, Mobile, Analytics and Cloud) paradigm [1] has been globally accepted as a trending software development technology by both researchers & practitioners. Pertinent studies have shown the extent of using social web for collaborative, focused and strategic development. Most of the work reported till date is to mine developer's sentiment. The research has focused on improving development, maintenance and evolution of software by applying opinion mining on social web applications like twitter [2], app reviews [3], question answering systems such as Stack Overflow [4, 5] and collaborative development platforms such as GitHub [6, 7,8] & Jira [9,10]. The work presented in this paper depicts the novel use of opinion mining for software testing.

Testing is a critically essential, continuous process that primarily measures the quality of the software system, amongst others. User Acceptance Testing (UAT), also known as beta testing, application testing, and/or end user testing is a discrete and definite way to test the business functionality of the system developed. Formally, it is a phase of software development in which the software is tested in the "real world" by the intended audience or a business representative [11]. The merits that UAT carries to positively and effectively impact a project's success are abundant. These are but not limited to, reducing the cost of system development and ownership, ensuring that the system behaves exactly as expected and exploiting loyalty and word-of-mouth market share. Figure 1 illustrates Testing in the Software Development Lifecycle (SDLC).

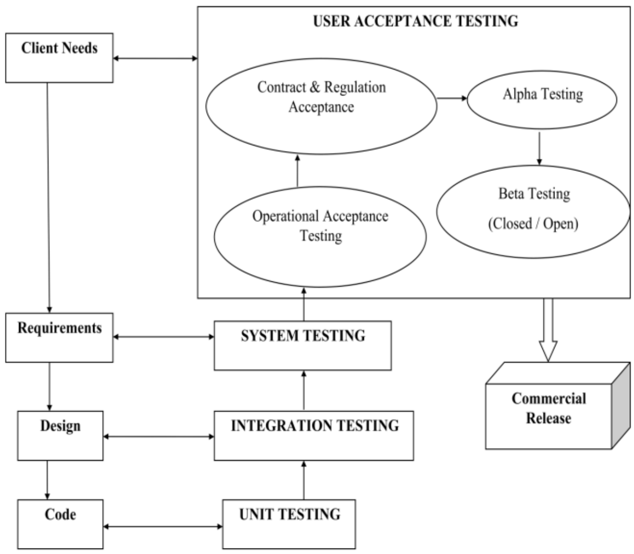

Figure 1. Testing in SDLC

There are two very distinct types of beta testing, namely, private beta (a.k.a "closed") and public beta (a.k.a "open").

For all intents and purposes, Public Beta means test-driving a pre-release software to validate it fit for purpose. They are open to any user who wishes to get involved and are often used to endorse and perfect software, website or video games. Public Beta tests usually follow post an extensive private beta testing. Table 1 illustrates the difference between Closed and Open Beta Testing.

Attribute	Closed (Private) Beta	Open (Public) Beta
Goal	Bugs, performance, accuracy, acceptance.	Marketing, Buzz, Business Intelligence
Access	Selected group of users that match the product's target market.	Available to the general public
Participants	Dozens to hundreds.	Thousands to millions
Feedback Type	Bugs, features, suggestions, tasks, surveys, forums	User Experience, Focus on trials
Product Type	All technology products	Online games, websites, apps
Tester Qualification	Extensive Qualification, often demographic	Little/ no requirements
Duration	Fixed time (weeks to months)	Open length of time until release

Table 1. Difference between Closed and Open Beta Testing

News apropos Apple's iOS 11 public beta for iPhone and iPad being available to everyone for download has been going rounds since August 2017. This allows users who are not registered developers to test pre-release versions of iOS with new features for free. According to CNET [12], a leading consumer technology reviewer, "The iOS 11 public beta is out, months ahead of its official release, giving iPhone and iDevice users their first taste of Apple's new operating system and a chance to locate and report bugs. Prior to the public beta availability, iOS 11 has only been available to test with a $99/year developer account [13]. In September 2017, Firefox 57 Beta 'Quantum' With Next-Generation Browser Engine was released [14] with the primary goal of testing about-to-be-released features in the most stable pre-release build. Thus the purpose of open-beta testing is to generate awareness and buzz about the product, rather than an actionable feedback from the "testers" [15]. The evolving Software development, testing and maintenance practices foster the industry and researchers to look for intelligent supporting technologies and tools that can help improve and assist the user acceptance criteria for business goal conformance. With advancements in web technology, participation and communication have been acknowledged as two key attributes that facilitate uncovering opinions within the vast pool of people. More specifically, the advent of real-time, social networking sites like Twitter, Facebook have instigated the creation of an unparallel public collection of opinions about every object of interest [16]. The term opinion mining was first noticed in a paper by Dave et al. [17] The paper defined that an opinion mining tool would ―process a set of search results for a given item, generating a list of product attributes (quality, features, etc.) and aggregating opinions about each of them (poor, mixed, good).

