© November 2023| IJIRT | Volume 10 Issue 6 | ISSN: 2349-6002

User-Generated Bug Reports Using Topic Modelling and Sentimental Analysis

Dr. Bhavesh M. Patel[1], M. Sule[2]
[1]*Assistant Professor, Department of Computer Science, H.N.G. University, Patan, Gujarat, India*
[2]*Scholar, Department of Computer Science, H.N.G. University, Patan, Gujarat, India*

Abstract-Bug reports are a valuable resource for software troubleshooting and bug fixes, however, the process of obtaining such reports is cumbersome and expensive. One of the effective channels of receiving bug reports is from user reviews, these often show how end users of the app feel about features and bugs. But the problem with manually iterating through a bunch of user reviews and feedback is that it is time-consuming and inefficient. This study deploys a state-of-the-art technique using topic modeling and sentimental analysis to cluster user reviews into manageable topics and identify the sentiments behind them. The study reveals how topic modeling and sentimental analysis can be useful techniques for digging into user reviews to find bug reportsand spot serious issues that could have a negative impact on user experience. Byemploying these techniques, software developers can better interpret user feedback efficiently and prioritize their bug-fixing efforts accordingly. User reviews related to bugs in mobile applications are examined using NLP tools such as BERTopic and VADER (Valence Aware Dictionary and sEntiment Reasoner). The research has practical applications for app developers who can use it to improve user experience and make their apps better. By offering new perspectives on the relationship between subjects covered in bug reports from user reviews and the sentiments behind them, this study adds to the increasing body of knowledge on the efficient generation of bug reports, topic modeling, and sentiment analysis.

Keywords: NLP Tools, topic modeling, sentimental analysis

INTRODUCTION

Today, mobile applications are an integral part of our daily lives because it provides a lot that makes our lives easy and efficient. For every category of application our there, there is a significant number of alternatives. If developers want to ensure that their application stands out in this crowded market, they must focus on offering a superior user experience. One of the key elements that affect the user experience is the software's quality [1], which can be affected by bugs and other flaws in the application.

Developers detect and fix these issues using bug reports from different sources, many large companies maintain an internal team that tests software and detects bugs early on before features are shipped, however, it is still difficult to catch certain bugs, especially with consistent scalability. The development team, therefore, resorts to different methods of getting bug reports, one such method that has proven effective is bug reports collected from end users. This can be done through surveys, questionnaires, and app reviews. Compiling and analyzing app reviews individually is a highly challenging task, time-consuming, and labor-intensive. Furthermore, user- submitted bugs through app reviews might not always include a complete and in-depth explanation of the problem.

In this work, bug reports detected in user reviews are generated using topic modeling and sentiment analysis using BERTopic[2] and VADER (Valence Aware Dictionary and sEntiment Reasoner) [3].

A.Topic Modelling

Large amounts of text data can be analyzed and understood effectively using topic modeling[3,4]. It is a statistical method that makes it possible to spot hidden patterns in unstructured data. The analysis of social media[5], consumer feedback[6], market research[7], and other disciplines have all made extensive use of topic modeling. Topic modeling will be used in this study to analyze user-generated bug reports to better understand the topics users discuss and their sentiments about the apps. We will be able to find a cluster of reviews that are related to one another and extract user sentiments, giving us a better understanding of the main concerns and feelings that users have when reviewing these apps.

Types of Topic Modelling Techniques:

1. Latent Dirichlet Allocation(LDA)[4]:
   This method of probabilistic topic modeling assumes that each text consists of a variety of subjects, with each topic representing a probability distribution over words. LDA is frequently used in both academic and industrial subject modeling [19].
2. Non-Negative Matrix Factorization (NMF) [8]: Another popular topic modeling method is NMF, which breaks down a word frequency matrix into a collection of non-negative basis vectors that stand in for topics. Because it generates understandable subjects and can handle big datasets.
3. Hierarchical Dirichlet Process (HDP)[9]: HDP is a non-parametric Bayesian extension of LDA that determines the number of topics from the data automatically. HDP is usually used In situations where the number of topics is unclear.
4. Correlated Topic Models (CTM)[10]:As an extension of LDA,CTMtakestopiccorrelations into account. This is very helpful when evaluating datasets with similar subjects.
5. BERTopic [2]: BERTopic is a recently created technique that builds dense embeddings of documents using BERT[11] pre-trained language models. In contrast to other approaches, it generates themes by grouping the dense embeddings of documents.

