DEFECTS PREDICTION IN APPS USING USER REVIEWS AND RATINGS

A Dissertation

Submitted in fulfilment of the

requirements for the award of the degree

Of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted By

ANKUR TAGRA

(14535005)

Under the guidance of

DR. DURGA TOSHNIWAL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

ROORKEE – 247667 (INDIA)

MAY,2016

CANDIDATE'S DECLARATION

I hereby declare that the work, which is presented in this dissertation report entitled " Defects prediction in apps using user reviews and ratings" towards the fulfilment of the requirements for the award of the degree of Master of Technology with specialisation in Computer Science And Engineering submitted in the department of Computer Science and Engineering, Indian Institute of Technology, Roorkee (India) , is an authentic record of my own work carried out during the period of July 2015 to May 2016 under the guidance of Dr. Durga Toshniwal, Associate Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee.

I have not submitted the matter embodied in this dissertation for the award of any other degree or Diploma.

Date :

Place : ANKUR TAGRA

CERTIFICATE

This is to certify that the above statements made by the candidate is correct to the best of my knowledge and belief.

Date :

Place :

Dr. Durga Toshniwal,

Associate Professor,

Department of Computer Science and Engineering,

IIT Roorkee

ACKNOWLEDGEMENT

It gives me immense pleasure to thank all those people who have, at various stages and in various ways have played a key role in successful completion of this work. I would take this opportunity to extend my heartfelt gratitude to my guide and mentor Dr. Durga Toshniwal, Associate Professor, Indian Institute of Technology, Roorkee for her invaluable advice, guidance, encouragement and for sharing her knowledge. Her wisdom and commitment to the highest standards motivated me throughout. She has been very generous in providing the necessary resources to carry out the research. She is an inspiring professor, a great advisor and most importantly a person.

I would also like to thanks Karthikeyan Dakshinamurthy, Ashish Mathur, Vijay Ekambaram, sarath chandar from IBM for their valueable suggestions. I am also highly indebted to my colleague Maj. Sumit Prakash Gupta, Bharat Goel, and my family who gave me the moral support and valuable suggestions. On a personal note, I owe everything to the almighty.

Ankur Tagra

ABSTRACT

In the current digital era approximately 2 million applications (a.k.a. apps) are present on app store which allow users to give ratings and reviews. App developers face serious challenges in getting user feedback. Every app developer is in constant dilemma of DIPMAP: Did I program a poor mobile app? The app developer constantly strives for eliminating the defects to increase the user base and app rating. The app developer wants to exploit the expressive power of raw user reviews regarding issues faced by app users while using the app. But with the sheer volume of these raw reviews a lot of knowledge goes untapped which is useful for app developers. We propose an unsupervised novel model for defect prediction using app reviews by (i) review preprocessing (ii) Making Vector Representations of reviews (iii) classifying review into broad classes (iv) Making prioritized defect phrases.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 Introduction

With the advent of android operating system for the mobile technology, the mobile based applications (apps) approach has come as an alternative to traditional web based developmental approach. Thousands of apps are published on the Google Play Store daily. As of Feb 2016, there are about 2 million apps on the Play Store [1]. Official data released by Google in 2016, says that there are more than 1 billion active android users on a monthly basis [2]. These users install and use these applications in their day to day life .A lot of these android applications contain some defects or errors or bugs which occur because of either the hurry of the app developer to launch their app in the Play store or because of the limited knowledge the developer possesses. After using these applications the users provide their feedback for that application in two ways. First is in the form of giving reviews and ratings for that app in the Play store and second is via reporting an error [1]. Developing team goes through mostly reported errors provided by users and manually test for each defect in the application and come up with the revised edition of the application (new version) this takes a lot of time and effort. The developers do not tap the information provided by user reviews and ratings on the Play Store for that application because of the sheer volume of these user reviews. It is very difficult for developer to go for each and every review manually and read these reviews to identify the defects that the user is talking about [1]. As per the Google Play Store data there are applications that have huge number of user reviews and ratings, such as JUST DIAL has 0.13 million reviews and ratings, True Caller has around 2.2 million reviews and ratings. So to manually go for each review and finding defect is a difficult task. So they generally tend to ignore it. We need to come up with a solution that automatically predicts top "k" defects that users are facing after using the app. That will give the developer an easy task to remove these defects in a later release version so that it improves its user experience and then users will give an improved rating to it in the subsequent version. So the overall rating of app will improve on the Play Store, which means that the users are quite happy with the latest version.

1.2 Motivation

As of now, the usual way of finding defects in an application is by testing it rigorously. It takes a lot of time and human effort to manually test all the app features and processes for errors. An easy way is to use the user reviews as a base to find errors because users are assumed to download the app use it, analyze it and report some errors by giving their comments for these apps. But with such huge volume of these reviews it is difficult to manually read these reviews and find out the defects in the app. Here our project comes in, it automatically detects all the defects in the app by analyzing all the user reviews and ratings for the app.

1.3 Problem Statement

The above problem is stated as *To automatically generate set of defects in Google Play Store app using reviews and ratings provided by users for that application***.**

An app developer publishes his app on Play Store. He is in the dilemma Did I program a poor mobile app (DIPMAP). Users use these apps and they generally give reviews and ratings for that app which they have used. These reviews contain a vast amount of information about the app which they have used extensively. Actually this is typically a crowd sourced testing for the app. These reviews contain information about defects in a particular app. But due to the vast amount of user reviews it is difficult to manually read each and every review and come up with various defects in that app. So, this user review data is usually left untapped. This review data can generate valuable information which is helpful in automatic detection of app defects by App developer, so that he can come up with the improved version of the app in the subsequent release.

We don’t have a list of defects for comparison of our work with the actual defects as the app developers don’t publish the defects in the app for commercial reasons. So we have taken the help of domain experts for manual tagging of reviews with associated defects. Thus we have generated the ground truth of apps. We have assumed that this ground truth represents the actual list of defects.

1.4 Research Contribution

In the present world of ‘Android’ boom, all app developers want to make their apps user friendly and defect free. For that developer wants to predict the defects in the apps, which is not an easy task. This problem has been referred to as DIPMAP(Did I program a poor mobile app).

No other research work right now is predicting the defects in the apps. They are just extracting features for the app. Our approach is a novel approach for predicting the prioritized list of defects in the apps. It finds the semantic relation between words in an unsupervised way.

We have proposed a unique model TF-mIDF for conversion of word vectors into review vectors. It is better than the original TF-IDF in our work as it generates unique representations for words.

We have proposed an algorithm for tuning the value of k in K-Means clustering algorithm which is helpful in making of clusters that define popular defects.

We have also proposed criteria for measuring accuracy of our DIPMAP problem as there is no previous work to compare our results with we have termed it as DIPMAP Analyzer (DA).

1.5 Organization of Report

The Report consists of 4 chapters. First chapter is of Introduction to the project. Second chapter discuss about Literature survey. Third chapter discusses about the proposed framework that includes various techniques while developing the project such as word representations, TF-midf, Classification, Clustering and. Fourth chapter is of Experiments and results. The fifth chapter is of conclusion.

CHAPTER 2 LITERATURE REVIEW

2.1 Background and Related Work

Thus far there has been very little work for DIPMAP (i.e. “Did I Program a Poor Mobile App?”) using raw user reviews. The work done [6, 7, 8, 9, 10] so far concentrates only on feature extraction using, sentiment analysis skipping the main motive of defect prediction.

Guzmen [6] proposed a framework for feature extraction and thereby producing features of the app with their sentiment scores without predicting the defects. Features extracted here have no relation with the defects e.g., great, good, bad ,like... are also coming as features but they do not represent any defect.