A rising trend to exploit opinion mining for business intelligence is well recognized [18]. We examine the strategic alliance of opinion mining to Software Testing, specifically User Acceptance Testing (UAT), where the former can be used as a supporting tool to gauge the acceptance of a beta release version of software. We propose a framework for extracting and scoring opinion markers (adjectives along with the verbs and adverbs) using a hybrid approach to consequently find the emotion values of the tweets. The idea is to mine the public mood and opinion with a deeper emotional implication enumerated by fixed set (happiness, anger, sadness, fear and disgust) that can quantify acceptance criteria attributes such as usability, understand ability etc. which characterize quality, for an open-beta version of the software, thus testing the business goals intelligently.

II. Related Work

Applying opinion mining on Web 2.0 has been a potential direction of research with scientific trials and promising applications been explored substantially. Approaches to mine the opinion are notably divided into lexicon-based and machine-learning based. The machine learning-based approach typically trains sentiment classifiers using features. Unlike machine learning methods, Lexicon-based methods, do not necessitate labeled training data but require readily available dictionaries with initial sentiment polarity values. These can either be generated manually, where polarity values are human-assigned or determined semi-automatically, where either corpus-based (uses collection of documents) or Dictionary-based methods (uses machine-readable dictionaries) are used. A state-of-art on past, present and future of Sentiment Analysis, has been examined critically in the survey presented by Kumar and Sebastian [18].

The collaborative expansion of software engineering practices and simultaneous emergence and evolution of collaborative Web makes the alliance, adaption and adoption between the two apparent. Literature exploring this union is very little and further endorses research in this direction. Storey et al. [19] advocate the impact of social media on software engineering practices and tools. Dehkharghani and [20] used semi-automatic approach of identifying quality attributes such as security, reliability and user-friendliness of software product using sentiment analysis of tweets used in micro-blogging site twitter. The proposed approach reduced the human effort required to capture feedback from the user about quality of software product. El-Halees [21] used opinion mining to automatically evaluate software's subjective usability using survey papers through the use of interviews and questionnaires. His model focused on three aspects of usability namely effectiveness, efficiency, and satisfaction. The proposed model achieved accuracy of 85.41%. The experiment further shows that precision, recall and f-measure of the evaluated reviews are acceptable. Kim et al. [22] used opinion mining on the tweets of micro-blogging site twitter to examine which factors could affect the user's interest or preferences by analyzing and comparing smartphone reviews. Selvan and Moh [23] presented a framework for fast feedback opinion mining on twitter data streams and have shown 84% accuracy in sentimental analysis. Their framework reduced the human effort required to know the feedback to companies/ organization about software product. Jurado and Rodriguez [24] proposed Sentiment Analysis techniques to identify and monitor the underlying sentiments in the text written by developers analyzing GitHub's project issues. Guzman et al. [7, 8] have applied sentiment analysis to the content available in collaborative development environments such as GitHub. More recently, Goyal and Sardana [25] proposed a Sentiment Based Model for Predicting the Fixability of Non-Reproducible Bugs for software maintenance. In 2017, Islam & Zibran [26] developed SentiStrength-SE, a tool for improved sentiment analysis especially designed for application in the software engineering domain. Calefato et.al [27] also have put forward a novel Senti4SD classifier specifically trained to support sentiment analysis in developers' communication channels that is to study software developers' emotions by mining crowd-generated content within social software engineering tools.

To the best of our knowledge, there is no research probing the use of opinion mining to assist software testing, more distinctively User Acceptance Testing. Acceptance criteria is defined on the basis of the attributes such as functional correctness and completeness, data integrity, data conversion, usability, performance, timeliness, confidentiality and availability, installability and upgradability, scalability, and documentation [28]. Thus, we probe a novel dimension of research to mine the opinion to gauge the acceptance of an open-beta release version of software that can assist as a supporting tool for User Acceptance Testing.

III. Opinion Mining Model for Open- Beta Versions

Opinion mining on Twitter has been trending in both research and practice. We propose a model to determine the emotion value of the tweet that can quantify acceptance criteria attributes such as usability for an Open –Beta Software. The idea is to test the business objective of the system developed by acquiring feedback based on the results of applying opinion analytics to twitter data. Figure 2 depicts the proposed research model.

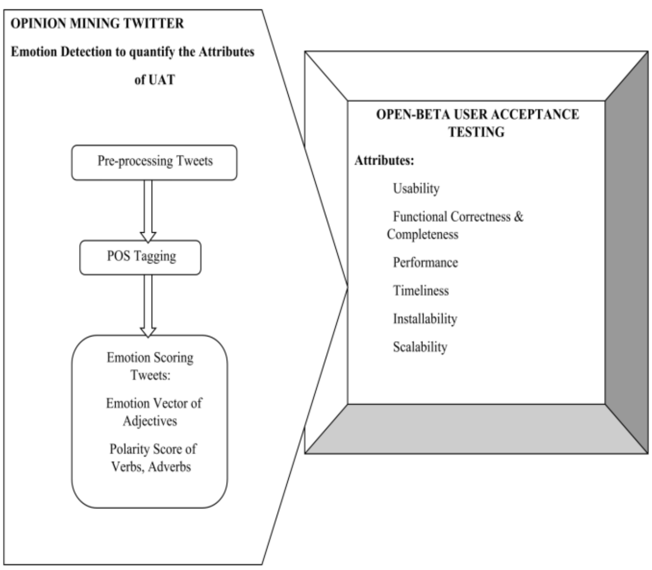

Figure 2. The Proposed Model

The following subsections expound the details:

