

Sentiment User Review Analyzer for Android Applications

Name of the Student : T I M P SILVA

Registration No : 169334E

Name of the Supervisor : Mr.SamindaPremaratne

M.Sc. in Information Technology
Department of Information Technology
University of Moratuwa

DECLARATION

I do declare that this thesis is my own work and has not been submitted in any form for another diploma or degree at any university or other institution of tertiary education. This thesis was prepared for degree of Master of Science/ Information Technology under the guidance and supervision of Mr. Saminda Premaratne. All information taken from the published or unpublished work of others has been acknowledged in the text and a list of references is shown.

Name of Student

T I M P Silva

Signature of Student

.....

Date:

Name of Supervisor

Mr. S C Premaratne

Signature of Supervisor

.....

Date:

ACKNOWLEDGMENTS

I would like to thank all who helped me to success this project. I would particularly like to single out my supervisor at the University of Moratuwa, Mr. SamindaPremaratne, Senior Lecture, I want to thank you for your excellent cooperation and for all of the opportunities I was given to conduct my project and further my thesis. When I was in trouble with the subject matters and personal matters, your guidance and encouragement make me to successfully complete this project.

I would also like to thank all the other lectures of the University of Moratuwa who have shared your knowledge throughout the whole period of study at university. And also giving my grateful thank to all my batch mates of 10th batch of MSc in IT. You encourage me every time when I was needed.

My colleague at the office, you gave me your support to evaluate the system taking your valuable time. Thank you.

ABSTRACT

Usage of mobile phones rapidly increased for the last decade globally. Introducing smart phones has led to a renaissance in mobile application development. Not only for make calls, send SMS phone to become smart so that it can be used for email access, web browsing, e-commerce, bill payments and entertainments etc. Because of this popularity application markets became massive. For the same purpose there are many applications deployed by different developers. Some apps can be considered as good, and some are not working as the user expect. And also there are many harmful applications which are stealing sensitive user information.

As Android applications have been used by many people, lots of issues have been already identified and many type of research are done on this topic. Even though there are many android applications available in the market, it is difficult to select the most suitable one before use. To get an idea about the application which is going to be used, user needs to read many reviews and ratings. This manual process is time consuming and not efficient.

In this project main aim is to develop a User Review analyzer which is gives a good analytical picture of the application before install.

Table of Contents

DECLARATION	i
ACKNOWLEDGMENTS	ii
ABSTRACT.....	iii
Table of Contents.....	iv
Table of Figures	vi
List of Tables	vii
CHAPTER 01	1
1.1. Introduction.....	1
1.2. Background and Motivation	2
1.3. Statement of Research Problem.....	2
1.4. Aim and Objectives.....	2
1.4.1. Main objectives of the project.....	3
1.5. The scope of the Project.....	3
1.6. Overview of the Report.....	3
CHAPTER 02	4
Review of Literature	4
2.1. Introduction.....	4
2.2. Android OS and Android Apps.....	4
2.3. Malware Detection.....	5
2.3.1. Static Analysis.....	5
2.3.2. Dynamic Analysis	7
2.3.3. Hybrid Analysis.....	8
2.4. User Reviews/Feedback Analysis	8
2.5. Current State	9
Chapter 3.....	10
Technologies used for developing the solution.....	10
3.1. Introduction.....	10
3.2. Back End/Server Side	10
3.3. Front End	10
3.4. Languages used.....	10
3.4.1. C# Language	10
3.4.2. Python	11
3.4.3. PHP Language:	11

3.4.4.	Java Programming Language	11
3.5.	Database.....	11
3.5.1.	MySQL:.....	11
3.6.	Tools and platforms used	12
3.6.1.	NLTK	12
3.6.2.	JSON	12
3.6.3.	Apache:	12
3.6.4.	Php Myadmin:.....	12
3.6.5.	Android Studio	12
3.7.	Summary	12
CHAPTER 04		14
Approach.....		14
4.1.	Introduction.....	14
4.2.	Data collection	14
4.3.	Data Analysing.....	14
4.3.1	Data pre-processing	14
4.3.2.	Feature Extraction.....	15
4.3.3.	Classification	15
4.4.	Visualization	15
4.5.	Process	16
4.6.	Summary	16
CHAPTER 05		17
Analyze and Design		17
5.1.	Introduction.....	17
5.2.	User Review extraction.....	18
5.3.	Sentiment Analysis	19
5.4.	Mobile Application	20
5.5.	Summary	21
CHAPTER 06		22
Implementation		22
6.1.	Introduction.....	22
6.2.	Extracting user Reviews from Google Play Store.....	22
6.3.	Analysing the Reviews	24
6.3.1.	Stop word removal.....	25

6.3.2.	Stemming	25
6.3.3.	Tokenization.....	26
6.3.4.	Feature extraction.....	26
6.3.5.	Sentiment Classification and calculation the confident	26
6.3.6.	Calculating most discussed features.....	27
6.4.	Web Service.....	29
6.5.	Mobile Application.....	29
6.6.	Summary	32
CHAPTER 07	33
Evaluation	33
7.1.	Introduction.....	33
7.2.	Evaluation of User Review Extractor	33
7.3.	Evaluation of Sentiment Analysis Module	34
7.4.	Evaluation of the Final Output.....	38
7.5.	Evaluation of Data Visualization Module.....	39
7.6.	Summary	39
CHAPTER 08	40
Discussion & Future Works.....		40
8.1.	Introduction.....	40
8.2.	Conclusion	40
8.3.	Future Works	40
References.....		42
Appendix A.....		44
A.1.	Image of the User Review Extractor.....	44
A.2.	Image of Extracted data	45
A.3.	First screen of mobile application.....	46
A.4.	Result screen of mobile application.....	47

Table of Figures

Figure 1: The Android component stack.....	4
Figure 2: Hybrid feature extraction.....	8
Figure 3: Sentiment Analysis Process.....	15
Figure 4: Top Level Design of the System	17

Figure 5: User Review Extraction Flow Chart.....	18
Figure 6: Sentiment Analysis Flow Chart.....	19
Figure 7: Mobile Application Flow Chart.....	20
Figure 8: Sample URL List.....	22
Figure 9: Code segment of expanding web page	23
Figure 10: Codes for Read Data from HTML.....	24
Figure 11: Information taken from CSV.....	25
Figure 12: Stop Words.....	25
Figure 13: Result of Tokenization	26
Figure 14: Feature Extraction	26
Figure 15: Codes for Classification	27
Figure 16: Result of Analysis	27
Figure 17: Codes for Classify to Selected Features	28
Figure 18: Loading App Names.....	30
Figure 19: Codes for Graphing	31
Figure 20: Codes for Bar Chart.....	31

List of Tables

Table 1 Sentiment result of 25 games	34
Table 2 Feature-wise positive negative ratio.....	35
Table 3 Sample of result.....	36

1.1. Introduction

Even though there are several operating systems available for smartphones such as iOS, Windows, RIM, Bada, Symbian, etc. Android is the most popular which has taken a market share of 87.7% when it is 12.1% for iOS as of 2017[1]. Android was introduced by Google for mobile devices such as smart phones, PDA and netbooks on 5th November 2007[2].

Because of this popularity of smartphones, application marketplaces such as Google Play and Apple's App Store have massively grown. These market places give an opportunity for developers to publish their own applications to the public and also users to download them to their devices directly in a convenient way.

Android app is a software application which runs on Android OS. Developers may download the Android Software Development Kit from the Android website. They may share developed apps through their websites or can publish on Android Market for free or for sale. Android apps are written in Java language. They first compiled to Dalvik executables run Dalvik virtual machine which is designed for mobile devices.

Some developers are using these technologies to develop good useful apps to the public while some are using to spread malware and harmful apps and steal sensitive data from the users.

If some user needs an application for a specific requirement, there are many applications to be considered. To get a good idea about the performance, weaknesses, advantages, etc. of the application, need to read many user reviews. It will be a time-consuming task. Although someone read many reviews it may be difficult to get a clear picture of the application. In this project, it is proposed to develop a User Review analyzer which gives a good analytical picture of the application.

1.2. Background and Motivation

Researchers have realized that spreading malware and attacks are significantly increased. Hence many studies have been done and many techniques have been introduced to detect malware and risk analysis under static analysis, dynamic analysis and hybrid analysis of Android apps.

In static analysis basically, features/data are extracted from AndroidManifest.xml file without executing the application. Permission and API calls are commonly used in this methodology. Features such as memory usage, CPU usage, network traffic, etc. are extracted while the application is running and then do the analysis in dynamic analysis. The hybrid analysis is a combination of static and dynamic analysis. Even though it could increase the accuracy of malware detection, it could make the system slower and high analysis process time-consuming.

From the above techniques, it can only detect malware and risk levels. But it is very important to measure the attributes like usefulness, performance, weaknesses, user satisfaction level, etc. To measure these attributes the best way is to do a sentiment analysis of user reviews. But in practice, it is very hard and time-consuming to read lots of reviews. In this proposed application user reviews will be analyzed and visualize these attributes so that user will be able to get a good idea about the application he/she is going to use, at a glance.

1.3. Statement of Research Problem

In this project, a mobile application has been developed to do a sentiment analysis for reviews of the mobile application.

1.4. Aim and Objectives

The aim of this project is to develop an android application to analyze user reviews of a selected android application and visualize graphically the result so that can have a good understanding of the application

1.4.1. Main objectives of the project

- Study on how to use NLP on sentiment analysis of user reviews.
- Propose a solution/method for the proper user review analysis.
- Design and develop an android application using the above method to solve the problem.
- Learn the basics of Android development and Python.
- Evaluate the system.
- Prepare the final report.

1.5. The scope of the Project

Considering the time limit the scope of the project is limited to selected applications of the game category of Google's Play store.

1.6. Overview of the Report

In this report, it is illustrated the development of the implementation based research carried on. Chapter 01 includes the introduction and a summary with an overview of the project. Chapter 02 contain the literature review done based on the topic. Chapter 03 has summarized the technologies of major areas. Chapter 04 is for the topic of approach. Chapter 05 includes the design and analysis of the research. Chapter 06 covers the implementation details and chapter 07 evaluate the methods used in the implementation and the results obtained with the current manual system. Finally, chapter 08 discusses the limitations and future developments for the solution.

Review of Literature

2.1. Introduction

This section includes a brief review of papers which have been discussed about android applications, malware detection and sentiment analysis of reviews. Here recently published papers were concerned since this area of study is frequently updating.