Chen [7] proposed a framework for statistical analysis of user reviews and gives most popular user reviews in a supervised way thus will not have generalisation capability.

Bin [8] addressed whether the app is popular among users or not but failing to unveil why they do not like the app. It does statistical analysis of user reviews disregard of semantic analysis.

Cuiyun [9] did the high level feature extraction but feature may not represent the defects. It requires knowledge of developers for prioritizing the issues in the app. Manual intervention of the app developer is the bottleneck in this approach.

Gao [10] predefined the features and used statistical analysis of these reviews to predict the severity of these features. Hence it fails to detect new features.

We have studied various techniques used in the paper “Guzman, Emitza, and Wiem Maalej. "How do users like this feature? a fine grained sentiment analysis of app reviews." *Requirements Engineering Conference (RE), 2014 IEEE 22nd International*. IEEE, 2014” [6]. In this paper they have taken the user reviews and applied collocation finding algorithm provided in NLTK toolkit to extract the features from the reviews. The collocation finding algorithm is discussed in the next section. The features extracted are then assigned a negative score based on the negative sentiment of the review containing them [5]. They have done this by using a lexical sentiment extraction tool called “sentiStrength”. After assigning the negative score to all the features, they extract high level features from these features using topic modeling techniques. These high level features are the list of defects that are evaluated on the basis of user reviews.

Collocation Finding Algorithm

This algorithm is used for extracting features from the user reviews. It is a collection of words that co-occur unusually [6]. Taking an example <strong tea> is a collocation while < powerful tea> is not a collocation because it does not occur usually in English dictionary. So, by applying collocations we find words which are uniquely identifiable, they may contain defects. Further they will process these words to get refined reviews.

Sentiment Analysis

The analysis is done for these reviews containing these features that are found by collocation finding algorithm. These reviews are assigned so scores based on their negativity in range [-1, 3] “using sentiStrength” [6]. This score is assigned to features that are present in these reviews.

Topic Modeling

This approach gives high level Features form the previously found features [6]. It does sampling to assign reviews to topics. Each review may be assigned to several topics. Then the features are grouped in more refined based on these topics [6]. This gives highly summarized features.

So Guzman only extracts the features from the apps but unable to give the prioritized list of defects.
Whereas Gao is just doing the statistical analysis of reviews and not concentrating on defects.

Cuiyan also didn’t find the defects in the apps. He also didn’t do the semantic analysis of reviews , his work is not generalized for all apps. It is also taking developer input as a base for his research which is a bottleneck to its scalability. He only did the high level feature extraction.

2.2 Research Gaps

In the existing research the main motive is to only extract features from user reviews. It does not concentrates on predicting defects. It does not take into consideration the semantic relationship between the words. So it may predict two reviews which are different in selection of words but are talking similarly i.e., semantically almost equal as different. Take a scenario, there are two reviews “My phone has become very slow after installing this app. Every time I open this app it crashes!!” and a review “This app is crap, it hangs a lot on my Samsung galaxy Note” so, both are discussing about similar problem i.e. App hangs. But it considers these as different defects.

Our approach takes into consideration the semantic relationship between words. It does so using an unsupervised approach. And it predicts the defects in the app.

CHAPTER 3 PROPOSED WORK

3.1 Proposed Approach

The main goal of our approach is to predict the defects in apps by automatically analyzing the Phase I: Data collection and pre-processing

Phase II: word to vector conversion model

Phase III: Review Vectorization and Classification

Phase IV: Prediction of Defects

3.1.1 Phase I: Data Collection and Pre-processing

Data set is collected for various apps from Google Play. It contains App name, also user reviews and corresponding user ratings.

S.No	App Name	Category	Platform	No. of Reviews	No. of Neg Reviews
1	Bfs Ninja jump	Gaming	Google Play	15245	2176
2	Espn fantasy football	Gaming	Google Play	11436	1908
3	Fruit ninja	Gaming	Google Play	9166	3217
4	Dead trigger	Gaming	Google Play	2987	883
5	Shoot bubble Deluxe	Gaming	Google Play	3055	2685
6	Stick Cricket	Gaming	Google Play	21524	3449
7	Three d Bowling	Gaming	Google Play	6589	1378

Table 3.1: Data collection

Table 3.1 shows data set collection for various apps, category, platform and no. of reviews collected and no. of negative reviews.

CHAPTER-4 EXPERIMENTS AND RESULTS

4.1 Data Set Description

Data set was collected for 7 apps. It consists of app name, user reviews, user ratings

S.No	App Name	Category	Platform	No. of Reviews	No. of Neg Reviews
1	Bfs Ninja jump	Gaming	Google Play	15245	2176
2	Espn fantasy football	Gaming	Google Play	11436	1908
3	Fruit ninja	Gaming	Google Play	9166	3217
4	Dead trigger	Gaming	Google Play	2987	883
5	Shoot bubble Deluxe	Gaming	Google Play	3055	2685
6	Stick Cricket	Gaming	Google Play	21524	3449
7	Three d Bowling	Gaming	Google Play	6589	1378

Table 4.1: Data set collection

Table 4.1 shows the apps with no. of collected reviews. The reviews were collected using Google. It shows no. of negative reviews, which were pre-processed from all set of reviews using rating criteria as explained in section 3.1.1.

Data set for various apps had been collected figure shows total reviews, No. of negative reviews, platform and app name.

4.2 Results

Phase 2: Word Vector Representations

We performed some sample words on our tool and found out words which are closely related to a particular word. Let us take an example of word “screen”.

S.No	Word	Related Word	Cosine Distance
1	Screen	Screens	0.701141
		Monochrome	0.576821
		Pixels	0.516020
		Display	0.496503
		Logon	0.480673
2	Ads	Adds	0.861832
		Advertisement	0.711657
		Commercial	0.524137
		Promotions	0.435819
3	Hangs	hang	0.891287
		paragliding	0.721327
		crashes	0.612837
		lean	0.412234

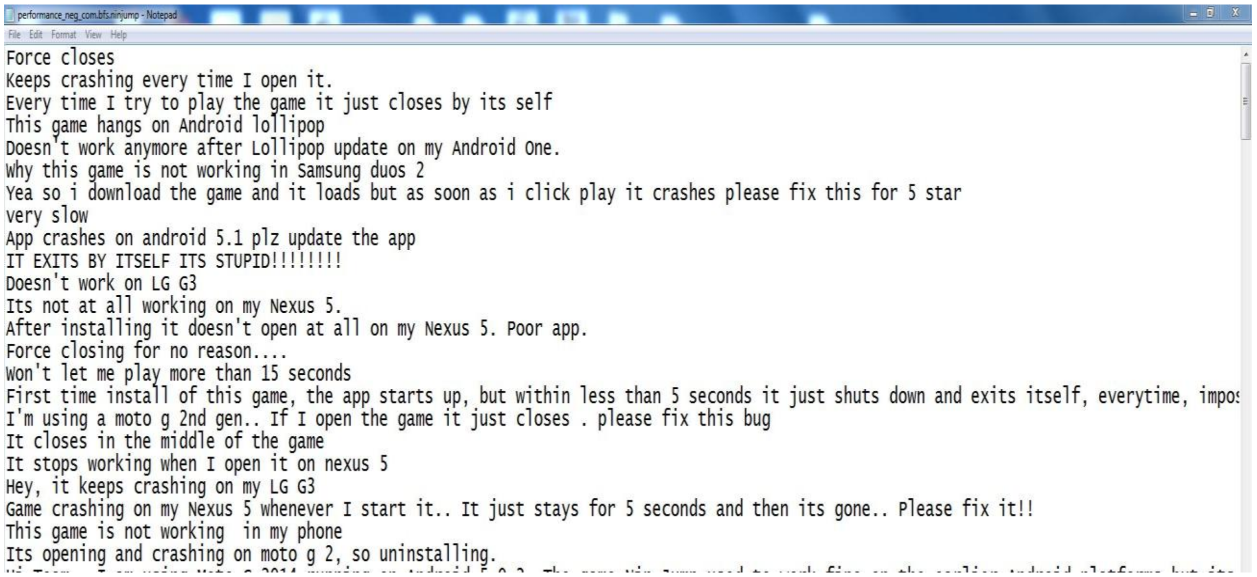
Table 4.2 : Closest words

Table 4.2 shows the closest words to word “screen”, “Ads”, “Hangs”.