A. Pre-processing Module

The objective of the pre-processing module is to prepare the desired transaction file that contains opinion markers, distinctively the adjective, adverb and verb. In this process the desired data is acquired from the publically available Twitter datasets by using Twitter API by cleaning it for extracting the features. The cleaning procedure comprises of removal of hashtags, hyperlinks, usernames, punctuations, non-English words and emoticon replacement, amongst others. The extracted terms are next put to the Part-of-Speech (POS) tagger, where only the opinion markers (adjective, adverb, verb) are retained, discarding everything else.

B. Emotion Scoring of Tweets

Once the transaction file with the opinion markers, namely the adjective, adverb and verb is ready, the next step is to quantify their scores to calculate the emotional value of each tweet. That is, once the POS tagging is done, the words are scored either individually (only an adjective) or as a group (adverbs or verbs followed by an adjective) [29].

We follow a two-step procedure where each adjective is associated with an emotion vector, whose values are determined using a hybrid technique (corpus + machine learning-based) and Adverb-Verb Polarity scores are estimated using a dictionary-based method. Accordingly, in the first step we determine the score values of 5 basic emotions namely, Happiness, Anger, Sadness, Fear and Disgust, represented as vector, using a hybrid approach which is a combination of lexicon (corpus-based) and machine learning method. The emotion values of terms are initially assigned manually and then used as features to train the classifier using WEKA Tool [30].

We implement & compare the Naïve Bayesian Classifier, its 3 variants, namely Multinomial Naïve Bayesian, Bernoulli Naïve Bayesian and Gaussian Naïve Bayesian along with Support Vector Machine. Naïve Bayesian classifier (NB) is the simplest probabilistic classifier to use and is implemented using a bag-of-words approach for opinion mining. Bernoulli Naive Bayesian (BNB) classifier is typically used when the absence of a particular word matters. The Multinomial Naive Bayesian (MNB) is suitable for classification with discrete features & is typically used when occurrence of the word matters more than the frequency. The two variants of MNB were tested, firstly binarized MNB where it has binary weighting function such that the value 1 means that the word occurs in the particular document, and 0 means that the word does not occur in this document and secondly a tf-idf MNB Classifier is used where the tf-idf approach assumes that the importance of a word is inversely proportional to how often it occurs across all documents. The Gaussian Naïve Bayes (GNB) & support vector machine (SVM) were also used for a comparative performance analysis.

The initial file (seed-list) of adjectives is created by conducting an online survey among 350 Undergraduate students to score the emotions on a scale of 0 to 5. The file included 1000 adjectives with respective emotion score vectors; the snapshot is shown in Table 2.

Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 12 (2017) pp. 146-153
© MIR Labs, www.mirlabs.net/jias/index.html

WORD	HAPPINESS	ANGER	SAD-NESS	FEAR	DISGUST
Damaging	1.33	3.5	3.06	2.73	2.42
Dirty	1.28	2.3	1.94	1.94	3.7
Easy	3.92	1.11	1.15	1.19	1.09
Ecstatic	4.08	1.34	1.31	1.8	1.52
Elated	3.93	1.22	1.19	1.17	1.12
Famous	3.32	1.3	1.21	1.2	1.38
Fantastic	4.07	1.19	1.31	1.25	1.22
Greedy	1.41	3.14	2.68	2.27	2.94
Hard	1.65	2.22	1.75	2.21	1.40
Innocent	3.17	1.37	1.49	1.66	1.27
Lady	1.49	2.01	1.83	1.40	2.39
Menacing	1.17	2.94	1.78	1.97	2.18
Merry	4.38	1.07	1.14	1.08	1.08
Noisy	1.39	2.97	1.39	1.41	1.45
Nonchalant	1.85	1.40	1.31	1.26	1.47
Protected	4.11	1.24	1.33	1.47	1.08
Proud	3.18	1.55	1.29	1.58	1.26
Quartan	1.39	1.18	1.17	1.17	1.15
Rejected	1.05	3.50	3.91	3.47	2.00
Relaxed	4.32	1.12	1.14	1.10	1.04
Scared	1.14	2.41	3.02	4.09	1.83
Scornful	1.16	3.31	2.13	2.17	1.74
Serious	1.45	1.92	1.84	1.97	1.29

Table 2. Sample adjective Emotion vectors

Next, to compute the opinion polarity score for adverbs and verbs, the seed lists of positive and negative adverbs and verbs whose orientation we know is created and then grown by searching in WordNet [31], thus using a dictionary-based approach. The use and effect of adverb/verb in opinion mining has been studied extensively in research [16, 18, 29]. Adverbs, such as 'not', completely change the polarity of tweet and adverbs like 'hopelessly', 'seriously' intensify the effect making them imperative to our work as this leads to intensification of the emotions. Also, some verbs like 'appreciate', 'cry', 'yell', 'like' convey opinions and are important in depicting the opinion polarity of the tweet. We consider the same values considered by Kumar and Sebastian [29], assigned to the most frequently used verbs and adverbs depicted in Table 3.