2.2. Android OS and Android Apps

Google entered into the mobile market by introducing a mobile platform called Android for mobile devices such as smart phones, PDA and netbooks on 5th November 2007[2]. Android is a software stack for mobile devices that includes an operating system, middleware, and key applications[3] run on Linux kernel.

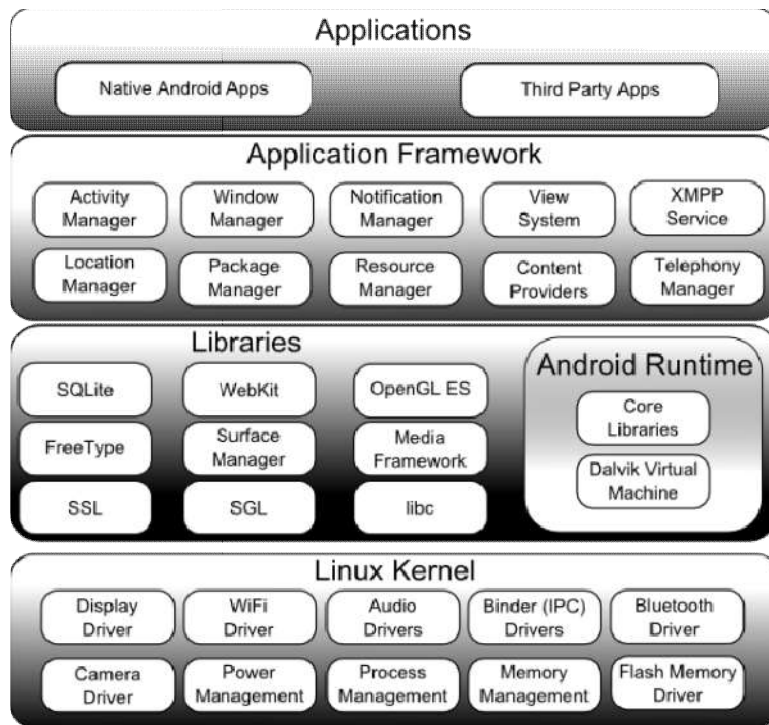


Figure 1: The Android component stack

Android architecture is consisting of a number of layers as Application, Application Framework, Libraries, Android Runtime and Linux Kernel as shown in Figure 1. The application layer is the top layer with a set of applications such as SMS

applications, email applications, games, Maps, browsers, etc. The Application Framework is a software framework which is used to implement a standard structure of an application for an operating system. The layer which is presented below the application framework consists of two parts. First part is Libraries which are all written in C/C++. They will be called through a Java interface. The second part is Android Runtime which consists of a set of core libraries that provides the most of the functionality in the core libraries of the Java programming language. The lowest layer is the Linux Kernel, for core system services such as memory management, security, network stack, process management, and driver model. The kernel acts as an abstract layer between the hardware and the rest of the software stack too [4].

Android app is a software application which runs on Android OS. Developers can download the Android Software Development Kit (Android SDK) from the Android website. They may share developed apps through their websites or can publish on Android Market for free or for sale.

Android apps are written in Java language. They are first compiled to Dalvik executables run on Dalvik virtual machine which is designed for mobile devices.

2.3. Malware Detection

Malware detection is done by analyzing application features which can be extracted directly or indirectly. This analysis can be categorized as static, dynamic and hybrid based on the features used. In static analysis, features such as permissions, API calls, etc. are analyzed which can be extracted from AndroidManifest.xml without running the app. Dynamic analysis, features are extracted while running the app such as Network traffic, IP addresses, Battery usage, etc. The other one is a hybrid analysis which is a combination of these two analyses.

2.3.1. Static Analysis

In static analysis basically, features/data are extracted from AndroidManifest.xml file without executing the application. Permission & API calls are commonly used in this methodology.

In this paper [5] a risk evaluation based system is proposed as a reference model for static analysis of Android application. In here they have categorized 2

phases as app pre-processing phase to collect and analyze. Here the user has to upload the software. The other phase is Model training phase to construct a feature pattern and converted to a downloadable format. Then it can be downloaded to the smartphone. Here authors have not mentioned the methodology properly other than mentioning the structure of the system.

There is a proposal for a security advisory system[6] with two parts.1.Risk Assessment of application 2.Decission support by presenting risk information.

In the first part, the system will read its permission information from the manifest file. Then from the combination of permissions, the assessment will be done with the pre-defined rule set. As example “INTERNET & READ PHONE STATE”: Stealing identity information.

Decision support part consists of four parts. I.e. Risk level, Risk Type, Reputation of the market, Critical comments.

They have collected 180 malware with different hash values and evaluated 261 applications. The rate of Correctly Detection of Malware is 95.6% while the Rate of Incorrectly Detection of Innocent Application as Malware 62.1%.

From the result, since incorrect detection is high, accuracy is not good enough. And also they have not mentioned the method of defining rules. If they analyze the comments sentimentally they could increase the accuracy of the system. However, consideration of user reviews is a good point.

A study was done by C. Guo, J. Xu, L. Liu, and S. Xu[7] to design a reasonable model and find out appropriate parameters for rank risk of Android applications. Authors have found that, only from the number of permissions, malware cannot be determined. Based on permission combinations, a new “association statistics” approach which is called as FAS (Frequent Association Seed) algorithm has been proposed. There are four steps

a) Extract the permission combinations by an improved association classification algorithm;

b) Reduce the redundancy computation by using the “support discount” and corresponding algorithms.

c) Balance2 referential criterions "benign seed" and "malicious seed".

d) Computethe final scores.

In this research, a large data set was used and done good analyze. In this paper, it was found that some benign apps also get high malicious score due to having some sensitive permission. In this paper also, only permission combinations have been considered.

Permission, API calls, Strings extracted, Native commands, XML elements, Metadata, etc., can be identified as most used features for static analysis.

2.3.2. Dynamic Analysis

Features such as memory usage, CPU usage, network traffic, etc. are extracted while the application is running and then do the analysis in dynamic analysis.

Dynamic analysis has been done here with latent network behavior of android application[8]. After the extraction of the Android malware network feature such as IP addresses, real-time DNS response information, Time zones etc. authors could develop the Apps-Geo matrix (APG) that is an $m \times n$ matrix, where m is the number of collected Android apps. Then APG has been decomposed as a linear mixture of l , with the weights in the columns of the hidden matrix H , using Independent Component Analysis (ICA). Finally, an automatic Android malware detection mechanism was proposed based on the results from Android sandbox. We extracted network spatial features of Android apps and used independent component analysis (ICA) to determine the intrinsic Android malware domain name resolution communication behavior.

The number of features proposed by Kim and Choi[9] was 59, which were extracted on the basis of the Linux kernel in the Android. Most of the features were divided into 3 kinds such as CPU and network, memory. The analysis system was composed of a monitoring agent and analysis server, and a feature extraction module in the monitoring agent extracts features selected thru the feature selection and then sends the collected data to the data management module. Features were periodically extracted for every 10 seconds. It was found that the performance was gradually improved with removing unnecessary features one by one. As the final result of feature selection, 23 features out of 59 were removed, 36 had been selected. Authors have shown that this feature selection mechanism improved the accuracy of malware detection.

In dynamic analysis, API calls, network traffic, power usage, CPU usage, process ids, SMS usage, etc. can be identified as the most used features.

2.3.3. Hybrid Analysis

The hybrid analysis is a combination of dynamic and static analysis. Although it could increase the accuracy of malware detection, it could make the system slower and high analysis process time-consuming.

Hybrid model based on AdaBoost, J48 to classify applications was proposed by Vajdi, Torkaman, and Bahrololoum[10]. Several features from different applications collected by OCS lab were extracted. Figure 2 shows the proposed mechanism.

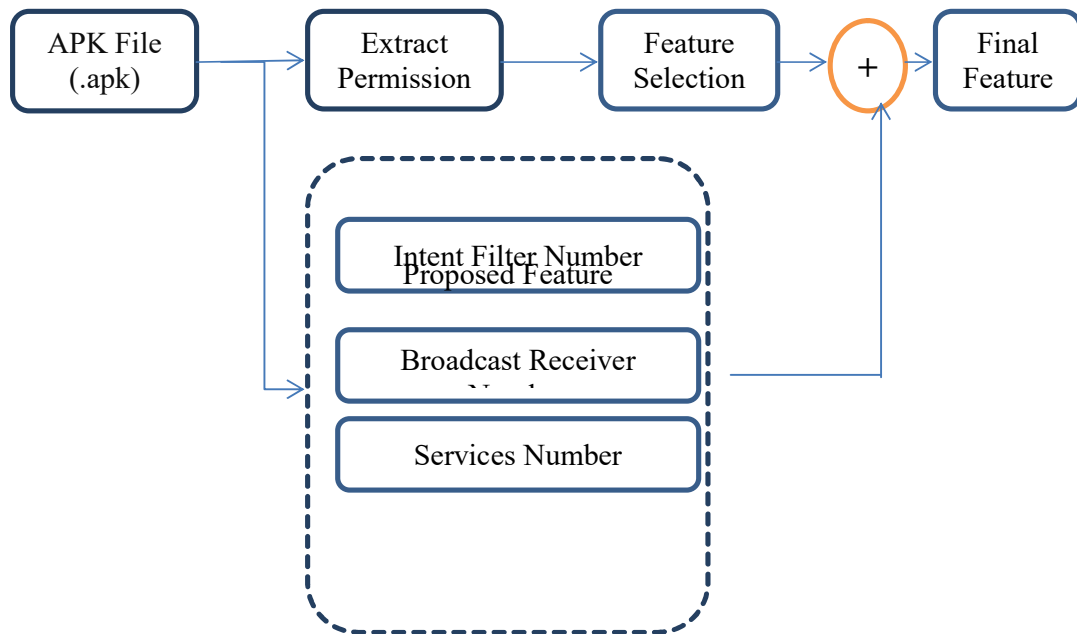


Figure 2: Hybrid feature extraction

2.4. User Reviews/Feedback Analysis

Only a few types of research have been done in this area. Among them, most researches focused on developers view. They have analyzed the user reviews so that to support developers to make decisions.

Sherlock A. Licorish and team have shown[11] that the Android development team generally responded to the top enhancement requests made by the Android community, suggesting user-focused design is necessary for Online application distribution platforms (OADPs) and in environments where market-driven requirements engineering (MDRE) is used as a strategy for developing a product.

They have also shown that developers should evolve applications with particular features more attention than others. This research is only focused on developers end. They did not consider how to give any benefit to the end users by analyzing other user's reviews.