This signifies that word “Screen” is related to word “Monochrome”, “Pixels”, “Display”, “Logon” and this whole thing is being done in an unsupervised way. So the semantic relation between words is maintained.

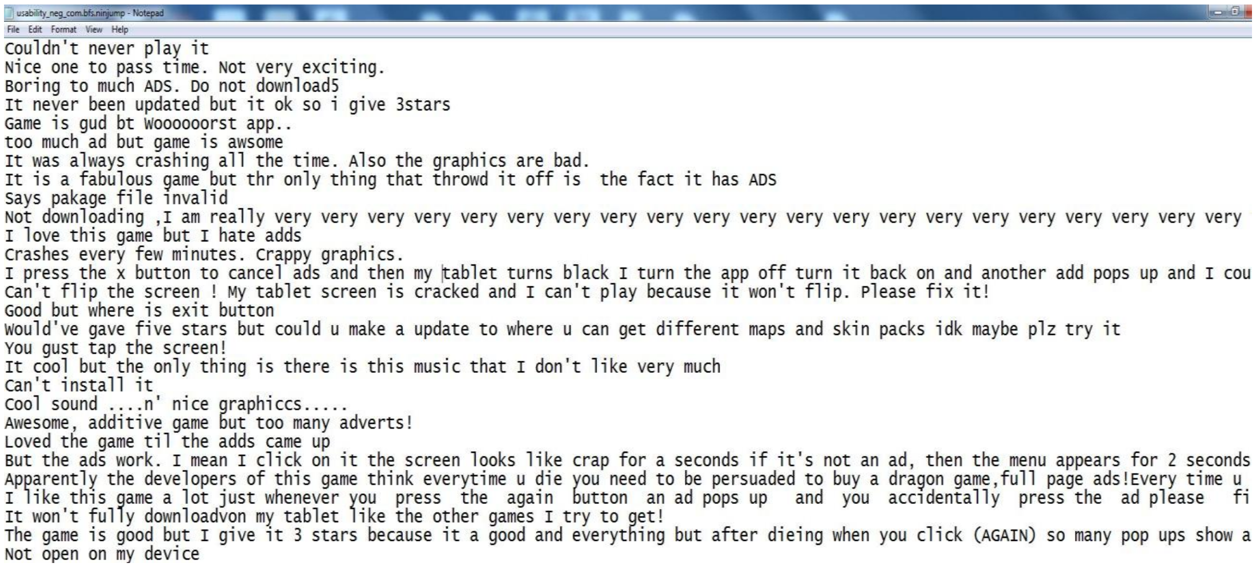
Phase 3: Review vectorization and classification

The sample results for Linear SVM classification of user reviews into various classes such as performance based, Usability based, functionality based and junk based are shown below.


performance_neg_com.bfs.ninjump - Notepad
File Edit Format View Help

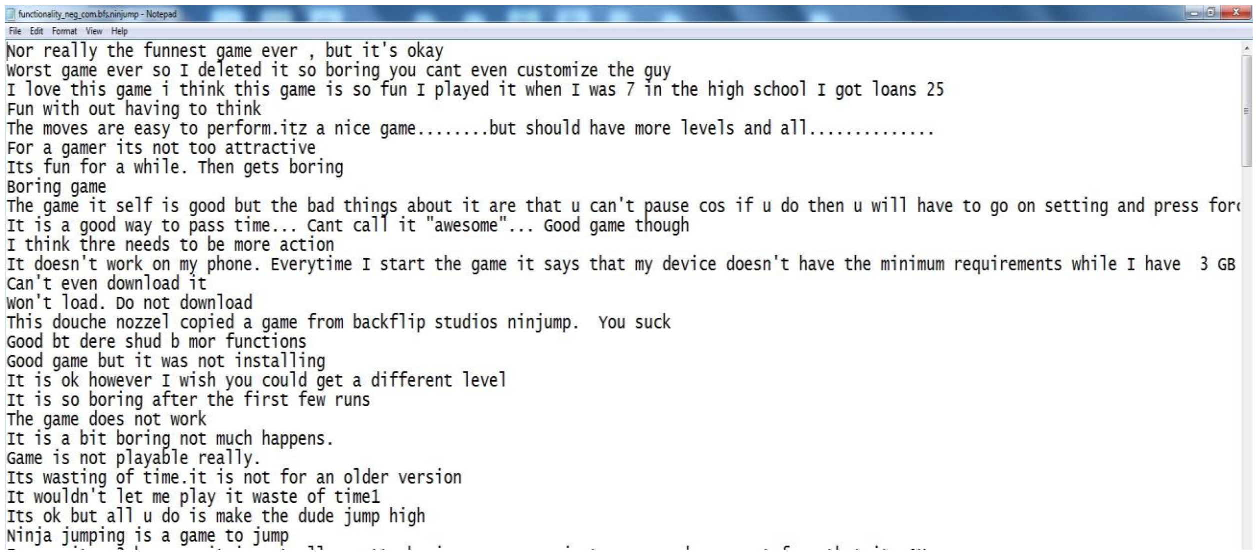
```
Force closes
Keeps crashing every time I open it.
Every time I try to play the game it just closes by its self
This game hangs on Android lollipop
Doesn't work anymore after Lollipop update on my Android One.
Why this game is not working in Samsung duos 2
Yea so i download the game and it loads but as soon as i click play it crashes please fix this for 5 star
very slow
App crashes on android 5.1 plz update the app
IT EXITS BY ITSELF ITS STUPID!!!!!!!!
Doesn't work on LG G3
Its not at all working on my Nexus 5.
After installing it doesn't open at all on my Nexus 5. Poor app.
Force closing for no reason....
Won't let me play more than 15 seconds
First time install of this game, the app starts up, but within less than 5 seconds it just shuts down and exits itself, everytime, impos
I'm using a moto g 2nd gen.. If I open the game it just closes . please fix this bug
It closes in the middle of the game
It stops working when I open it on nexus 5
Hey, it keeps crashing on my LG G3
Game crashing on my Nexus 5 whenever I start it.. It just stays for 5 seconds and then its gone.. Please fix it!!
This game is not working  in my phone
Its opening and crashing on moto g 2, so uninstalling.
```