ADVERB	STRENGHTH	VERB	STRENGHTH
Complete	-1	Love	1
Most	0.9	adore	0.9
Totally	0.8	like	0.8
Extremely	0.7	enjoy	0.7
Too	0.6	smile	0.6
Very	0.4	impress	0.5
Pretty	0.3	attract	0.4
More	0.2	excite	0.3
Much	0.1	relax	0.2
Any	-0.2	reject	-0.2
Quite	-0.3	disgust	-0.3
Little	-0.4	suffer	-0.4
Less	-0.6	dislike	-0.7
Not	-0.8	detest	-0.8
Never	-0.9	suck	-0.9
Hardly	-1	hate	-1

Table 3. Sample Adjective Emotion vectors

Finally, we calculate the Emotion score of the Tweet based on the notion that presence of a verb or adverb pre to any adjective, amplifies the emotion of that adjective. The algorithm for the same is as follow:

If an adverb or verb is encountered after another adverb or verb then we add it. We repeat this process till an adjective is encountered.

Now if the added value of adverbs or verbs is less than 0 i.e., negative, then for the upcoming adjective, we subtract its value from 5.

Or if added value of verbs or adverbs is positive and >= 0.5 then multiply it with the upcoming adjective else multiply 0.5 with upcoming adjective.

Later these multiplication results are added and the sum is divided by 5 times the number of adjectives encountered.

The following section demonstrates the implementation and empirical analysis of the proposed model with examples of recent open-beta versions of software available for public.

IV. Implementation and Analysis

To clearly illustrate the effectiveness of the proposed model, a case study is offered with a sample set of tweets.

A. Pre-processing of Tweets

To get large publically available Twitter datasets, we use Twitter Application Program Interface (API). After downloading tweets, we use the pre-processing module for data-cleaning, i.e., to primarily remove hashtags(#), usernames(@), hyperlinks, ReTweet (RT) symbol, punctuations and non-English characters. For removal of punctuations and non-English characters, we process the emoticons using Emoji Unicode Table. Next using the Natural language (NL) Processor linguistic Parser, we tag the adjectives, verbs and adverbs. The resultant file is a list of tweets that only has adjectives, verbs and adverbs (in the original order), which are referred to as opinion markers. For example, a tweet: "Firefox 57.0 Beta 4 is pretty impressive! #Firefox57 #ProjectQuantum" is converted to "pretty impressive".

B. Emotion Scoring

Once the POS tagging is done, the words are scored either individually (only an adjective) or as a group (adverbs or verbs followed by an adjective). For example: the above POS tagged tweet is scored as follows:

S. No	TWEET	HAPPINESS	ANGER	SADNESS	FEAR	DISGUST
1.	*Great news! #Apple #iOS11 #iOS11Beta*	0.936	0.216	0.234	0.222	0.221
2.	*Life's been SLOW ever since updating my phone to #iOS11 #iOS11Beta*	0.214	0.798	0.394	0.212	0.196
3.	*iOS 11 feels great. #iOS11Beta #iOS11*	0.812	0.226	0.211	0.201	0.191
4.	*This #iOS11 update was the worst thing to ever happen to my phone. #iOS11Beta*	0.189	0.822	0.398	0.215	0.196
5.	*Super fast on my old MacBook Air 2013. Beats chrome in speed and has lower memory usage! Been using for the last 2 days. Love it! Excited #Firefox57,*	0.912	0.251	0.226	0.272	0.226
6.	*Gotta admit its damn good! #Firefox57, #ProjectQuantum*	0.584	0.110	0.112	0.119	0.121
7.	*Firefox 57.0 Beta 4 is pretty impressive! #Firefox57 #ProjectQuantum*	0.449	0.115	0.115	0.128	0.119

a. Here we can see that "pretty" is an adverb and "impressive" is an adjective.
b. The adjective emotion values of "impressive" are represented by a vector [4.49, 1.15, 1.15, 1.28, 1.19] ([<Happiness>, <Anger>, <Sadness>, <Fear>, <Disgust>])
c. In the list of adverbs we get the values of "pretty" as 0.3
d. Now using the algorithm defined in Section 3 for calculating the Emotion score of the Tweet, since the sum of adverbs is 0.3, the emotion values are multiplied by 0.5. And then these values are divided by 5 (only 1 adjective).
e. The resultant score of the tweet is given by vector [0.449, 0.115, 0.115, 0.128, 0.119]

We repeat this for each adverbs/verbs - adjective group obtained and then the tweet is scored by averaging the results. Finally, to get a tally of the respective hashtag/username score of each tweet is averaged. We applied our approach to approx 5,000 tweets for various hashtags/users relating to open-beta versions of software, for example #Firefox57, #ProjectQuantum, etc. A sample set of 7 tweets with the result of emotion analysis is depicted in Table 4.