The capability of BERTopic[2] to handle overlapping subjects [19] sets it apart from other established topic modeling techniques like LDA[4]. It can recognize varying subjects that share common keywords or concepts and allocate them to various clusters. It [2] also provides a user- friendly interface for parameter adjustment and visualization of the generated topic clusters. Additionally, It can analyze millions of documents quickly while utilizing the strength of BERT's [11]pre-trained model which is crucial for the unstructured nature of user-generated content.

A. SentimentAnalysis

Sentiment analysis, which refers to the underlying emotion, opinion, or attitude indicated by the writer or speaker, is a technique used to extract and categorize subjective information fromtextual data. The goal of sentiment analysis is to identify the text's polarity which tells whether it is positive, negative, or neutral[2].

Sentiment analysis has been applied in areas such as Market research [7], social media [5], and customer feedback analysis [6] over the years. Businesses use it to determine how customers feel about their goods or services and then adjust their business decisions accordingly.

There are many approaches to sentimental analysis such as rule-based [3,12,13,14], statistical-based[15,16],andusingmachinelearningtechniques.Rule-basedapproachesidentifyatextas positive, negative, or neutral based on the presence of particular keywords or phrases by applying established criteria. To categorize text based on statistical patterns in the training data, statistical approaches use machine learning algorithms like Naive Bayes, Support Vector Machines (SVM), or Logistic Regression.

Popular sentiment analyzers include TextBlob [13] and VADER (Valence Aware Dictionary and sEntiment Reasoner). VADER [2] is a rule-based sentimental analyzer that uses a pre-assigned score lexicon of words and phrases to determine the polarity of the text. On the other hand, TextBlob[3]isa Python package that conducts sentiment analysis on text using machine learning methods. Sentiment analysis is a great tool for companies to understand the behavior of their customers regarding their products and services, and it may be utilized to enhance client satisfaction, brand loyalty, and reputation.

LITERATUREREVIEW

There are a lot of tasks involved in bug reporting and fixes. One such task is Bug triaging. Bug triaging is an important task in software development where bugs are assigned to the most appropriate developer for fixing. However, with the increase in software size and the number of developers, bug triaging becomes more complex. In the paper [17], the authors propose a novel framework for bug triaging that uses a specialized topic modeling algorithm called a multi-feature topic model (MTM) to map the words in bug reports to corresponding topics. The MTM algorithm extends the Latent Dirichlet Allocation (LDA) [4] by considering the product and component information of bug reports. The authors also propose an incremental learning method called TopicMiner that considers the topic distribution of new bug reports to assign the appropriate fixer based on the affinity of the fixer to the topics. The proposed framework, Topic

MinerMTM, is evaluated on five large bug report datasets, including GCC, OpenOffice, Mozilla, Netbeans, and Eclipse, with a total of 227,278 bug reports.

Author [18] proposed JST model is based on Latent Dirichlet Allocation (LDA). The JST model extends LDA by adding a sentiment variable to the topic model, which represents the sentiment polarity of each word in the document. The joint modeling of sentiment and topic enables the model to capture the relationship between topics and sentiments, which improves sentiment classification accuracy. The JSTmodel is fully unsupervised and has a significant advantage over traditional supervised approaches, which often require a large amount of labeled data for training a sentiment classifier.

Trying to extract topics while maintaining context is extremely challenging, but it is necessary for areas such as legal. the relevant topics in legal documents strongly depend on the context inwhich they will be presented. To address this issue, the authors of this paper proposed using LEGAL-BERT, which extends BERTopic to model topics in legal documents.

The authors focused on a subset of landmark cases from the US Caselaw dataset and evaluated the impact of topic modeling on these cases. They investigated different variations of generating sentence embeddings from the cases and found that considering references to statutory law, such as the USCode, during the process of textem beddings significantly improved the quality of topic modeling. The study presented in this paper highlights the potential of using BERTopic for topic modeling while maintaining context. This makes it ideal and gives it an edge over other topic modeling techniques such as LDA [4]

METHODOLOGY

A. Data Collection and Preprocessing

To generate the topics, user reviews for two of the most popular mobile apps (Facebook and Instagram) on Play Store were scrapped. These apps were chosen with the assumption that they have a wide range of user base with different backgrounds and opinions. Each review represents a user's feedback from using the app. A total of 10760 reviews were scrapped, 5240 were scrapped from Facebook users, and 5520 from Instagram users.

There views were filtered using a list of predetermined terms linked to bug reports, such as "bug,""crash,""error," and "issue," etc., this strips our dataset down to only relevant reviews. Since we are using Bertopic [2] and not LDA [4], no preprocessing procedures were necessary [2].