Fig 4.1: Performance reviews


usability_neg_com.bfs.ninjump - Notepad
File Edit Format View Help

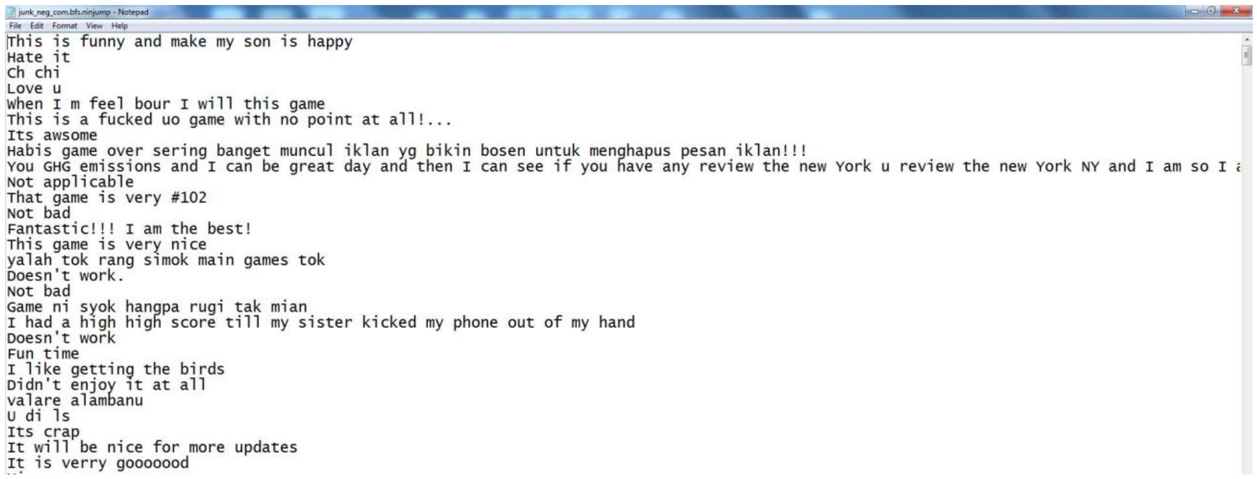
```
Couldn't never play it
Nice one to pass time. Not very exciting.
Boring to much ADS. Do not download5
It never been updated but it ok so i give 3stars
Game is gud bt Woooooorst app..
too much ad but game is awsome
It was always crashing all the time. Also the graphics are bad.
It is a fabulous game but thr only thing that throwd it off is  the fact it has ADS
Says pakage file invalid
Not downloading ,I am really very very very very very very very very very very very very very very very very very very very very very
I love this game but I hate adds
Crashes every few minutes. Crappy graphics.
I press the x button to cancel ads and then my tablet turns black I turn the app off turn it back on and another add pops up and I cou
Can't flip the screen ! My tablet screen is cracked and I can't play because it won't flip. Please fix it!
Good but where is exit button
Would've gave five stars but could u make a update to where u can get different maps and skin packs idk maybe plz try it
You gust tap the screen!
It cool but the only thing is there is this music that I don't like very much
Can't install it
Cool sound ....n' nice graphiccs.....
Awesome, additive game but too many adverts!
Loved the game til the adds came up
But the ads work. I mean I click on it the screen looks like crap for a seconds if it's not an ad, then the menu appears for 2 seconds
Apparently the developers of this game think everytime u die you need to be persuaded to buy a dragon game,full page ads!Every time u
I like this game a lot just whenever you  press  the  again  button  an ad pops up   and  you  accidentally  press the  ad please   fi
It won't fully downloadvon my tablet like the other games I try to get!
The game is good but I give it 3 stars because it a good and everything but after dieing when you click (AGAIN) so many pop ups show a
Not open on my device
```

Fig 4.2: Usability reviews


functionality_neg_com.bfs.ninjump - Notepad

```
Nor really the funnest game ever , but it's okay
Worst game ever so I deleted it so boring you cant even customize the guy
I love this game i think this game is so fun I played it when I was 7 in the high school I got loans 25
Fun with out having to think
The moves are easy to perform.itz a nice game........but should have more levels and all.............
For a gamer its not too attractive
Its fun for a while. Then gets boring
Boring game
The game it self is good but the bad things about it are that u can't pause cos if u do then u will have to go on setting and press for
It is a good way to pass time... Cant call it "awesome"... Good game though
I think thre needs to be more action
It doesn't work on my phone. Everytime I start the game it says that my device doesn't have the minimum requirements while I have  3 GB
Can't even download it
Won't load. Do not download
This douche nozzel copied a game from backflip studios ninjump.  You suck
Good bt dere shud b mor functions
Good game but it was not installing
It is ok however I wish you could get a different level
It is so boring after the first few runs
The game does not work
It is a bit boring not much happens.
Game is not playable really.
Its wasting of time.it is not for an older version
It wouldn't let me play it waste of time1
Its ok but all u do is make the dude jump high
Ninja jumping is a game to jump
```

Fig 4.3: Functionality reviews


junk_neg_com.bfs.ninjump - Notepad

```
This is funny and make my son is happy
Hate it
Ch chi
Love u
When I m feel bour I will this game
This is a fucked uo game with no point at all!...
Its awsome
Habis game over sering banget muncul iklan yg bikin bosen untuk menghapus pesan iklan!!!
You GHG emissions and I can be great day and then I can see if you have any review the new York u review the new York NY and I am so I
Not applicable
That game is very #102
Not bad
Fantastic!!! I am the best!
This game is very nice
yalah tok rang simok main games tok
Doesn't work.
Not bad
Game ni syok hangpa rugi tak mian
I had a high high score till my sister kicked my phone out of my hand
Doesn't work
Fun time
I like getting the birds
Didn't enjoy it at all
valare alambanu
U di ls
Its crap
It will be nice for more updates
It is verry gooooood
```

Fig 4.4: Junk reviews

The above figures i.e. Fig. 4.1, Fig. 4.2, Fig. 4.3 and Fig. 4.4 shows classification outputs of performance based reviews, Usability based reviews, Functionality Based and Junk Based Reviews respectively.

We classified reviews into Performance, usability and functionality based reviews.

For apps "Ninja Jump" and "Stick Cricket" results of classification are as follows:

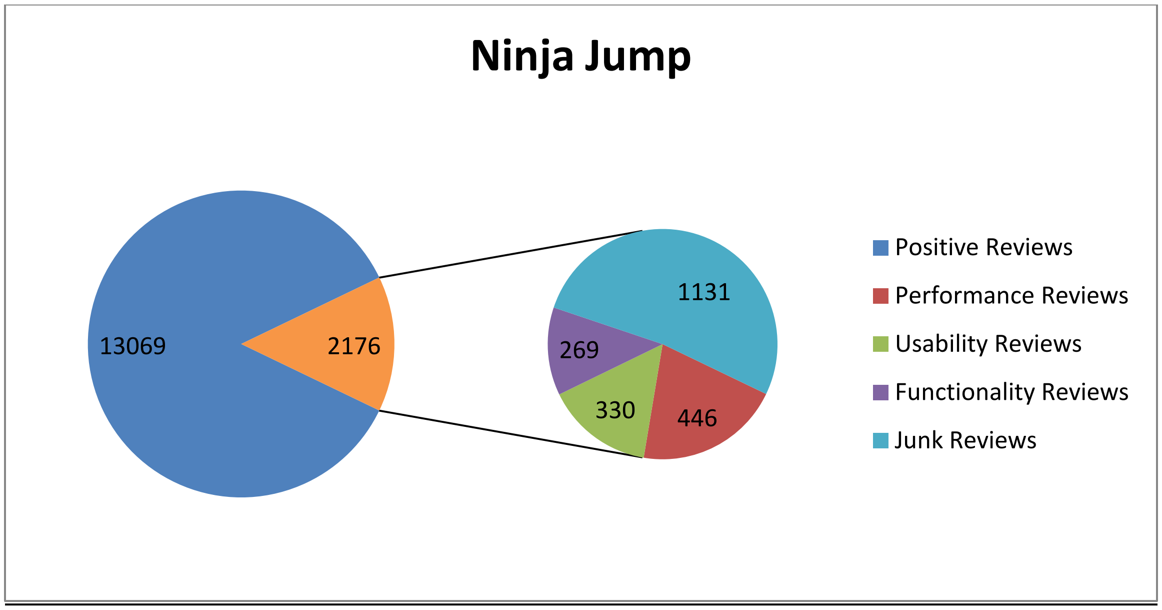


Fig 4.5 : Review classification for "Ninja Jump"