In the examining of consumer reviews in google play done by Elizabeth Ha and David Wagner[12] discovered that the reviews in Google Play consist useful information about Android applications, but most information is about the quality of the Android application and not about security & privacy concerns. However, they found that a small number of users who questioned the use of permissions in applications, indicating that there is a chance to educate and address such concerns in android application markets. This study provides very useful information for proposed research in this paper such as about what kind of things users generally write.

2.5. Current State

Researchers have done many studies on malware detection and risk analysis for Android applications. Many mechanisms and algorithms have been proposed under static analysis, dynamic analysis, and hybrid analysis. Some features like permissions, native commands, API calls, network traffic, and CPU usage, etc., have been identified as the most relevant features for malware detection. Some studies have been done for review analysis but centralized to developers.

Technologies used for developing the solution

3.1. Introduction

This chapter illustrates the methodologies and techniques which have been used to develop the system. To implement this system in more attractive and efficiently, few tools, technologies were used. The details of the implementation environment, hardware infrastructure, development strategies, tools used, database and the Application development architecture are described in this chapter.

It is possible to categorize the system into 2 main parts as below.

3.2. Back End/Server Side

Since the solution is based on web serviced development with centralized client-server architecture, identified minimum hardware and software requirement for hosting the server are listed as follows,

- Intel Core I3 processor
- RAM 4GB
- 50GB HDD
- Network speed 1Gb Ethernet
- Window OS with WAMP
- PHP version 7
- My SQL 5
- Internet with static IP

3.3. Front End

Mobile phone with Android or later.

Internet connectivity

3.4. Languages used

3.4.1. C# Language

C# was used as the programming language and Microsoft Visual Studio was used as the developing environment to develop data extraction application.

3.4.2. Python

Python was used as the programming language to developing data analyzing part. Python is very powerful and easy to use language. And the natural language processing tool kit is supported to python. So python is the best language to this kind of programming. IDLE is the selected IDE for python developments.

3.4.3. PHP Language:

Hypertext Pre-processor (PHP) is used to develop web services since PHP is server-side scripting.

3.4.4. Java Programming Language

Java is a general-purpose computer-programming language that is, object-oriented, concurrent, class-based and specifically designed to have as few implementation dependencies as possible. It is aimed to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can be run on all platforms that support Java without recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM). Android development typically uses Java language, so Java was used to code mobile application in this project.

3.5. Database

The database is used to keep information about android application names and analysis results.

3.5.1. MySQL:

MySQL is an open source relational database management system which is running on virtually all platforms, including Windows, Linux & UNIX. MySQL is more often associated with online publishing & web applications.

MySQL is based on a client-server model. MySQL comes with WAMP for windows platform.

3.6. Tools and platforms used

3.6.1. NLTK

NLTK (Natural Language Toolkit) is a leading powerful platform for developing Python programs to interact with human language data. It provides easy-to-use interfaces to over 50 lexical and corpora resources such as WordNet, along with a suite of text processing libraries for tokenization, classification, stemming, tagging, parsing, & semantic reasoning, wrappers for industrial-strength NLP libraries, & an active discussion forum. So NLTK was used to do sentiment analyze.

3.6.2. JSON

JSON stands for JavaScript Object Notation. JSON is a very light-weight format for transporting & storing data. So JSON was used to transport data from server to mobile application via web services.

3.6.3. Apache:

Apache is the most widely used web server software and Apache 2 was used as the web server of this project. Apache is open source freely available software which is a fast, secure and reliable. It can be customized to meet the needs of many different environments by using modules and extensions.

3.6.4. PhpMyadmin:

This is an open source tool to query the database values and execute SQL statements.

3.6.5. Android Studio

Android Studio is the official integrated development environment (IDE) for Android OS, built on JetBrains' IntelliJ IDEA and designed for Android development. So Android Studio was used to develop the mobile application to visualize the result.

3.7. Summary

In this chapter, it was discussed about the technologies used in the project. C#, Python, and Java are identified as main programming languages. Visual Studio, IDLE and Android studio are the main IDEs for developing. MySQL is the database engine used.

Next chapter has described the approach taken including data collection, analyzing and visualizing.

Approach

4.1. Introduction

In this chapter, it is discussed the approach taken to implement the system. The system basically can be divided into 3 modules as data collection, data analyzing and visualizing the result.

4.2. Data collection

Here the data set is user reviews in google play store. Hence User reviews need to be extracted. Since the reviews are in online it is possible to trace HTML tags. By using this technique, a software tool will be developed to extract user reviews from the web and store them in a CSV file.

4.3. Data Analysing

In this project data analyzing takes the main role since it is very important to give the correct result. Sentiment analysis is the process of computationally identifying and categorizing opinions expressed in a text, in order to determine whether the user's opinion of a particular topic, product, etc. is positive, negative, or neutral. To do sentiment analysis, NLP (natural language processing) tools are used. So here NLTK was used. Data analyzing can be divided as data pre-processing, feature extraction and classification.

4.3.1 Data pre-processing

In general user reviews in the web are noisy. That is while some data are not informative and they are not meaningful. Some data cannot be analyzed since they contain garbage such as symbols only. Further, on words level, many words in the text don't have an effect on the general meaning of it. These words make the scale of the problem high and so the classification is more difficult since each word in the text is treated as one dimension. Here is the hypothesis of having the data well pre-processed to reduce the noise in the text may help to improve the performance of the classifier. And also speed up the classification process, therefore aiding in real time sentiment analysis.[13]

Data pre-processing contains white space removal, expanding abbreviation, stemming, stop words removal, negation handling and finally feature selection.

4.3.2. Feature Extraction

The main step in typical sentiment analysis is to convert raw text into features, which give a machine learning model a simpler, more focused view of the text. The Feature Extraction task aims to identify the entities being referred to [14]. In this step features such as graphics, cost, performance, risk, etc. are identified. For this NLP tools were used.

4.3.3. Classification

The Classification task aims to classify the opinions' polarity (positive, negative and neutral) regarding the features being used on [14].

4.4. Visualization

Finally, the result of analyze will be visualized graphically so that the user can get a good idea about the application. Here the android application is implemented. By the use of a pie chart the result of sentiment analysis that shows how many negative comments and how many positive comments available. And the aid of bar charts how many people talk about specific feature is displayed.

Figure 3 shows the process of sentiment analysis.

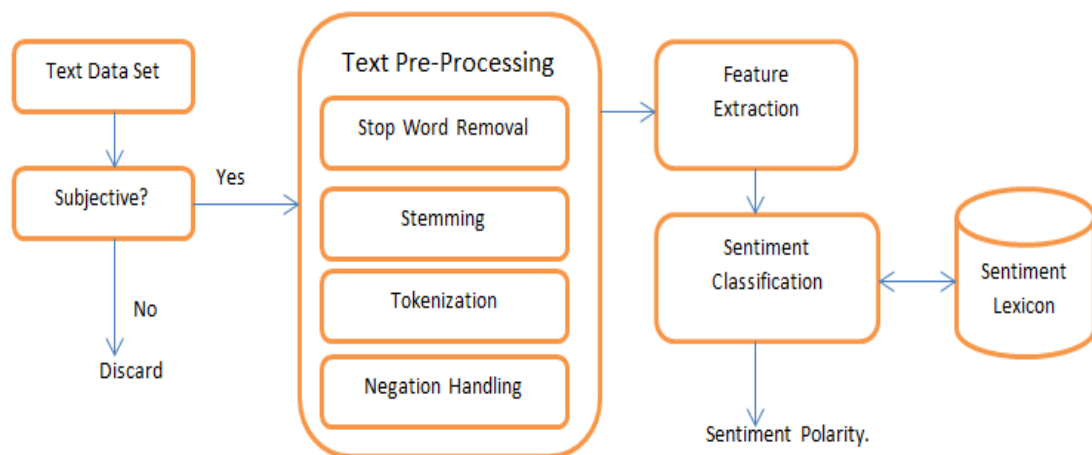


Figure 3: Sentiment Analysis Process

4.5. Process

As the first step user reviews are extracted from google play store to the data files as CSV. Then those data files read one by one and do the analytical process and store the necessary information to the MySQL database. As the final step from the mobile application, the user can select the application name and get the result thru the internet and visualize it in the screen.

4.6. Summary

In this chapter, it was discussed the approach of the system. This system based on online. The system extracts reviews from the cloud and saves to disk files. Then analyze and save to the database. Hence the user is facilitated to see the result thru the application. Next chapter will be discussed analyze and the design of the system.

Analyze and Design

5.1. Introduction

In this application, user reviews of android application are analyzed sentimentally. Hence the data set is the user reviews of the application, and the collecting data is taken a considerable part of this project. Collecting data is thru an automated application that extracts user reviews. After extracting the user reviews, they are to be stored in a database. Then the application analyzes the database on NLP methods for the result. The final result is displayed as the user can understand easily.

Basically, android applications can be categorized into two main categories, app, and games. Apps can be divided into sub-categories as Art & Design, Auto & Vehicles, Beauty, Dating, Events, Food & Drink, House & Home, and Parenting. Games have been divided as Action, Arcade, Board, Adventure, Card, Casual, Casino, Educational, Puzzle, Music, Racing, Simulation, Role Playing, Strategy, Sports, Trivia, & Word.

Considering time constrain the scope of the project is limited to the game category of android applications. Top level design of the system is shown in Figure 4.

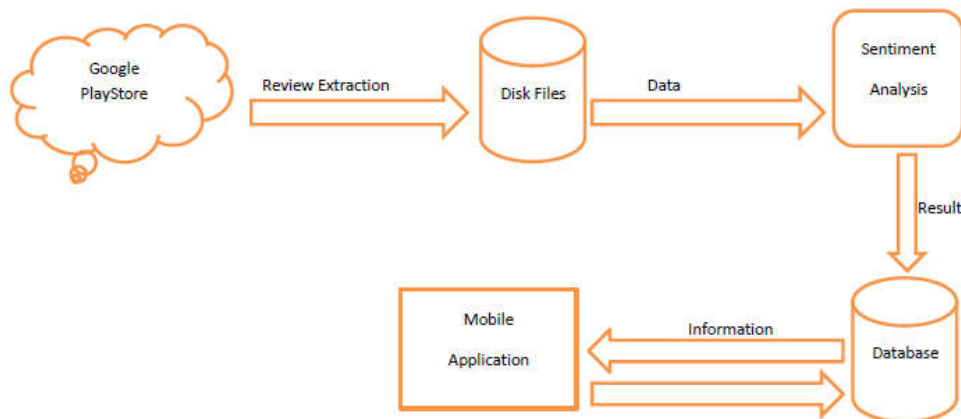


Figure 4: Top Level Design of the System

As described earlier this system mainly can be divided into 3 parts as Data collection, data analyzing, and result visualizing.