Table 4 Sample Tweet Emotion Values

Figures 3 and 4 show the snapshots of the sample test hashtag *#iOS11Beta* for 347 tweets and *#Firefox57* for 1000 tweets, followed by Figures 5 and 6 that represent the bar graph and spider chart plots of emotion values calculated displaying the comparisons for both the hashtags.

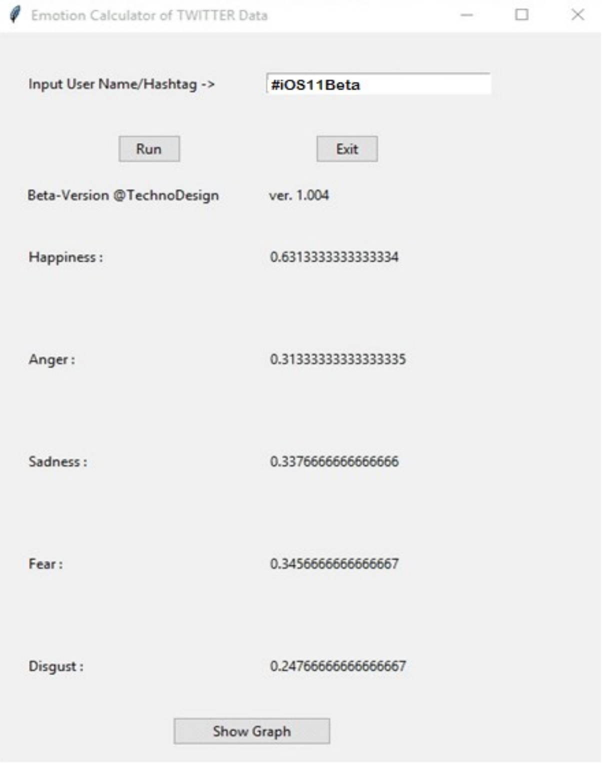

Figure 3. For 347 Tweets with hashtag *#iOS11Beta*

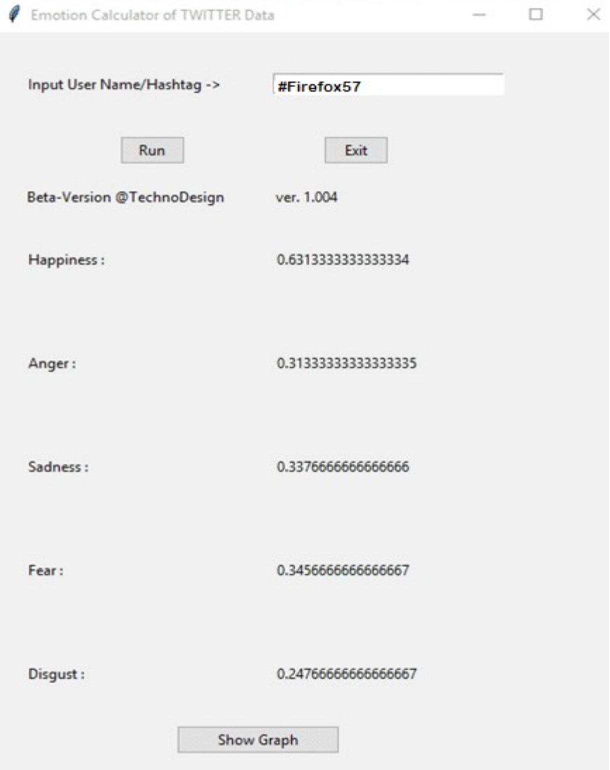

Figure 4. For 1000 Tweets with hashtag *#Firefox57*

Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 12 (2017) pp. 146-153
© MIR Labs, www.mirlabs.net/jias/index.html