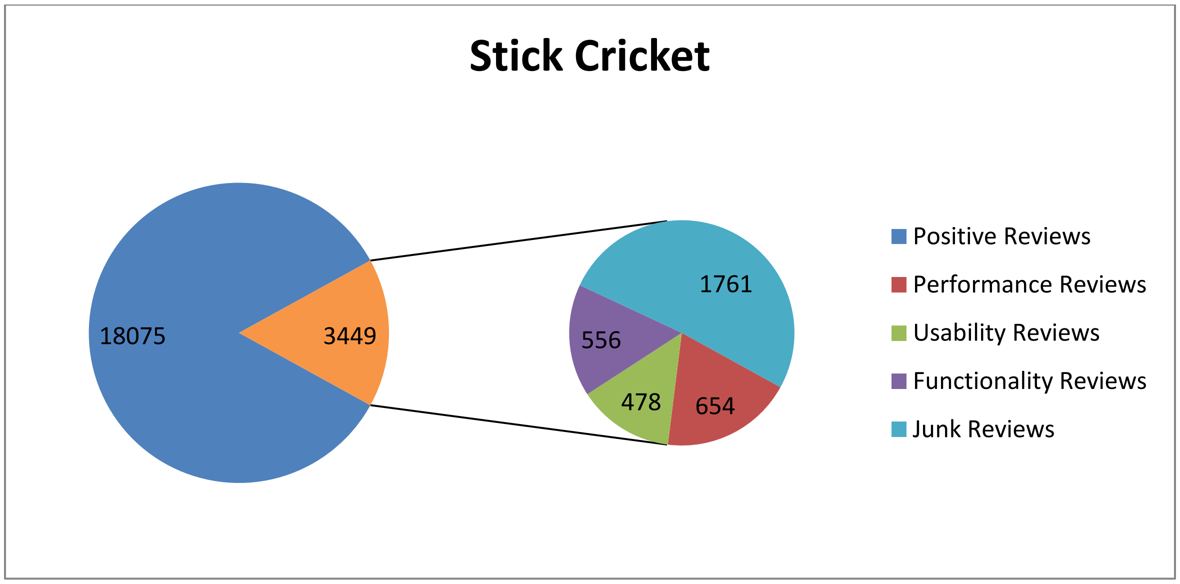
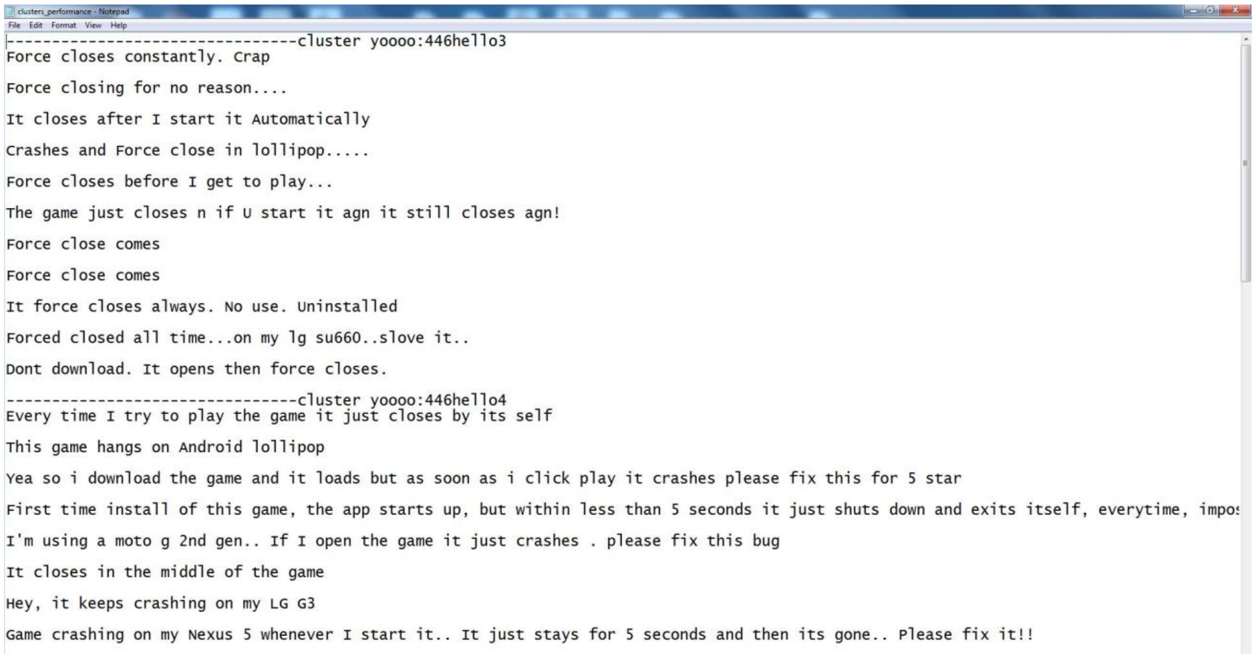


Fig 4.6 classification for "Stick Cricket"

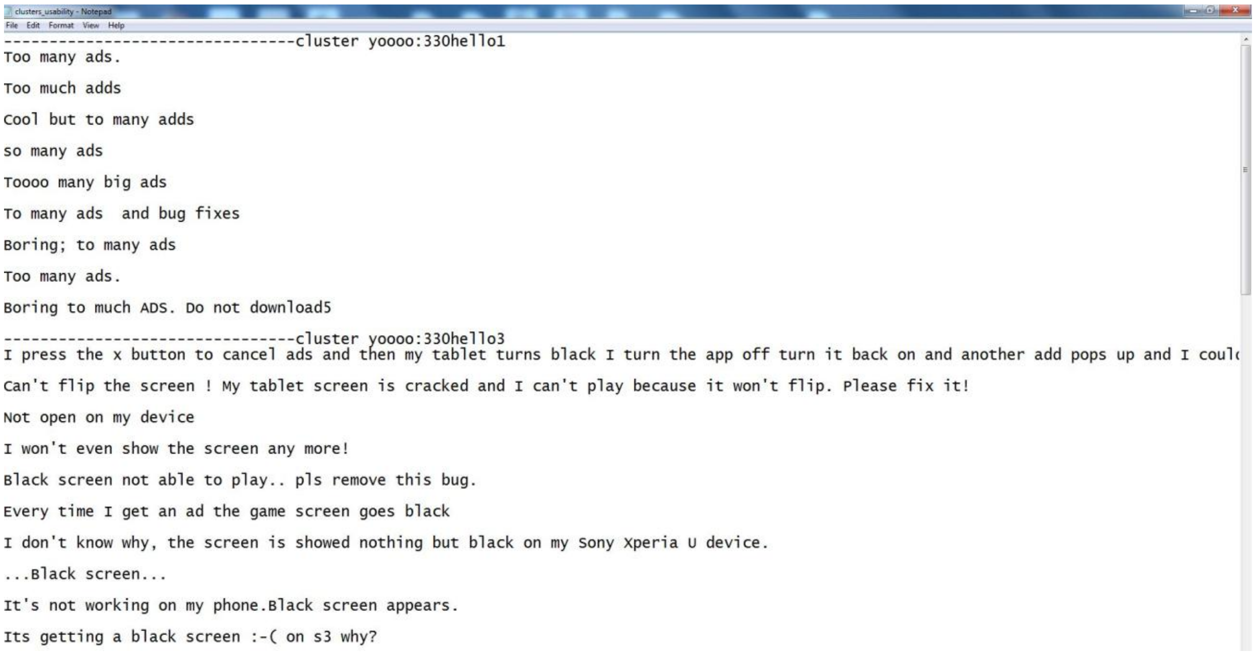
Phase 4: Prediction of Defects

After Classifying these user reviews into Performance based, usability based, functionality based and junk based. Then we did clustering on each of these class reviews. Fig shows various clusters formed.