B. Topic Modelling and Sentimental Analysis

To cluster reviews into manageable chunks of topics, we used BERTopic [2], a cutting-edge unsupervised topic modeling technique that uses BERT. In comparison to other conventional topic modeling algorithms like LDA[4] and Non-negative Matrix Factorization[8], BERTopic[2] demonstrates superior performance [19].We also used VADER (Valence Aware Dictionary and sEntiment Reasoner) [3] to analyze the sentiment of each review in a cluster. VADER [3] is a lexicon-based sentimental analyzer that rates each sentence in a text with a positive, negative, or neutral score. It can be used to classify emotions, moods, and other sentiments. It works by assigning each word in the text a polarity score to determine the sentiment of the text. By adding together, the polarity scores of all the words in the text, the overall sentiment score of the text can be determined.

RESULTS AND DISCUSSION

A. Overview of the results

The findings of the sentiment analysis and topic modeling performed are summarised in Table 1 below. The table lists the number of documents/reviews in the dataset as well as the sentiment type at different stages of the analysis.

Table1:Summary of Results

	Facebook	Instagram
Dataset	5240	5520
PositiveSentiments	1905	2480
NegativeSentiments	2911	2701
Neutral Sentiments	424	339
Datasetafterfiltering	1412	1990
TotalDocuments/ReviewsClustered	975	1401
PositiveSentiments(afterfiltering)	375	561
NegativeSentiments(afterfiltering)	513	744
NeutralSentiments(afterfiltering)	87	96
Topics Generated	24	29

For each app, the table shows the number of articles clustered, as well as the number of positive, negative, and neutral sentiments. The initial dataset for

Facebook had 5,240 documents, of which 1,905 were positive sentiments, 2,911 negative sentiments, and 424 neutral sentiments. The dataset was downsized to 1,412 documents after filtering out unnecessary reviews using keywords such as bugs, error, etc. The number of sentiments after the filtering was also recorded. The result was a total of 24 topics generated for Facebook and 29 for Instagram.

B. Topic Modeling Results

We performed topic modeling on the user-generated bug reports to obtain an understanding of the main topics and as well as sentiments around those topics. The goal was to cluster bugs that were related to one another.

Table2:User-Generated Bug Reports for Facebook Reviews

Topic ID	Topic	Total Clustered Documents/ Reviews	Positive Sentiments	Negative Sentiments	Neutral Sentiments
1	facebook_the_and_to	195	77	102	16
2	log_in_it_my	126	40	75	11
3	video_videos_the_play	84	30	44	10
4	update_the_app_and	68	24	35	9
5	app_it_the_crashing	57	18	27	12
6	account_login_my_facebook	47	17	29	1
7	reels_comments_reel_it	38	15	19	4
8	marketplace_items_and_you	38	12	23	3
9	photos_featured_to_photo	35	17	14	4
10	account_my_facebook_hacked	34	11	23	0
11	facebook_loading_it_and	32	15	16	1
12	notifications_notification_no_the	29	9	17	3
13	feed_news_the_posts	23	6	16	1
14	music_story_it_bug	20	11	8	1
15	comments_the_comment_but	20	6	10	4
16	page_create_created_creating	19	11	6	2
17	profile_picture_change_it	18	9	7	2
18	video_comment_comments_videos	16	8	7	1
19	ad_ads_of_the	15	7	8	0
20	dating_support_to_the	13	4	8	1
21	restricted_account_facebook_block	13	6	7	0
22	friend_friends_list_show	12	11	0	1
23	groups_group_members_member	12	6	6	0
24	code_authentication_get_to	11	5	6	0

The findings of the user-generated bug reports for Facebook reviews, which were clustered into several bugs, are presented in Table2. A large number of clustered documents/reviews in Topic 2 demonstrate that problems logging into Facebook were one of the frequent problems mentioned by users, which resulted in a large number of negative sentiments (75 out of 126 documents).

The results also mention issues with Facebook video playback (Topic 3), problems with app upgrades (Topic 4), and app crashes (Topic 5). The findings imply that users run into a variety of problems when using the Facebook app, and these problems more often than not are associated with negative sentiments. Overall, the findings of Table 2 offer insightful information about the kinds of problems that Facebook users have reported, which can be utilized to guide future efforts at app development and customer service.