5.2. User Review extraction

Google is not exposing all the reviews for the public to download at once. Hence other method had to be considered. In this project reviews of the application get by reading HTML tag in the HTML page. Here the automated system is to load HTML page to the application and expand all the reviews. Otherwise, only a few data would be a load to the application. Figure 5 describes the flow of the module.

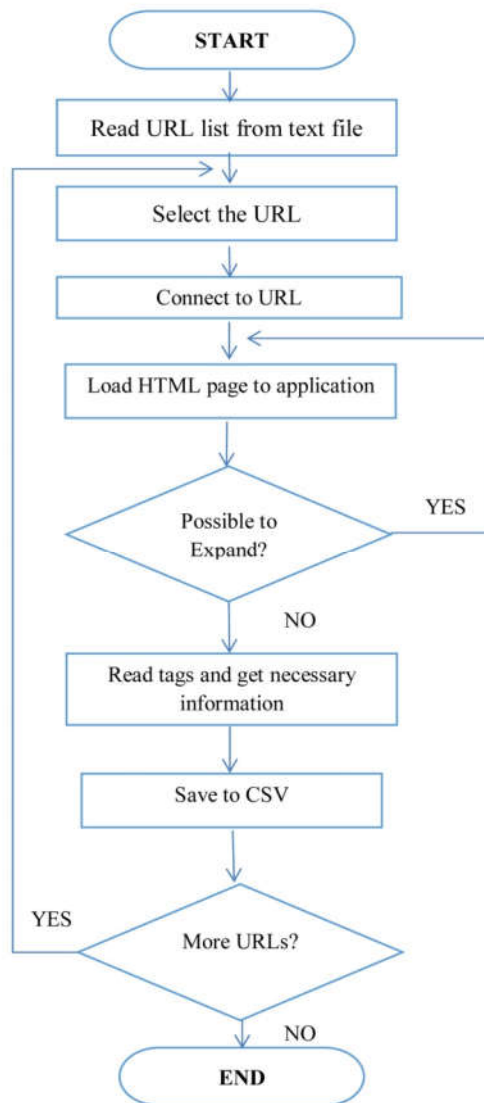


Figure 5: User Review Extraction Flow Chart

5.3. Sentiment Analysis

This module is doing all the analytics. First, it reads data from CSV files then do sentiment analysis and find most discussed features. Finally, all the results save to database so that mobile application can access this information via webservices. Figure 6 illustrates the flow of this module.

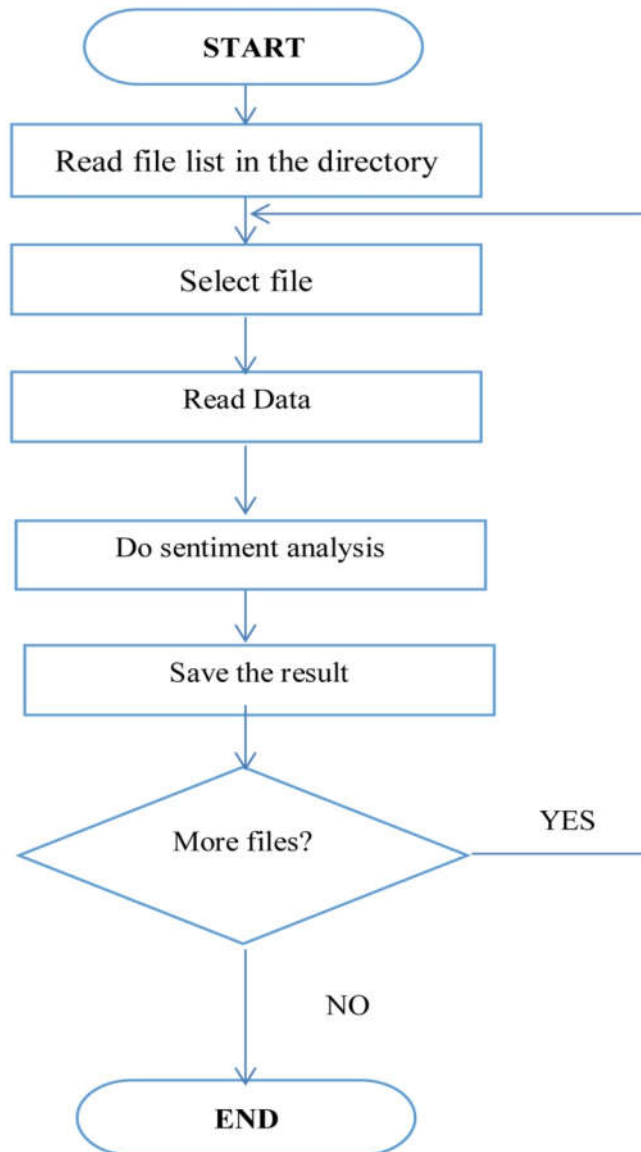


Figure 6: Sentiment Analysis Flow Chart

5.4. Mobile Application

The mobile application is the user end application to select the mobile application and see the analytical results. Here Android application designed runs on Android OS. The user needs to type or enter the desired application name then click on the button. Then the Android application connects to the data server via web services and gets the information and display the result graphically with the aid of pie and bar charts. Figure 7 is the flow chart for this module.

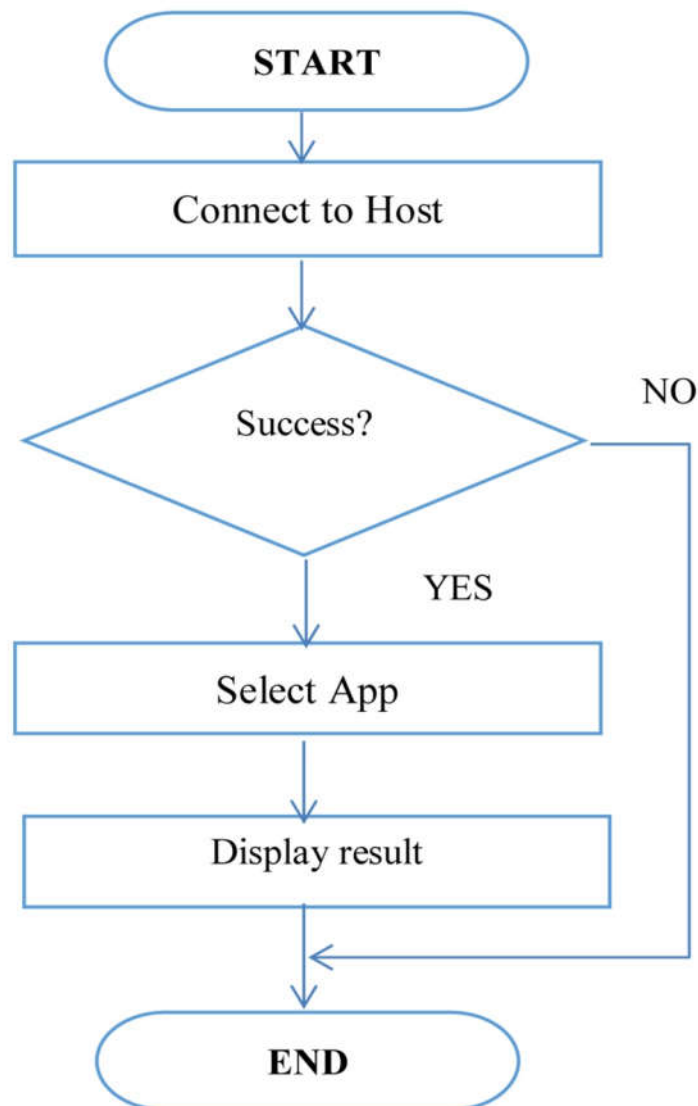


Figure 7: Mobile Application Flow Chart

5.5. Summary

This chapter described the top level design the aid of flow charts. Development of all modules including the Sentiment Analyzer for Android application was mainly based on this design. Next chapter discussed the implementation of the system.

Implementation

6.1. Introduction

This chapter describes the implementation of the system's modules, functions and process. As the system design is described in the above chapter the implementation is described here accordingly. Three modules user review extraction, sentiment analysis and mobile application along with the web service have been described in this chapter.

6.2. Extracting user Reviews from Google Play Store.

According to this project data set is the user reviews of Games in Google Play Store. After many studies found that there is no way to get data set directly from the Play Store. Initially it was planned to get the data set manually copy and paste. But it was very time consuming and not efficient. Hence, it was required to find a way to extract a large data set efficiently. Getting data set became more challenging.

Finally, after some R&D works, it was realized that from HTML tags, user reviews could be extracted. Once the web page is loaded to the browser; from the source code, all elements can be read. From this technique, using C#, a tool was developed to read all reviews.

As the first step, all the URLs manually copied to a text file named url_list.txt as shown in figure 8.

```
https://play.google.com/store/apps/details?id=com.nextwave.wcc2
https://play.google.com/store/apps/details?id=com.firsttouchgames.dls3
https://play.google.com/store/apps/details?id=com.ea.gp.fifamobile
https://play.google.com/store/apps/details?id=com.nautilus.RealCricket3D
https://play.google.com/store/apps/details?id=com.firsttouchgames.story
https://play.google.com/store/apps/details?id=com.touchtao.soccerkinggoogle
https://play.google.com/store/apps/details?id=com.generagames.soccerstarleaguesfootball
https://play.google.com/store/apps/details?id=com.firsttouchgames.smp
https://play.google.com/store/apps/details?id=com.mobirix.shotingking
https://play.google.com/store/apps/details?id=com.nextwave.wcc_1+
```

Figure 8: Sample URL List

Once the application starts all the list of URLs is loaded to the dropdown list. What the user needs to do here is click on the read button only. Then the application read all the reviews from listed URLs one by one automatically and saved as .csv files to the

disk. This took considerable time since the program had to expand all the data in the web page and reload to the application. For this timer object was used and code segment inside the timer as shown in figure 9.

```
y+= 1000;
HtmlDocument htmlDoc = webBrowser1.Document;
if ( 0<htmlDoc.All.Count && htmlDoc.All.Count < 220000)
{
    c++;
    Thread.Sleep(100);
    try
    {
        if (webBrowser1.Document.Body.ScrollRectangle.Height < y)
        {
            webBrowser1.Document.Window.ScrollTo(0, webBrowser1.Document.Body.ScrollRectangle.Height);
            ExpandPage();
        }
        else
            webBrowser1.Document.Window.ScrollTo(0, y);
        Thread.Sleep(100);
        return;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
}
```

Figure 9: Code segment of expanding web page

After loading all information of the web page application itself save necessary information to pre-defined location as CSV files. For this, each and every HTML element was read and checked the class name and got the inner text to appropriate variables. Then those values in the variables were saved accordingly to CSV files on the disk.

```

foreach (HtmlElement el in htmlDoc.All)
{
    strEle = el.GetAttribute("className");
    if (strEle == "AHFaub")
    {
        strAppName = el.GetAttribute("InnerText");
        strAppName = Regex.Replace(strAppName, "[^a-zA-Z0-9]", "_");
    }
    if (strEle == "p2Tk0b")
    {
        strDate = el.GetAttribute("InnerText");
    }
    if (strEle == "X43Kjb")
    {
        strUser = el.GetAttribute("InnerText");
    }
    if (strEle == "nt2C1d")
    {
        int pFrom, pTo;
        strRate = el.GetAttribute("InnerHtml");
        pFrom = strRate.IndexOf("Rated ") + "Rated ".Length;
        pTo = strRate.IndexOf("stars");
        strRate = strRate.Substring(pFrom, pTo - pFrom);
    }
}

```

Figure 10: Codes for reading Data from HTML

6.3. Analysing the Reviews

The next application is to analysis the data. Here Python language was selected with NLTK libraries. NLTK is a very powerful NLP library available for free to download. It was the biggest advantage to select this technology for this project. Also there are many tutorials for how to use NLTK for sentiment analysis with sample codes.