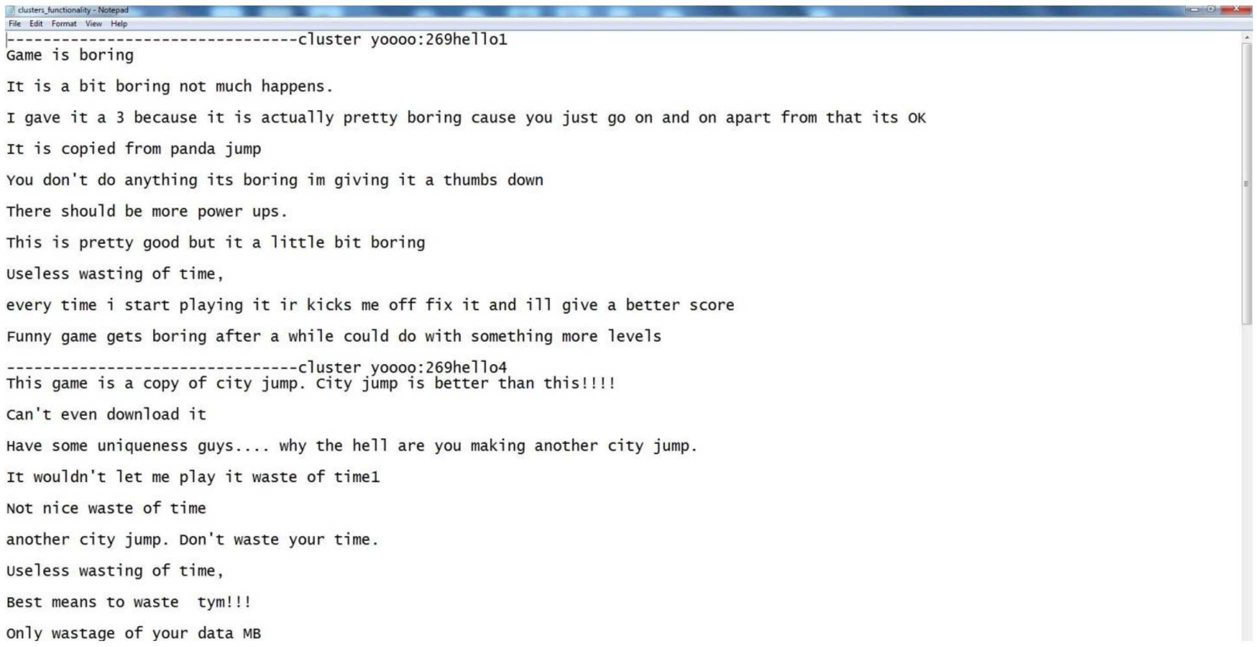
```
---------------------------------cluster yoooo:446hello3
Force closes constantly. Crap

Force closing for no reason....

It closes after I start it Automatically

Crashes and Force close in lollipop.....

Force closes before I get to play...

The game just closes n if U start it agn it still closes agn!

Force close comes

Force close comes

It force closes always. No use. Uninstalled

Forced closed all time...on my lg su660..slove it..

Dont download. It opens then force closes.
---------------------------------cluster yoooo:446hello4
Every time I try to play the game it just closes by its self

This game hangs on Android lollipop

Yea so i download the game and it loads but as soon as i click play it crashes please fix this for 5 star

First time install of this game, the app starts up, but within less than 5 seconds it just shuts down and exits itself, everytime, impos

I'm using a moto g 2nd gen.. If I open the game it just crashes . please fix this bug

It closes in the middle of the game

Hey, it keeps crashing on my LG G3

Game crashing on my Nexus 5 whenever I start it.. It just stays for 5 seconds and then its gone.. Please fix it!!
```

Fig 4.7: Performance Defect clusters



```
---------------------------------cluster yoooo:330hello1
Too many ads.

Too much adds

Cool but to many adds

so many ads

Toooo many big ads

To many ads  and bug fixes

Boring; to many ads

Too many ads.

Boring to much ADS. Do not download5
---------------------------------cluster yoooo:330hello3
I press the x button to cancel ads and then my tablet turns black I turn the app off turn it back on and another add pops up and I coul

Can't flip the screen ! My tablet screen is cracked and I can't play because it won't flip. Please fix it!

Not open on my device

I won't even show the screen any more!

Black screen not able to play.. pls remove this bug.

Every time I get an ad the game screen goes black

I don't know why, the screen is showed nothing but black on my Sony Xperia U device.

...Black screen...

It's not working on my phone.Black screen appears.

Its getting a black screen :-( on s3 why?
```

Fig 4.8: Usability Defect clusters



```
---------------------------------cluster yoooo:269hello1
Game is boring

It is a bit boring not much happens.

I gave it a 3 because it is actually pretty boring cause you just go on and on apart from that its OK

It is copied from panda jump

You don't do anything its boring im giving it a thumbs down

There should be more power ups.

This is pretty good but it a little bit boring

Useless wasting of time,

every time i start playing it ir kicks me off fix it and ill give a better score

Funny game gets boring after a while could do with something more levels

---------------------------------cluster yoooo:269hello4
This game is a copy of city jump. City jump is better than this!!!!

Can't even download it

Have some uniqueness guys.... why the hell are you making another city jump.

It wouldn't let me play it waste of time1

Not nice waste of time

another city jump. Don't waste your time.

Useless wasting of time,

Best means to waste  tym!!!

Only wastage of your data MB
```

Fig 4.9: Functionality Defect clusters

Silhouette Score

Comparison between approach 1 and approach 2:

We randomly took 120 reviews from different classes such as Performance based reviews, Usability based reviews and Functionality based reviews.

The output of clustering is:

using approach 1:

S. No	Cluster Number	Reviews
1	Cluster 1	Very slow . remove this game
		Very slow . remove this game
2	Cluster 2	Error 502
		Error 491
		Kept saying Error 502. Pls fix

3	Cluster 3	Crashes on android 5.0.1
		Crashes on android 5.0.1
		Crashed at first use do not installll

Table 4.3: clusters formed using approach 1

Using approach 2:

S. No	Cluster Number	Reviews
1	Cluster 1	Unable to download very slow!
		Takes for ever to download
		Rubbish game! Don't download
		Downloading sucks!
2	Cluster 2	Boring game, such a time pass
		Nice game, but it is boring
		Please increase its difficulty.. childish boring game
3	Cluster 3	Error 502
		Error 491
		Every time I open it, a lot of erros are there. Pls fix it.
		Boring game. Always kept saying a lot of errors. Developers go to hell!
		Errors errors errors…. What the hell
		Remove error. I m gonna delete your game.

Table 4.4 : clusters formed using approach 2

Table 4.3 and table 4.4 shows clustering of reviews some random sample of reviews from performance based, usability based and functionality based reviews using approach 1 and approach 2 respectively.

Results

The approach 2 is better than the approach 1 as in approach 2 the average silhouette score values for clusters are not extreme so it minimizes the possibility that some cluster has average silhouette score value as 1 which means repeating review. And also minimizes the possibility of average silhouette score value as -1 which means all points in cluster are misplaced.