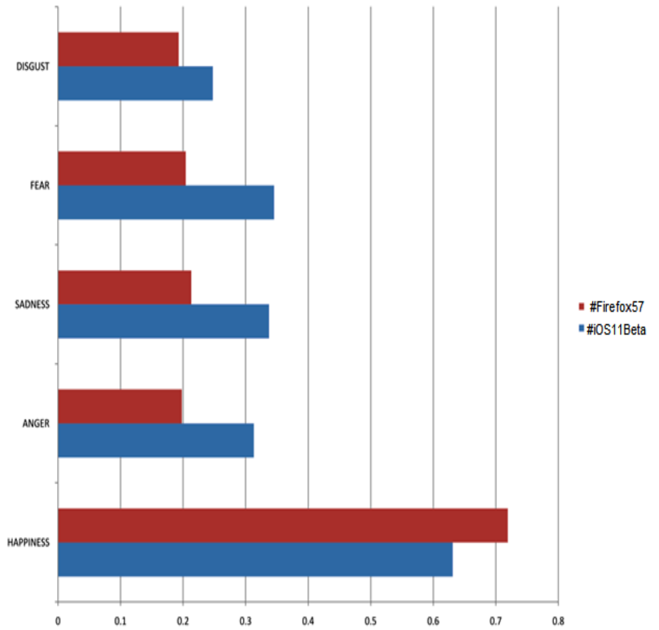

Figure 5 Emotion Value Bar graph for hashtag *#Firefox57 vs# iOS11Beta*

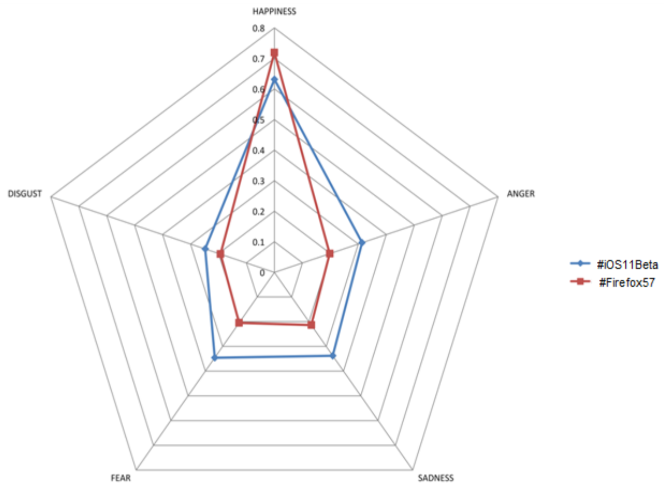

Figure 6 Emotion Value Spider Chart for hashtag *#Firefox57 vs # iOS11Beta*

C. Performance of Opinion Polarity Classifier

The Naïve Bayesian Classifier, and its Multinomial, Bernoulli and Gaussian variants, along with Support Vector Machine were used. The following table 5 depicts the performance analysis on the basis of accuracy and time for these supervised machine learning classifiers implemented.

ALGORITHM	ACCURACY (%)	TIME(SEC)
NB	78.670	0.1563
MNB	82.744	0.0311
BNB	79.657	0.0463
GNB	65.643	0.0154
SVM	76.333	0.031

Table 5. Analysis of Opinion Polarity Classifier

The Multinomial Naive Bayesian Classifier gave the highest accuracy and Gaussian Naïve Bayesian gave the lowest accuracy. The performance of Gaussian Naïve Bayesian was compromised as our dataset was discrete whereas it works well for continuous numerical data. The Bar graphs are shown for the same in Figures 7 & 8.

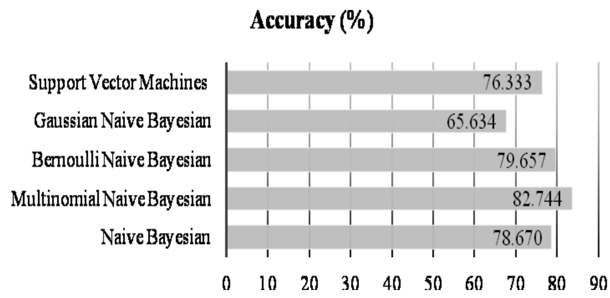

Figure 7. Accuracy Analysis of Opinion Polarity Classifier

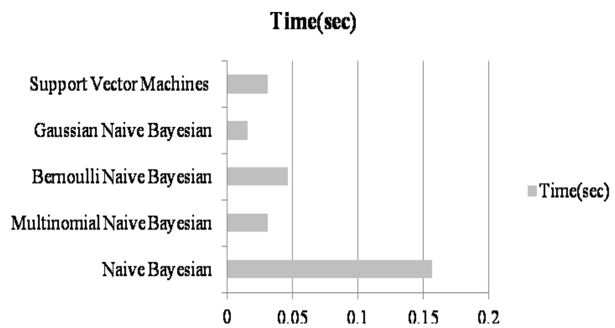

Figure 8. Time Analysis of Opinion Polarity Classifier

V. Conclusions

Despite ample research on opinion mining for marketing, governance, business intelligence, few have explored the role and effect of opinion and more explicitly emotion in software development phases. This research introduced a model that exploits opinion mining at a finer-grain level of emotions to ascertain user acceptance criteria for open-beta testing. A hybrid technique has been proposed to perform emotion analysis by finding opinion markers in tweets related to Public-beta software versions. The variants of Naïve Bayesian (Gaussian, Multinomial, and Bernoulli) along with Support Vector Machine were examined. The preliminary evaluation shows this approach as a promising one for business objective attainment as it helps to capture the attributes that characterize User Acceptance Testing for open-betas. As probable extension to this work, we would like to introduce and augment a summarization tool for coherent summary of generated opinions/emotions to users enabling public-beta product review. Further, we would also like to evaluate for other classification algorithms such as decision trees and ensemble methods (Bagging & Boosting).