Table3:User-Generated Bug Reports for Instagram Reviews

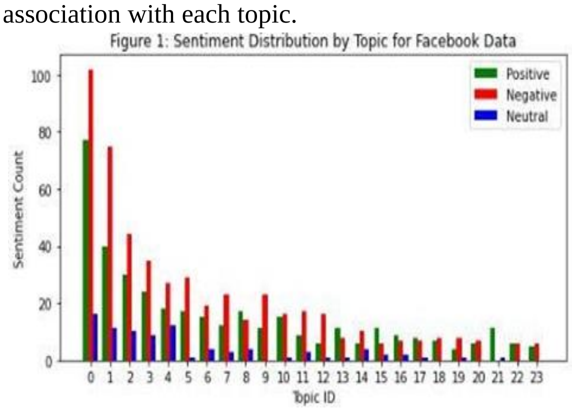
Topic ID	Topic	Total Clustered Documents/ Reviews	Positive Sentiments	Negative Sentiments	Neutral Sentiments
1	photos_the_post_photo	194	64	115	15
2	messages_message_to_the	132	44	69	19
3	app_it_and_of	116	51	58	7
4	reels_option_not_the	91	27	51	13
5	account_my_login_to	90	34	55	1
6	instagram_not_and_my	68	27	39	2
7	photos_instagram_post_photo	58	23	32	3
8	account_my_instagram_to	55	20	33	2
9	audio_reels_reel_the	53	23	25	5
10	reels_instagram_option_reel	53	24	27	2
11	story_post_stories_to	43	26	15	2
12	followers_following_follow_follower	37	20	14	3
13	feed_posts_the_it	36	9	23	4
14	app_crashing_crashes_open	36	16	18	2
15	instagram_app_is_that	35	18	17	0
16	chat_reply_instagram_message	34	22	11	1
17	video_videos_the_post	31	14	15	2
18	stories_story_app_the	31	7	21	3
19	reels_and_it_app	30	15	14	1
20	story_stories_instagram_it	27	17	10	0
21	dark_mode_theme_white	25	8	15	2
22	tiktok_to_the_and	21	8	13	0
23	reels_reel_instagram_on	19	5	13	1
24	music_songs_song_instagram	16	9	7	0
25	notes_feature_account_but	16	10	4	2
26	collaborator_invite_collaboration_colla borate	15	4	11	0
27	notes_feature_reported_times	15	6	7	2
28	bugs_instagram_new_updates	13	7	6	0
29	crashing_instagram_time_crashes	11	3	6	2

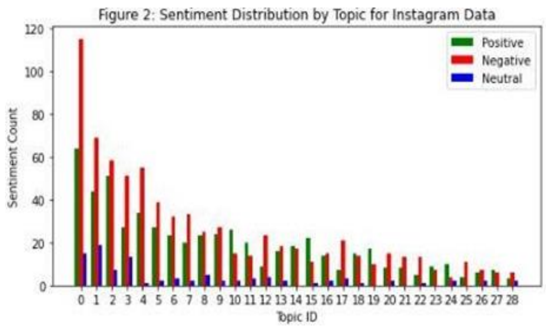
The sentiments for bug reports generated for Instagram were similar to that of Facebook, most topics received a high number of negative sentiments with a much lower neutral sentiment, while positive sentiments fell in between. For instance,Topic1(photos_the_post_photo) has 194 grouped evaluations that address problems with Instagram photo publishing. In these reviews, there were 115 negative reviews, 64 negative reviews, and 15 neutral.

The topic modeling findings show that users have a range of problems when using Instagram, with the most frequent problems being those involving photos, reels, messaging, and login issues. The majority of opinions in these clusters were negative, emphasizing the necessity for Instagram to fix these problems and enhance the user experience.

C. Sentiment Analysis Results

To understand the sentiments behind the topics, figure1 and figure 2 show the presents of each of the 3 sentiment classes (negative, positive, neutral) and their association with each topic.





According to the results of the sentiment analysis, the majority of the sentiments were negative and had a compound sentiment score of -0.5 or lower. Even with fair and accommodating compound ratings ranging from -0.5 to 0.5, neutral attitudes were relatively low. We carefully looked at a sample of reviews from each sentiment category to obtain a further understanding of the sentiment expressed in the reviews. We discovered that the app's features, interface, and usability were frequently associated with positive sentiments. Negative sentiments, on the other hand, frequently questioned the app's functionality and the existence of bugs.