4.3 Discussions

Here we are showing top k defects for the app "Ninja Jump". For this thesis we are showing top 5 defects.

Top 5 Performance Defects for "Ninja Jump"

S.No	App Name	Defects
1	Ninja Jump	Hey, it keeps crashing on my LG G3
2	Ninja Jump	Force closes constantly.
3	Ninja Jump	Poor download speed.
4	Ninja Jump	very slow
5	Ninja Jump	Crashes a lot, Force closes

Table 4.5 : Top 5 Performance Defects

Table 4.3 shows performance defects in the app “Ninja Jump”. The defects are Force closing of app, Frequent crashing, App is slow, downloading issues.

Top 5 Usability Defects for “Ninja Jump”

S.No	App Name	Defects
1	Ninja Jump	Too many ads.
2	Ninja Jump	gud game...bt should hve better graphics
3	Ninja Jump	Black screen not able to play.. pls remove this bug.
4	Ninja Jump	Good but where is exit button
5	Ninja Jump	Ads, Black Screen

Table 4.6: Top 5 Usability Defects

Table 4.4 shows usability top 5 usability defects in the app. These are too many ads, bad graphics, black screen and misplacement of exit button.

Top 5 Functionality Defects for “Ninja Jump”

S.No	App Name	Defects
1	Ninja Jump	Game is boring
2	Ninja Jump	Updates version sucks
3	Ninja Jump	This game is a copy of city jump. City jump is better than this!!!!
4	Ninja Jump	It is so hard because I fall every one minute then I get good at it and then my big sisters get geles

5	Ninja Jump	It is so boring after the first few runs.

Table 4.7: Top 5 Functionality defects

Table 4.5 shows functionality top 5 usability defects in the app. These are game is boring, updated version is not good, game is copy of city jump, and game is difficult.

4.4.1 Performance based defects

Some of the critical and most talked about performance based defects were Frequent Crashes in the app followed by Force closing of the app. Many users were also complain- ing about downloading problem, slow processing. Memory requirement was also a defect in ground truth with a weak support count but our approach didn't find it. Figure 6 shows the ground truth for performance reviews. Table 3 shows the predicted defects and acheived Accuracies. Overall Accuracy obtained was 71.57 %

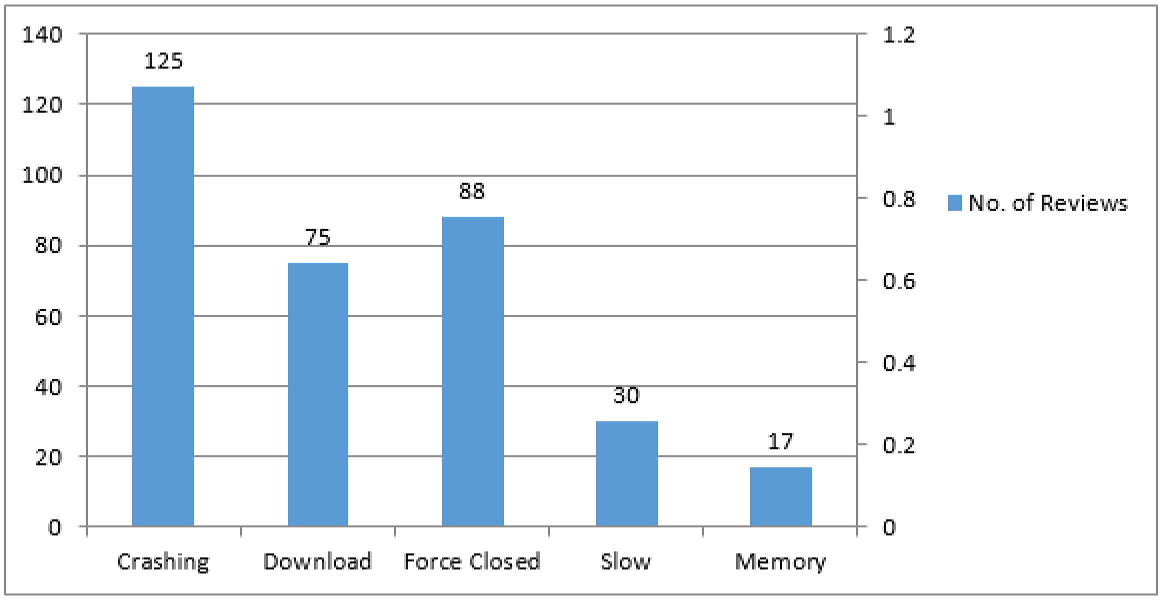


Fig 4.10: Ground truth for performance based defects

S. No	Defect Description	W_i	Matching Reviews	Total Reviews	Accuracy

1	Crashing	0.37	8	10	0.80
2	Force close	0.27	9	11	0.81
3	Download	0.22	6	8	0.75
4	Slow	0.09	4	10	0.40
5	Memory	0.05	0	9	0
Accuracy		71.57%			

Table 4.8: Performance Accuracy

A=0.37*0.80+0.27*0.81+0.22*0.75+0.09*0.40+0.05*0=0.7157

4.4.2 Usability based Defects

Frequent usability based defects found were a lot of advertisements followed by black screen problem in the app. Many users were also expressing their concern about bad graphics, non-availability of exit button. Figure 7 shows the ground truth for usability reviews. Table 4 shows the predicted defects and achieved Accuracies. Overall Accuracy obtained was 74.67 %

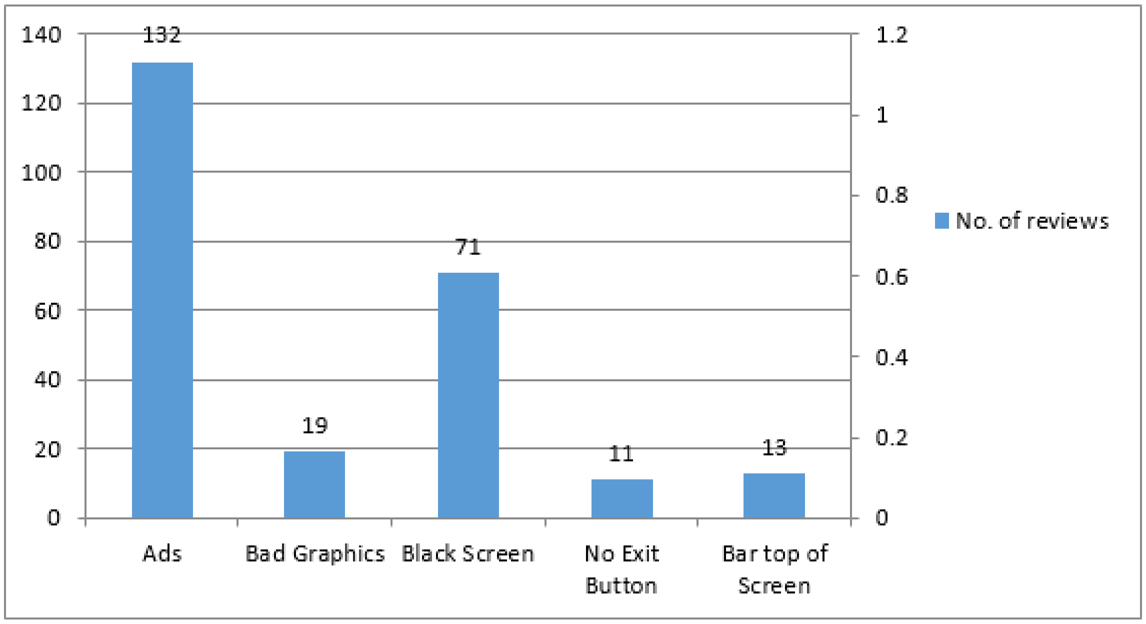


Fig 4.11: Ground truth for usability based defects

S. No	Defect Description	W_i	Matching Reviews	Total Reviews	Accuracy
1	Ads	0.53	8	9	0.88
2	Black screen	0.29	7	10	0.70
3	Bad Graphics	0.08	8	13	0.61
4	Exit Button	0.05	4	7	0.57
5	Bar	0.05	0	10	0
Accuracy		74.67%			

Table 4.9: Usability Accuracy

$$A=0.53*0.88+0.29*0.70+0.08*0.61+0.05*0.57+0.05*0=0.7467$$

4.4.3 Functionality based Defects

Many users found the app boring while some users found it very difficult to play. A lot of users were not happy with the updated version. While some user found it similar to another game ”City Jump” by quoting yet another city jump. Figure 8 shows the ground truth for usability reviews. Table 5 shows the predicted defects and achieved Accuracies. Overall Accuracy obtained was 77.04 %.