Acknowledgment

We would like to thank Vikrant Dabas, Graduate Student, Department of Computer Science, College of Computing and Informatics, University of North Carolina, Charlotte, USA for his assistance with the statistics used in this paper.

References

[1] H. I. Alfouzan, "Introduction to SMAC-Social Mobile Analytics and Cloud", *International Journal of Scientific & Engineering Research*, vol. 6, pp. 128-130, Sept. 2015.

[2] Guzman E, Alkadhi R, Seyff N, "A needle in a haystack: What do twitter users say about software?" In *24th IEEE International Requirements Engineering Conference In: Proceedings of the IEEE 24th International Requirements Engineering Conference (RE)*, pp. 96-105, 2016.

[3] Maalej W, Kurtanovic Z, Nabil H, Stanik C, "*On the automatic classification of app reviews. Requirements Engineering*" 21(3):311–331, DOI 10.1007/s00766-016-0251-9 , 2016.

[4] Rahman MM, Roy CK, Keivanloo, "I Recommending insightful comments for source code using crowdsourced knowledge", In *15th IEEE International Working Conference on Source Code Analysis and Manipulation*, SCAM 2015, Bremen, Germany, September 27-28, 2015, pp 81–90, 2015.

[5] Calefato F, Lanubile F, Marasciulo MC, Novielli N , "Mining successful answers in stack overflow", In *Proceedings of the 12th Working Conference on Mining Software Repositories*, IEEE Press, Piscataway, NJ, USA, MSR '15, pp 430–433, 2015.

[6] Sinha V, Lazar A, Sharif B, " Analyzing developer sentiment in commit logs", In *Proceedings of the 13th International Conference on Mining Software Repositories*, ACM, New York, NY, USA, MSR '16, pp 520–523, 2016.

[7] Guzman E, Azocar D, Li Y, "Sentiment analysis of commit comments in Github: An empirical study", In *Proceedings of the 11th Working Conference on Mining Software Repositories, ACM*, New York, NY, USA, MSR 2014, pp 352–355, 2014.

[8] Guzman E, Bruegge B, "Towards emotional awareness in software development teams", In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, ACM*, New York, NY, USA, ESEC/FSE 2013, pp 671–674, DOI 10.1145/2491411.2494578, 2013.

[9] Mäntylä M, Adams B, Destefanis G, Graziotin D, Ortu M, "Mining valence, arousal, and dominance: Possibilities for detecting burnout and productivity?", In *Proceedings of the 13th International Conference on Mining Software Repositories, ACM*, New York, NY, USA, MSR '16, pp 247–258, 2016.

[10] Ortu M, Murgia A, Destefanis G, Tourani P, Tonelli R, Marchesi M, Adams B, "The emotional side of software developers in Jira", In *Proceedings of the 13th International Conference on Mining Software Repositories, ACM*, New York, NY, USA, MSR '16, pp 480–483, 2016.

[11] Bordo V, *Overview of User Acceptance Testing (UAT) for Business Analysts (BAs),* https://www.develop.com/useracceptancetests

[12] Apple iOS11 News, www.cnet.com

[13] iOS 11 - Apple (IN): https://www.apple.com/in/ios/ios-11/

[14] *Firefox Quantum 57 for developers:* https://developer.mozilla.org/en-US/Firefox/Releases/57

[15] Perino J, 2014, *Private & Public Betas: What's the Difference*, http://www.betabound.com/private-beta-tests-vs-public-beta-tests-whats-difference

[16] Kumar, A., Dogra, P. and Dabas, V., "Emotion analysis of Twitter using opinion mining". In *Contemporary Computing (IC3),* 2015 Eighth International Conference on (pp. 285-290). IEEE, 2015, August.

[17] Dave, K., Lawrence, S. and Pennock, D.M., "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews". In *Proceedings of the 12th international conference on World Wide Web* (pp. 519-528). ACM, 2003, May.

[18] Kumar, A. and Teeja, M.S., "Sentiment analysis: A perspective on its past, present and future". In *International Journal of Intelligent Systems and Applications*, 4(10), p.1, 2012.

[19] Storey, M.A., Treude, C., van Deursen, A. and Cheng, L.T., "The impact of social media on software engineering practices and tools". In *Proceedings of the FSE/SDP workshop on Future of software engineering research* (pp. 359-364). ACM, 2010, November.

[20] Dehkharghani, R. and Yilmaz, C., "Automatically identifying a software product's quality attributes through sentiment analysis of tweets". In *Natural Language Analysis in Software Engineering (NaturaLiSE),* 2013 1st International Workshop on (pp. 25-30). IEEE, 2013, May.

[21] El-Halees, A.M., "Software usability evaluation using opinion mining". In *Journal of Software*, 9(2), pp.343-349, 2014.