As in design chapter sentiment analysis contains the following steps.

1. Stop word removal
2. Stemming
3. Tokenization
4. Feature extraction
5. Sentiment classification

As the first step in this module, the information in the CSV files was read to python application for analyzing one by one. A sample of information taken from the CSV file is shown in below Figure 11.


```

UserName->      zoomary58
Rank->          5
Review ->       the only thing i don't like are the farm animals that give
bacon which means they would actually be killed. i haven't eaten a mammal since my
25th birthday and i'm 61 in march now. at least i try to make my city county schoo
ls farm and zoo to look realistic! i wish the humans weren't way bigger then the h
ouses and buildings and most have their set up just full of colors and almost all
don't have real looking places... i love everybody anyway i just hope it all gets
better!

```

Figure 11: Information taken from CSV

6.3.1. Stop word removal

Before sentiment analysis first step is to remove all stop words. In NLTK libraries there are methods to filter out all stop words. After filter the stop words in the text we can remove them. Figure 12 shows the output of the stop words of the nltk.corpus library.

```

stop words
{'my', "doesn't", 'have', 'd', 'only', 'any', 'of', 'her', 'were', 'on', 'these',
'against', "mightn't", "hasn't", 'themselves', 'ma', 'just', 'what', "won't", 'll',
'so', 'them', 'they', 'by', 'mustn', 'too', 'below', 'some', 'after', 'down', 'i
s', 'our', 'over', 'your', 'being', 'other', 'herself', 'are', 'for', 'isn', 'i',
'same', 'has', 'you', 'does', 'during', 'wouldn', '.', 'because', 'as', 'mightn',
'aren't', 'hasn', 'the', 'haven', 'a', 'if', 'be', 'here', 'at', 'but', 'such', 't
o', 'out', 'when', 'we', "needn't", 'shouldn', 'again', 'whom', "you're", 'hers',
's', 'myself', "that'll", 'no', 'its', 'doesn', 'why', 'needn', 'from', 'had', 'of
f', "she's", 'before', "you've", 'aren', 'am', 'ain', 'shan't', 'yourselves', 'his
', 're', 't', 'their', 'and', 'while', 'nor', "couldn't", 'most', 'won', "don't",
'it's", 'that', 'those', 'can', 'will', "haven't", 'me', 'who', 'than', 'weren', '
theirs', 'through', 'him', 'both', 'y', 'did', 'o', "shouldn't", 'few', 'wasn', 'w
ith', 'until', 'been', 'm', 'between', "should've", 'yours', "wouldn't", 'she', 'e
ach', 'where', "you'll", 'he', 'which', 'under', 'further', 'ours', "hadn't", 'the
re', 'should', 'an', "weren't", 'or', 'how', 'not', 'do', 'very', 'now', 'itself',
'don', 'in', 'ourselves', "isn't", 'into', 'above', 'himself', 'own', 'shan', "you
'd', 'all', 'yourself', 'then', 'more', 'this', 'didn', 'once', 'having', 'up', 'w
as', "wasn't", 've', "didn't", 'couldn', 'it', 'doing', 'about', "mustn't", 'hadn'
}

```

Figure 12: Stop Words

6.3.2. Stemming

Stemming is the process of grouping words to one which has the same meanings. For example 'dog', 'dogs', 'doggy' includes to dog. This is also done inside the libraries.

6.3.3. Tokenization

Tokenization means break the text into words. The library 'nltk.tokenize' must be imported to use the method 'word_tokenize'. From this function, it is possible to tokenize the text. Below Figure 13 shows a set of words after tokenizing the above-mentioned review.

```
After Tokenised
['the', 'only', 'thing', 'i', 'do', "n't", 'like', 'are', 'the', 'farm', 'animal',
's', 'that', 'give', 'bacon', 'which', 'means', 'they', 'would', 'actually', 'be',
', 'killed', '.', 'i', 'have', "n't", 'eaten', 'a', 'mammal', 'since', 'my', '25t',
'h', 'birthday', 'and', 'i', "'m", '61', 'in', 'march', 'now', '.', 'at', 'least',
', 'i', 'try', 'to', 'make', 'my', 'city', 'county', 'schools', 'farm', 'and', 'z',
'oo', 'to', 'look', 'realistic', '!', 'i', 'wish', 'the', 'humans', 'were', "n't",
', 'way', 'bigger', 'then', 'the', 'houses', 'and', 'buildings', 'and', 'most', 'i',
'have', 'their', 'set', 'up', 'just', 'full', 'of', 'colors', 'and', 'almost', 'a',
'll', 'do', "n't", 'have', 'real', 'looking', 'places', '...', 'i', 'love', 'ever',
'ybody', 'anyway', 'i', 'just', 'hope', 'it', 'all', 'gets', 'better', '!']
```

Figure 13: Result of Tokenization

6.3.4. Feature extraction

Following figure 14 shows the code for feature extraction

```
def find_features(document):
    words = word_tokenize(document)
    features = {}
    for w in word_features:
        features[w] = (w in words)

    return features
```

Figure 14: Feature Extraction

6.3.5. Sentiment Classification and calculation the confident

The final step is to classify the sentiment and calculate confidence. To do this following code were used as shown in figure 15.

```

class VoteClassifier(ClassifierI):
    def __init__(self, *classifiers):
        self._classifiers = classifiers

    def classify(self, features):
        votes = []
        for c in self._classifiers:
            v = c.classify(features)
            votes.append(v)
        return mode(votes)

    def confidence(self, features):
        votes = []
        for c in self._classifiers:
            v = c.classify(features)
            votes.append(v)

        choice_votes = votes.count(mode(votes))
        conf = choice_votes / len(votes)
        return conf

```

Figure 15: Codes for Classification

Next, in Figure 16 show the resulted filtered sentence after stop words removal and the result of sentiment analysis with the confidence level.

```

filtered_sentence
['thing', 'n't', 'like', 'farm', 'animals', 'give', 'bacon', 'means', 'would', 'ac
tually', 'killed', 'n't', 'eaten', 'mammal', 'since', '25th', 'birthday', 'm', '6
1', 'march', 'least', 'try', 'make', 'city', 'county', 'schools', 'farm', 'zoo', '
look', 'realistic', '!', 'wish', 'humans', 'n't', 'way', 'bigger', 'houses', 'buil
dings', 'set', 'full', 'colors', 'almost', 'n't', 'real', 'looking', 'places', '..
.', 'love', 'everybody', 'anyway', 'hope', 'gets', 'better', '!']
[('n't', 4), ('farm', 2), ('!', 2), ('thing', 1), ('like', 1), ('animals', 1), ('g
ive', 1), ('bacon', 1), ('means', 1), ('would', 1), ('actually', 1), ('killed', 1)
, ('eaten', 1), ('mammal', 1), ('since', 1)]
sentiment -> pos : confident -> 1.0

```

Figure 16: Result of Analysis

6.3.6. Calculating most discussed features.

Apart from the sentiment analysis most discussed features were calculated in this section.

Feature list categorized as follows and then calculate how many users have been talking about these features so that we can get a good idea about the specific game.

Feature List

1. Speed.
2. Addiction

3. Satisfaction
4. Money
5. CPU/Memory
6. Graphics

To find the total number of reviews talk about these features, first remove the stop words then tokenize and last classified. Following Figure 17 shows the code segment.

```

stop_words = set(stopwords.words("english"))
stop_words.add('.')
words = word_tokenize(strText)
filtered_sentence = [w for w in words if not w in stop_words ]
all_words = []
for w in filtered_sentence:
    all_words.append(w.lower())
all_words = nltk.FreqDist(all_words)
for w in words :
    if w in list_speed:
        feature_speed=feature_speed+1
    if w in list_addictivity:
        feature_addictivity=feature_addictivity+1
    if w in list_satisfaction:
        feature_satisfaction=feature_satisfaction+1
    if w in list_money:
        feature_money=feature_money+1
    if w in list_memory:
        feature_memory=feature_memory+1
    if w in list_graphics:
        feature_graphics=feature_graphics+1

```

Figure 17: Codes for Classify to Selected Features

After doing these all analysis and calculations the result is to be saved in MySQL database.

Next step was semantic analysis. Here each review was analyzed for semantic so that to find the sentiment for each feature of review. Each review was split into an array of sentences. Then feature was found and the sentiment of the sentence was calculated. Figure 18 shows the python code for this analysis.

```

for sent in sents:
    senti= s.sentiment(strText)[0]
    words = word_tokenize(strText)
    print("After Tokenised ")
    print(words)
    filtered_sentence = [w for w in words if not w in stop_words ]
    print("filtered_sentence ")
    print(filtered_sentence)
    all_words = []
    for w in filtered_sentence:
        all_words.append(w.lower())
    all_words = nltk.FreqDist(all_words)
    print(all_words.most_common(15))

    for w in words :
        if w in list_speed:
            feature_speed=feature_speed+1
            senti_speed= senti
        if w in list_addictivity:
            feature_addictivity=feature_addictivity+1
            senti_addictivity= senti
        if w in list_satisfaction:
            feature_satisfaction=feature_satisfaction+1
            senti_satisfaction= senti
        if w in list_money:
            feature_money=feature_money+1
            senti_money= senti
        if w in list_memory:
            feature_memory=feature_memory+1
            senti_memory= senti
        if w in list_graphics:
            feature_graphics=feature_graphics+1
            senti_graphics= senti
    sentiment = res[0]
    confident = float(res[1])

```

Figure 18 Sentiments for Each Feature.