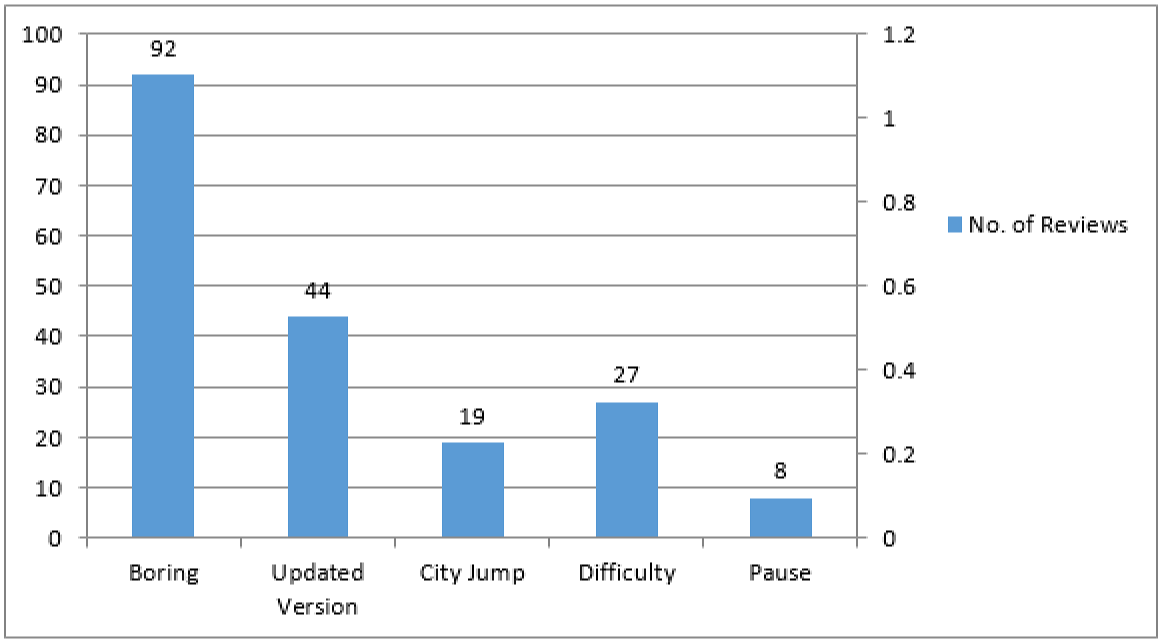


Fig 4.12: Ground truth for functionality based defects

S. No	Defect Description	W_i	Matching Reviews	Total Reviews	Accuracy
1	Boring	0.48	10	13	0.76
2	Updated Version	0.23	9	11	0.82
3	Difficulty	0.15	12	14	0.86
4	City Jump	0.10	7	8	0.88
5	Pause	0.04	0	6	0
Accuracy		77.04%			

Table 4.10: Functionality Accuracy

A=0.48*0.76+0.23*0.82+0.15*0.86+0.10*0.88+0.04*0=0.7704.

CHAPTER 5 CONCLUSION AND FUTURE WORK

User-generated reviews are indispensable repository for app developers. Since the user reviews are huge and messy, manual analysis is inapplicable. In this paper we propose a framework for the DIPMAP problem i.e. ”Did I Program A Poor Mobile App?”. We extracted user reviews and ratings of various apps then we converted every word in the reviews into n-dimensional vector representation using our word to vector conversion model . Then these word vectors were converted into review vectors using our proposed TF- mIDF technique. The reviews were classified into four broad categories i.e. performance based, usability based, functionality based and junk reviews. We removed junk reviews. Then we applied clustering on each of the above review class and we obtained various clusters which represent a defect. It gives the prioritized list of defects. We tested our model on various app from heterogeneous categories and found good accuracies. For ”Ninja Jump” app we obtained 71.57 % ac- curacy for performance based defects and 74.67 % accuracy for usability based defects and 77.04 % accuracy for functionality based defects. Two major advantages of our approach are 1) it captures the linguistic regularities and semantic relation between words 2) it is unsupervised and completely automated approach. In future a model for predicting the future ratings of the apps after these defects get removed in the subsequent version.

REFERENCES

[1] http://www.statista.com/statistics/266210/number-of- available-applications-in-the-google-play-store/

[2] http://www.androidcentral.com/gmail-now-has-over-1- billion-monthly-active-users

[3] https://play.google.com/store/apps/details?id=com. facebook.katanahl=endetails-reviews

[4] https://play.google.com/store/apps/details?id=com. truecallerdetails-reviews

[5] BIFET,A. AND FRANK, E. Sentiment knowledge discovery in twitter streaming data. In Discovery Science, Springer, 1-15,2010

[6] Guzman, Emitza, and Wiem Maalej. How do users like this feature? a fine grained sentiment analysis of app reviews. Requirements Engineering Conference (RE), 2014 IEEE 22nd International. IEEE, 2014.

[7] Chen, Ning, et al. AR-Miner: mining informative reviews for developers from mobile app marketplace. Proceedings of the 36th International Conference on Software Engineering. ACM, 2014.

[8] Fu, Bin, et al. Why people hate your app: Making sense of user feedback in a mobile app store. Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2013.

[9] Gao Cuiyun, et al. PAID: Prioritizing app issues for developers by tracking user reviews over versions. Software Reliability Engineering (ISSRE), 2015 IEEE 26th International Symposium on. IEEE, 2015.

[10] Gao, Cuiyun, et al. AR-Tracker: Track the Dynamics of Mobile Apps via User Review Mining. Service-Oriented System Engineering (SOSE), 2015 IEEE Symposium on. IEEE, 2015.
[11] Mikolov, Tomas, et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013).

[12] Chelba, Ciprian, et al. One billion word benchmark for measuring progress in statistical language modeling. arXiv preprint arXiv:1312.3005 (2013).

[13] Rong, Xin. word2vec parameter learning explained. arXiv preprint arXiv:1411.2738 (2014).

[14] Roelleke, Thomas, and Jun Wang. TF-IDF uncovered: a study of theories and probabilities. Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2008.

[15] Wilbur, W. John, and Karl Sirotkin. The automatic identification of stop words. Journal of information science 18.1 (1992): 45-55.

[16] http://scikit-learn.org/stable/modules/generated/ sklearn.metrics.silhouettescore.html

LIST OF PUBLICATION

[1] Ankur Tagra , Durga Toshniwal, “DIPMAP: Did I Program a Poor Mobile App?” in the IEEE International Conference on Data Mining ‘2016.