[22] Kim, J., Choi, D., Hwang, M. and Kim, P., "Analysis on smartphone related twitter reviews by using opinion mining techniques". In *Advanced Approaches to Intelligent Information and Database Systems* (pp. 205-212). Springer International Publishing, 2014.

[23] Selvan, L.G.S. and Moh, T.S., "A framework for fast-feedback opinion mining on Twitter data streams". In *Collaboration Technologies and Systems (CTS)*, 2015 International Conference on (pp. 314-318). IEEE, 2015, June.

[24] Francisco Jurado and Pilar Rodriguez, "Sentiment Analysis in monitoring software development processes". J. Syst. Softw. 104, C (June 2015), 82-89, 2015.

[25] Anjali Goyal, Neetu Sardana, "NRFixer: Sentiment Based Model for Predicting the Fixability of Non-Reproducible Bugs", In *e-Informatica Software Engineering Journal,* vol. 11, iss. 1, pp. 103-116, 2017.

[26] Md Rakibul Islam and Minhaz F. Zibran, "Leveraging automated sentiment analysis in software engineering". In *Proceedings of the 14th International Conference on Mining Software Repositories* (MSR '17). IEEE Press, Piscataway, NJ, USA, 203-214, 2017.

[27] Calefato, Fabio, Lanubile, Filippo Maiorano, Federico and Novielli, Nicole. 2017, "Sentiment Polarity Detection for Software Development", *Empirical Software Engineering, Springer.*

[28] Acceptance Testing, http://www.tutorialspoint.com/software_testing_dictionary /acceptance_testing.htm

[29] Kumar, A. and Sebastian, T.M., " Sentiment analysis on twitter". In *IJCSI International Journal of Computer Science Issues*, 9(4), pp.372-373, 2012.

[30] Holmes, G., Donkin, A. and Witten, I.H., "Weka: A machine learning workbench". In *Intelligent Information Systems*, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on (pp. 357-361). IEEE, 1994, December.

[31] What is WordNet, http://wordnet.princeton.edu.

Author Biographies

Akshi Kumar is an Assistant Professor in the Department of Computer Science & Engineering at Delhi Technological University (formerly Delhi College of Engineering). She has received her doctorate degree in Computer Engineering from the University of Delhi. She has received her M.Tech (Master of Technology) and BE (Bachelor of Engineering) degrees in Computer Engineering. She has many publications to her credit in various International Journals with high impact factor and International Conferences with best paper awards. Her research interests include Intelligent Systems, Text Mining, Sentiment Analysis, Recommender Systems, Social & Semantic Web, User-generated Big-data. She has authored books & co-authored book-chapters within her domain of interest. She is a member of IEEE, IEEE (WIE) and ACM and a life member of IACSIT, IAENG, ISTE and CSI.

Ajith Abraham is the Director of Machine Intelligence Research Labs (MIR Labs), a Not-for-Profit Scientific Network for Innovation and Research Excellence connecting Industry and Academia. The Network with Headquarters in Seattle, USA has currently more than 1,000 scientific members from over 100 countries. As an Investigator / Co-Investigator, he has won research grants worth over 100+ Million US$ from Australia, USA, EU, Italy, Czech Republic, France, Malaysia and China. Dr. Abraham works in a multi-disciplinary environment involving machine intelligence, cyber-physical systems, Internet of things, network security, sensor networks, Web intelligence, Web services, data mining and applied to various real world problems. In these areas he has authored / coauthored more than 1,000+ research publications out of which there are 100+ books covering various aspects of Computer Science. One of his books was translated to Japanese and few other articles were translated to Russian and Chinese. About 900+ publications are indexed by Scopus and over 700+ are indexed by Thomson ISI Web of Science. Some of the articles are available in the ScienceDirect Top 25 hottest articles. He has 700+ co-authors originating from 40+ countries. Dr. Abraham has more than 27,000+ academic citations (h-index of 78 as per google scholar). He has given more than 100 plenary lectures and conference tutorials (in 20+ countries). For his research, he has won seven best paper awards at prestigious International conferences held in Belgium, Canada Bahrain, Czech Republic, China and India. Since 2008, Dr. Abraham is the Chair of IEEE Systems Man and Cybernetics Society Technical Committee on Soft Computing (which has over 200+ members) and served as a Distinguished Lecturer of IEEE Computer Society representing Europe (2011-2013). Under his direct academic supervision, 12 students received Ph.D. degrees and is currently supervising 12 Ph.D. students in different Universities in Europe, USA, Africa and India. He has examined over 100 Ph.D. theses. Currently Dr. Abraham is the editor-in-chief of Engineering Applications of Artificial Intelligence (EAAI) and serves/served the editorial board of over 15 International Journals indexed by Thomson ISI. He is actively involved in the organization of several academic conferences, and some of them are now annual events. Dr. Abraham received Ph.D. degree in Computer Science from Monash University, Melbourne, Australia (2001) and a Master of Science Degree from Nanyang Technological University, Singapore (1998). More information at: http://www.softcomputing.net/