6.4. Web Service

The resulted data is sent to the mobile application from the data server via web service. Here data is encoded to JSON array and then send.

6.5. Mobile Application

The android application was built to user level communicate with the data server. This application facilitates users to select a game and see the analytical results graphically on the mobile screen.

Android studio was used as the Integrated Development Environment of this application. The android application communicates with the data server thru web services described above. Java is the language used for this development

Once the application starts the connection to the server is created. Then when the user clicks on the text field named 'Please select application' all the application names will be displayed. While the user enters the character, the name list will be filtered. Figure 19 shows the code for getting the app name to the list from the JSON object.

```
public class GetAppInfo {
    public static List<Application> getAppData(String o) {
        List<Application> list = new ArrayList<>();
        try {
            Log.e( tag: "YES", msg: "Y");
            JSONArray jsonObject = new JSONArray(o);
            for (int i = 0; i < jsonObject.length(); i++) {
                JSONObject jsonObject1 = (JSONObject) jsonObject.get(i);
                Application app = new Application();
                app.setAppId(Integer.parseInt(jsonObject1.getString( name: "AppID")));
                app.setAppName(jsonObject1.getString( name: "AppName"));
                list.add(app);
            }
        } catch (JSONException e) {
            Log.e( tag: "JSON EX", e.getLocalizableMessage());
        }
        return list;
    }
}
```

Figure 19: Loading App Names

When the user selects and clicks on the 'View' button next screen will appear with results. Figure 20 and 21 illustrate the code segments.

```

private void drawPieChart(float positivePercentage, float negativePercentage, int posi:
    pieChart.setUsePercentValues(true);
    pieChart.getDescription().setEnabled(true);
    pieChart.setExtraOffsets(5, 10, 5, 5);
    pieChart.setDragDecelerationFrictionCoef(0.9f);
    pieChart.setTransparentCircleRadius(61f);
    pieChart.setHoleColor(Color.WHITE);
    pieChart.getDescription().setText("Positive - "+positiveCount+"\n Negative - "+neg
    pieChart.getDescription().setEnabled(true);
    pieChart.animateY(1000);
    ArrayList<PieEntry> yValues = new ArrayList<>();
    yValues.add(new PieEntry(positivePercentage, "Positive"));
    yValues.add(new PieEntry(negativePercentage, "Negative"));

    PieDataSet dataSet = new PieDataSet(yValues, "Sentiment of application");
    dataSet.setSliceSpace(3f);
    dataSet.setSelectionShift(5f);
    dataSet.setColors(Color.parseColor( colorString: "#006400"), Color.parseColor( colorSti
    PieData pieData = new PieData((dataSet));
    pieData.setValueTextSize(10f);
    pieData.setValueTextColor(Color.YELLOW);
    pieChart.setData(pieData);
}

```

Figure 20: Codes for Graphing

```

private void setFeaturedGraphData(int featurd_addictivity, int featured_graphics, :

    //Add a list of bar entries
    ArrayList<BarEntry> entries = new ArrayList<>();
    entries.add(new BarEntry(0f, featurd_addictivity));
    entries.add(new BarEntry(1f, featured_graphics));
    entries.add(new BarEntry(2f, featured_memory));
    entries.add(new BarEntry(3f, featured_money));
    entries.add(new BarEntry(4f, featured_satisfaction));
    entries.add(new BarEntry(5f, featured_speed));

    //Note : These entries can be replaced by real-time data, say, from an API

    BarDataSet barDataSet = new BarDataSet(entries, "Rating");

    barDataSet.setColors(
        Color.parseColor( colorString: "#F78B5D"), Color.parseColor( colorString: "#
    BarData barData = new BarData(barDataSet);
    featuredBarChart.setData(barData);
    featured{...}

```

Figure 21: Codes for Bar Chart

6.6. Summary

In this chapter implementation of three modules was discussed. All the important parts with code segments and the results were shown in figures so that the reader can understand.

Next chapter is for the discussion of the evaluation of the system described in the above chapters.

Evaluation

7.1. Introduction

In this chapter evaluation of the system is discussed. Mainly since this is sentiment analysis it is required to evaluate the result of system generated result of sentiments. Furthermore, the final output is evaluated using 10 colleagues.

7.2. Evaluation of User Review Extractor

User Review Extractor was for extracting user reviews from Google Play Store. This application downloaded game names with reviews of 87 games automatically within 24 hours along with a total of 83337 reviews. Figure 22 shows the chart of a total number of reviews of the first 25 games of 87 and figure 23 shows the names of them. This tool functioned well for downloading user reviews from the Google Play Store without any interruption.

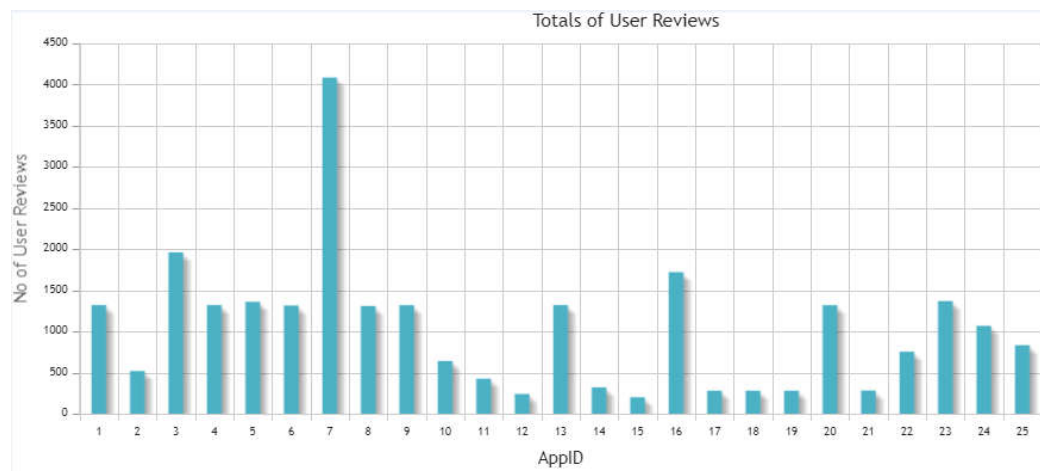


Figure 22 Totals of User Reviews

AppID	AppName
1	ABC_Kids___Tracing___Phonics
2	Air_Hockey_Challenge
3	Angry_Birds_Seasons
4	Archery_King
5	Asphalt_Street_Storm_Racing
6	Badminton_League
7	Barbie___Fashion_Closet
8	Basketball_Stars
9	Beach_Buggy_Racing
10	Bini_Super_ABC___Preschool_Learning_Games_for_Kids___
11	Block_Puzzle
12	Blossom_Blast_Saga
13	Bowling_King
14	Bubble_Fish
15	Bubble_Freedom
16	Bubble_Shooter
17	Bubble_Shooter_3
18	Bubble_Shooter_Genies
19	Bubble_Word_Games___Search___Connect_Word___Letters
20	Candy_Bomb
21	Candy_Crush_Friends_Saga
22	Candy_Crush_Jelly_Saga
23	Candy_Fever
24	Cars___Lightning_League
25	Coloring___Learn

Figure 23 Names of the first 25 games

7.3. Evaluation of Sentiment Analysis Module

As the main objective of this project, the total of positive and negative sentiments and feature-wise sentiment analysis for each application were calculated computationally. This combined model gives the very accurate and more informative picture about the application to the user than the existing models.

Table 1 shows the result of the sentiment analysis process for top 25 game applications according to the alphabetic order. It is clear that every application is analysed properly so that sentiment polarity is divided as negative and positive.

Table 1 Sentiment result of 25 games

Application Name	Total Comments	Negative	Positive

ABC_Kids__Tracing__Phonics	1320	54.394	45.606
Air_Hockey_Challenge	519	63.584	36.416
Angry_Birds_Seasons	1960	51.480	48.520
Archery_King	1320	60.076	39.924
Asphalt_Street_Storm_Racing	1360	65.000	35.000
Badminton_League	1315	58.327	41.673
Barbie_Fashion_Closet	4084	59.917	40.083
Basketball_Stars	1309	54.011	45.989
Beach_Buggy_Racing	1319	51.327	48.673
Bini_Super_ABC__Preschool_Learning_Games_for_Kids	641	53.198	46.802
Block_Puzzle	426	64.319	35.681
Blossom_Blast_Saga	240	55.000	45.000
Bowling_King	1320	63.864	36.136
Bubble_Fish	320	51.875	48.125
Bubble_Freedom	200	66.500	33.500
Bubble_Shooter	1720	73.779	26.221
Bubble_Shooter_3	280	62.857	37.143
Bubble_Shooter_Genies	280	50.357	49.643
Bubble_Word_Games__Search__Connect_Word__Letters	280	48.929	51.071
Candy_Bomb	1319	55.876	44.124
Candy_Crush_Friends_Saga	282	59.929	40.071
Candy_Crush_Jelly_Saga	754	41.114	58.886
Candy_Fever	1369	51.863	48.137
Cars__Lightning_League	1068	60.674	39.326
Coloring__Learn	833	58.944	41.056

Since the feature-wise semantic analysis was done, the positive/ negative ratio could be calculated for each application. Hence User can get a good picture of the game in which feature this game is good and in which feature the game is bad. Table 2 shows the feature-wise positive-negative ratio for 25 applications.