CONCLUSION

In this study, we used topic modeling and sentiment analysis with BERTopic and VADER to assess user-generated bug reports from mobile app reviews. Our goal was to learn more about the topics covered in user reviews and the sentiments behind them. Several significant conclusions were drawn from our research, we uncovered bugs and issues faced by users. Additionally, our research sheds light on how well BERTopic and VADER can be deployed together to b.We were able to build a robust model.

The findings of our sentiment analysis show how effective VADER is in identifying the general sentiment represented in app reviews. The information gathered from the sentimental analysis may be used by app developers to better understand how users feel about their apps as a whole and pinpoint areas for improvement. For instance, the developers may need to enhance their customer support procedures to increase user satisfaction if a large number of negative sentiments focus on customer support.

Overall, by offering a new perspective on the relationship between topics covered in app reviews and the sentiment behind them, our work adds to the increasing body of knowledge on bug reporting, topic modeling, and sentimental analysis. Our research has practical applications for app developers who may use it to enhance user experience and make their apps much better.

REFERENCE

[1] H. Khalid, E. Shihab, M. Nagappan and A. E. Hassan, "What Do Mobile App Users ComplainAbout?,"inIEEESoftware,vol.32,no.3,pp.70-77,May-June2015,doi: 10.1109/MS.2014.50.

[2] Grootendorst,M.(2022).BERTopic:Neuraltopicmodelingwithaclass-basedTF-IDF procedure. ArXiv (Cornell University). https://doi.org/10.48550/arxiv.2203.05794

[3] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for SentimentAnalysisofSocialMediaText.EighthInternationalConferenceonWeblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[4] D.M.Blei,A.Y.NgandM.J.Jordan," Latent dirichletal location", Journal of Machine Learning Research, vol. 3, pp. 993-1022, March 2003.

[5] Negara,E.S.,Triadi,D.,&Andryani,R.(2019,October).Topicmodellingtwitterdata with latent dirichlet allocation method. In 2019 International Conference on Electrical Engineering and Computer Science (ICECOS) (pp. 386-390). IEEE.

[6] Vallurupalli,V.,&Bose,I.(2020).Exploringthematiccompositionofonlinereviews:A topic modeling approach. Electronic Markets, 30, 791-804.

[7] Reisenbichler,M.,&Reutterer,T.(2019).Topicmodelinginmarketing:recentadvances and research opportunities. Journal of Business Economics, 89(3), 327-356.

[8] Lee,D.D.,&Seung,H.S.(1999).Learningthepartsofobjectsbynon-negativematrix factorization. Nature, 401(6755), 788-791.

[9] Y.W.Teh,M.J.Jordan,M.J.BealandD.M.Blei,"HierarchicalDirichletprocesses",JournaloftheAmericanStatisticalAssociation,vol.101,no.476,pp.1566-1581, 2006.

[10] Blei,D.,&Lafferty,J.(2006).Correlated topic models. Advances in neural information processing systems, 18, 147.

[11] Devlin,J.,Chang,M.W.,Lee,K.,&Toutanova,K.(2018).Bert:Pre-trainingofdeep bidirectional

transformers for language understanding. arXiv preprint arXiv:1810.04805.

[12] Baccianella,S.,Esuli,A.,&Sebastiani,F.(2010,May).Sentiwordnet3.0:anenhanced lexical resource for sentiment analysis and opinion mining. In Lrec (Vol. 10, No. 2010, pp. 2200-2204).

[13] Loria,S.(2018).textblobDocumentation.Release0.15, 2(8).

[14] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014, June). The StanfordCore NLP natural language processing tool kit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations (pp. 55-60).

[15] N.Friedman,D.Geiger,andGoldszmidtM.Bayesiannetworkclassifiers.Machine Learning, 29:131–163, 1997

[16] Boser, B.E.,Guyon,I.M.,&Vapnik,V.N.(1992, July).A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory (pp. 144-152).

[17] X. Xia, D. Lo, Y. Ding, J. M. Al-Kofahi, T. N. Nguyen and X. Wang, "Improving AutomatedBugTriagingwithSpecializedTopicModel,"inIEEETransactionson Software Engineering, vol. 43, no. 3, pp. 272-297, 1 March 2017, doi: 10.1109/TSE.2016.2576454.

[18] Lin,C.,&He,Y.(2009,November).Jointsentiment/topicmodelforsentimentanalysis. In Proceedings of the 18th ACM conference on Information and knowledge management (pp. 375-384).

[19] Egger,R.,&Yu,J.(2022).Atopicmodelingcomparisonbetweenlda,nmf,top2vec,and bertopic to demystify twitter posts. Frontiers in sociology, 7.