Table 2 Feature-wise positive-negative ratio

Application Name	Feature-wise positive/negative ratios					
	Speed	Additivity	Satisfaction	Money	Memory	Graphics
ABC_Kids__Tracing__Phonics	0 / 6	0 / 0	10 / 3	2 / 1	0 / 2	11 / 6
Air_Hockey_Challenge	0 / 0	8 / 2	3 / 0	1 / 0	3 / 4	6 / 6
Angry_Birds_Seasons	1 / 5	32 / 27	9 / 5	11 / 22	1 / 0	16 / 9
Archery_King	8 / 13	29 / 23	3 / 4	15 / 63	0 / 3	32 / 12
Asphalt_Street_Storm_Racing	6 / 20	6 / 6	3 / 0	8 / 30	1 / 6	30 / 57
Badminton_League	0 / 15	26 / 28	3 / 0	0 / 7	0 / 3	29 / 31

Barbie__Fashion_Closet	12 /14	0 /0	17 /12	17 /42	0 /0	31 /18
Basketball_Stars	1 /21	23 /29	1 /0	5 /34	0 /4	21 /11
Beach_Buggy_Racing	0 /7	18 /18	3 /3	3 /17	0 /4	78 /33
Bini_Super_ABC__Preschool_Learning_Games_for_Kids__	1 /0	0 /0	6 /2	2 /8	1 /0	3 /4
Block_Puzzle	0 /1	0 /6	2 /0	0 /0	0 /0	1 /0
Blossom_Blast_Saga	0 /0	9 /8	1 /0	0 /1	0 /1	18 /6
Bowling_King	5 /9	33 /45	8 /0	16 /41	1 /3	27 /21
Bubble_Fish	0 /0	3 /3	3 /1	0 /0	0 /0	2 /0
Bubble_Freedom	1 /1	6 /10	1 /0	3 /7	0 /0	5 /7
Bubble_Shooter	8 /28	44 /71	9 /9	3 /10	1 /3	39 /39
Bubble_Shooter_3	0 /3	3 /8	3 /0	0 /0	1 /1	4 /2
Bubble_Shooter_Genies	0 /2	21 /11	1 /0	4 /6	0 /0	7 /8
Bubble_Word_Games__Search__Connect_Word__Letters	1 /3	18 /15	1 /1	0 /6	1 /0	6 /3
Candy_Bomb	0 /2	7 /11	7 /1	0 /0	0 /1	8 /3
Candy_Crush_Friends_Saga	1 /8	14 /10	1 /2	2 /11	0 /0	12 /10
Candy_Crush_Jelly_Saga	3 /3	11 /15	3 /0	0 /3	1 /0	7 /3
Candy_Fever	1 /6	15 /23	6 /0	0 /0	0 /0	5 /5
Cars__Lightning_League	4 /12	5 /4	3 /0	0 /1	0 /2	15 /18
Coloring__Learn	0 /3	0 /0	3 /1	0 /1	0 /1	6 /11

To evaluate the result of sentiment analysis, the manual process was carried out.50 of reviews selected randomly and took into the table along with the result of the system. It could be taken from the MySQL database which contains all information.

Then all the reviews were manually checked for negative or positive. Furthermore, calculate the percentage of accuracy as below.

$$\text{Accuracy} = (\text{countif}(\text{system generated result} = \text{manual result}) / \text{No of reviews}) * 100$$

Table 2shows a sample of result taken to evaluate the sentiment analysis.

Table 3Sample ofthe evaluation result

User	Comment	System Generated Result	Manual Result
------	---------	-------------------------	---------------

SniperWolfMGS	Way too much behind a pay wall and constantly trying to sell you things for way too much money for such a simple straight forward concept. Feels like cash grab with little to offer. I'm also not 100% your opponents are real. I don't know how I can pick up a game and right away I can nail 20/20 points every round over 5 games and none of my opponents can do the same. Instead they are hitting 5s and 3s. It feels almost like they are just cup opponents trying to make you feel like you're good so you will spend money on a shallow game.	neg	neg
K Davis	A very fun and addictive game. I make bows all day but it's nice to be able to shoot from my couch at night. The influence of wind on the arrow is accurate when it comes to drift. A little costly on the upgrades for those that doesn't buy their way to the top but all games are that way.	pos	pos
Dustin Snider	First off the game is fun and addicting. Second I'm extremely unhappy with the games free cash program. I completed challenges to earn free cash that never shows up even after contacting their "help" line. So be prepared to	neg	neg

	jump through hoops and emails to be able to enjoy the game fully.		
--	---	--	--

According to the above method taken to evaluate the sentiment result, 88% accuracy was indicated.

7.4. Evaluation of the Final Output

Here evaluates the final output of the product that is user may like or dislike the game. To do this evaluation 10 colleagues were selected and facilitate to install and play one selected game. After that, their willingness was collected. Table 7.2 shows the result found.

Name of the Game:

Temple Run 2

Result of the Application:

Negative 55.6 %, Positive
44.4%

User	Like	Dislike
user1		1
uesr2	1	
user3		1
user4	1	
user5		1
user6		1
user7		1
user8	1	
user9		1
user10	1	

Dislike 60%

Like 40%

Table 7.2 Willingness of users

This proposed model contains overall sentiment analysis and feature-wise sentiment analysis. Hence user was more comfortable to predict the desired Android application than the classical sentiment analysis. Also it is clear that the accuracy of the final result was increased.

7.5. Evaluation of Data Visualization Module

A small Android application was implemented so that the user can select the desired application and see the result of analysis as graphical charts. This application was worked properly for every data in the database without any error. As well as the result displayed in few seconds after selecting the desired record. Hence it is clear that this module is working properly as expected. Appendix A.4 is an image of the final output of this application.

7.6. Summary

The application for the sentiment analysis of user reviews for the Android application facilitates to the user to see the result of the sentiments of user reviews in graphically. And also the features of the application that the users have been discussed can be seen as a chart. There for the user can get a good idea. The result of 88% accuracy guaranties the output of the project. Also, the result of the evaluation of the final output of the product got there is a slight difference between real data and the result.

In the next chapter focussed on conclusion and the future works to improve the system and the output of the system.

Discussion & Future Works

8.1. Introduction

This chapter contains the discussion of the overall findings and achievements of the project of Sentiment User Review Analyser for Android Applications. Furthermore, potential areas for future works are discussed. And also it is discussed how to improve this project.

8.2. Conclusion

A number of mobile users have significantly grown the last decade and in parallel, the mobile application market also has grown widely. There are various market places in the cloud and the google play store is the leading one in the market.

If consider the last few years, NLP is also developed fabulously in the information technology. NLP is being used fora lot of areas. Mainly in artificial intelligence, market analysis, etc. therefore there are many tools available in the market with high usability and high performance.

In this project, took a look, how NLP can be used to do sentiment analysis on user reviews of Android applications. Also,a successful system was implemented with an android application for obtaining the main aims and objectives mentioned in the beginning. It was realized that sentiment analysis can be successfully implemented so that the user is more comfortable to get an idea of a mobile application.

8.3. Future Works

In this project considering time constraints, only selected games were analyzed. As future development, it is possible to develop this system for more categories and more applications. Also, reviews were taken from google play store and then analyze offline. It is possible to check if there is a way to take all the reviews real time in a time efficient way. If so system can be developed to analyze data in real time.

Further, it is possible to find a pattern with taking user ratings, sentiment, other user's likes for the review, etc. so that the result is a very high level in accuracy.

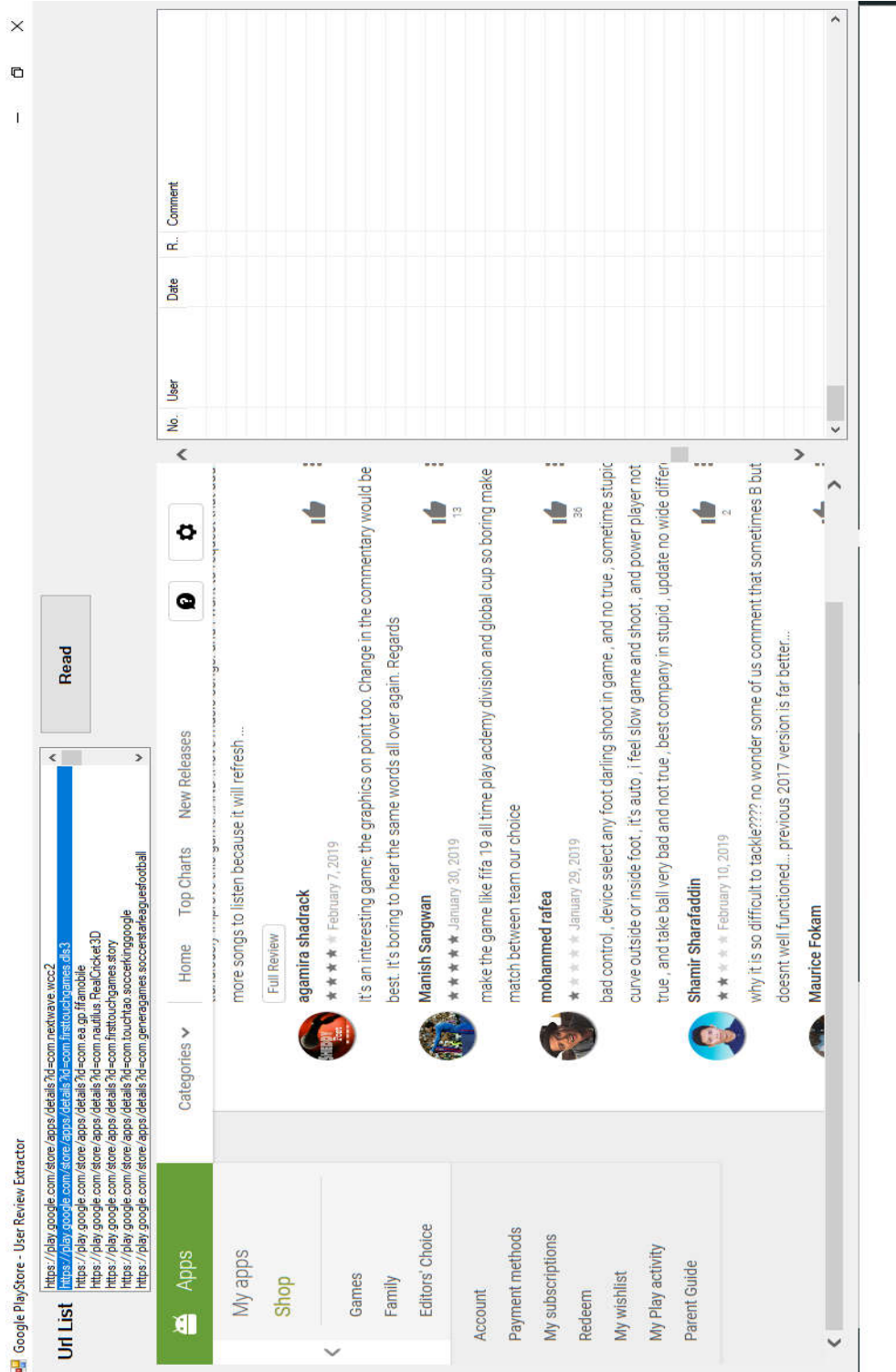
References

- [1] “• Mobile OS market share 2017 | Statista.” [Online]. Available: <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>. [Accessed: 07-Dec-2017].
- [2] “The history of Android OS: its name, origin and more | AndroidAuthority.” [Online]. Available: <https://www.androidauthority.com/history-android-os-name-789433/>. [Accessed: 08-Dec-2017].
- [3] www.linuxforu.com, “A Developers’s First Look At Android,” no. January, pp. 48–50, 2008.
- [4] N. Gandhewar and R. Sheikh, “Google Android: An Emerging Software Platform For Mobile Devices,” *Int. J. Comput. Sci. Eng.*, no. Special Issue, p. 6, 2010.
- [5] H. Hsueh, K. Huang, W. Wu, and C.-L. Li, “Evaluating the risk of Android application: Design and implementation of static analysis system,” in *Advanced Infocomm Technology (ICAIT), 2013 6th International Conference on*, 2013, pp. 236–237.
- [6] T. Matsudo, E. Kodama, J. Wang, and T. Takata, “A proposal of security advisory system at the time of the installation of applications on Android OS,” in *Proceedings of the 2012 15th International Conference on Network-Based Information Systems, NBIS 2012*, 2012, pp. 261–267.
- [7] C. Guo, J. Xu, L. Liu, and S. Xu, “Using association statistics to rank risk of Android application,” in *Proceedings of 2015 IEEE International Conference on Computer and Communications, ICC 2015*, 2016, no. SEPTEMBER, pp. 6–10.
- [8] T. E. Wei, C. H. Mao, A. B. Jeng, H. M. Lee, H. T. Wang, and D. J. Wu, “Android malware detection via a latent network behavior analysis,” in *Proc. of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012 - 11th IEEE Int. Conference on Ubiquitous Computing and Communications, IUCC-2012*, 2012, pp. 1251–1258.
- [9] H. H. Kim and M. J. Choi, “Linux kernel-based feature selection for Android malware detection,” *APNOMS 2014 - 16th Asia-Pacific Netw. Oper. Manag.*

Symp., 2014.

- [10] M. Vajdi, A. Torkaman, M. Bahrololum, M. H. Tadayon, and A. Salajegheh, “Proposed new features to improve Android malware detection,” *2016 8th Int. Symp. Telecommun. IST 2016*, pp. 100–104, 2016.
- [11] S. A. Licorish, A. Tahir, M. F. Bosu, and S. G. Macdonell, “On Satisfying the Android OS Community: User Feedback Still Central to Developers’ Portfolios,” 2015.
- [12] E. Ha and D. Wagner, “Do Android users write about electric sheep? Examining consumer reviews in Google Play,” in *2013 IEEE 10th Consumer Communications and Networking Conference, CCNC 2013*, 2013, pp. 149–157.
- [13] E. Haddi, X. Liu, and Y. Shi, “The Role of Text Pre-processing in Sentiment Analysis The Role of Text Pre-processing in Sentiment Analysis,” *Procedia Comput. Sci.*, vol. 17, no. December 2014, pp. 26–32, 2013.
- [14] H. Siqueira and F. Barros, “A Feature Extraction Process for Sentiment Analysis of Opinions on Services.”

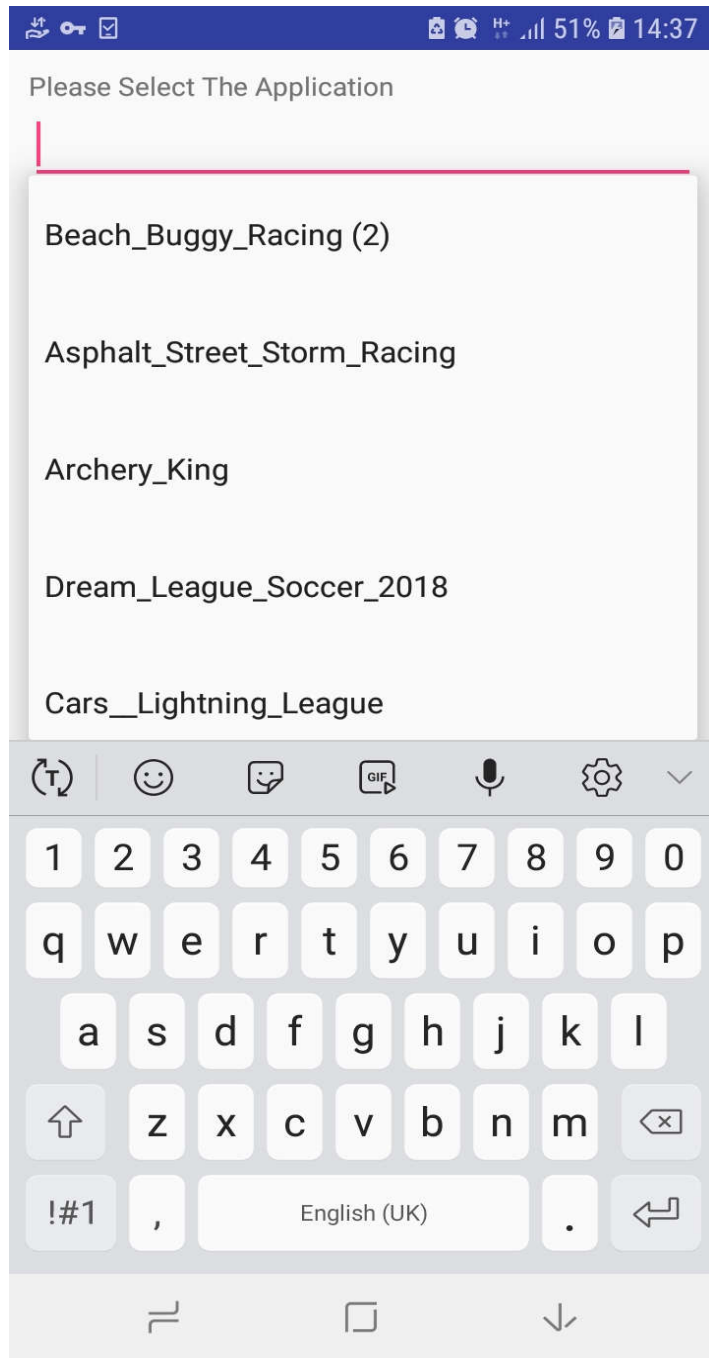
A.1. Image of the User Review Extractor



A.2. Image of Extracted data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	21/06/2018	Brandon Whitlock	2	The game is designed amazingly and is incredibly addictive. But it seems to me that both are paired with you once you start doing well to make you want to spend re															
2	04/07/2018	Tyler Livingston	1	Spent a few bucks in game and have got constant WAAAAAY out matched EVER since... Opponents have 150+ more trophies clubs I could only dream about and he															
3	29/06/2018	Eric Dees	2	This is the only game I have played where you are faulted for playing good and winning. If you move up your ranks fast because you're good at the game you will sta															
4	24/06/2018	Dawn Smithcamp	3	I enjoy this late wanting to relax and not hassle with crazy, frantic players as in some other games. here it searches and finds an opponent right away, its nice. BUT															
5	20/06/2018	Chaoxiege801x	4	Excellent game. It's fair and great emojis to communicate with. Please add more emojis. Also trying to invite Facebook friends but cannot since the app doesn't allow															
6	28/06/2018	Dhruwang Jariwala	5	It's damn addictive game. Once you start playing you just can't stop. Graphics are awesome. Controls are easy to use. Interface is also very friendly. Just one suggesti															
7	26/06/2018	Anakin Solo	4	I can't lie I didn't think this would be the best golf game. And while it isn't it is one of the best. It's perfect for mobile games are the best of one hole. The controls ar															
8	21/06/2018	A Google user	3	Games fun to play...very addictive...would love to see some special balls has prizes in tournaments. Also something needs to be done about club cards seems to get a															
9	19/06/2018	Pete Shelby	5	Golf Clash is the best golf app I've come across every period! It's totally addictive intuitive controls & UI. Graphics are awesome (I've got a Google Pixel) & it has a sm															
10	30/06/2018	A Parrot Buddy	1	Every time your opponent is extremely good or armed with better ammo...it's very hard to find that your opponent plays crazy shots but still ball goes the right dir															
11	08/07/2018	Scott Albanese	3	It's fun but here is why I gave it 3 stars. One you have to play against other people all the time. I prefer the option of playing a solo campaign. Two no big surpris															
12	02/07/2018	Dominic Kelly	1	Nothing but hate for this game! Wouldn't recommend it to anyone. Absolutely disgraceful support service. They cannot fix any issues because they either cant see t															
13	19/06/2018	Harry Foster	1	Edit: This game is still bull s**t. it's all algorithms that aren't consistent and no skill can give you any edge. No matter if you hit perfect shots with no wind and an acc															
14	03/07/2018	Nick Gangemi	3	Game is fun and addicting but why isn't matchmaking based off the players club levels? Also this game doesn't play like real golf in many scenarios. Biggest one peop															
15	19/06/2018	Ingvar GrAans	2	Bots are about. A player hit the narrow line between two bunkers after having aimed the previous shot on the fly the right amount left coming 2.67 m from the flag.															
16	01/07/2018	Ian McGuffin	3	Have just completed ladt tournament with -19 and played out of my skin to get that. some of the par 4 holes are really long and very hard to get a birdie. in my leagu															
17	20/06/2018	Andrew Faulkner	3	You guys need to make it where the person shooting 2nd can't see the power curl and ball spin we're using on our shots. It ticks me off because time and time again															
18	01/07/2018	John Andrews	5	Love the game all around but when you're just starting out and playing people with a lot of the clubs unlocked and good balls and a couple divisions above you that's															
19	24/06/2018	amyt sudarshan	4	There are issues with the game which needs to be fixed. The game would randomly close and forfeited without any reason and the shot at times would go - Hooked															
20	24/06/2018	Robert R	1	Terrible game imbalance. Wins and loss streaks swings are grossly exaggerated. Pretty obvious since the game only asks you to leave a rating after a win. Requires Wi															
21	04/07/2018	Nick C	2	Would be more fun if it didn't lag out all the time. Needs a single player mode for making money faster. it auto forfeits matches even when its not your turn super fr															
22	28/06/2018	Jon Coakley	1	Edited review: This game is terrible. No matter what connection I'm on internet wise it will freeze up for no reason. A lot of the time it will connect again but I just g															
23	18/06/2018	Jason Lockman	1	Game has been crashing a lot lately. In the middle of a game it crashes after my shot when I get back my shot is in a completely different area then where it landed. I															
24	30/06/2018	Michael Taling	2	Once you start it's great. But remember this is a games designed for you to buy and spend money. I have a suspicion that once you win too much you play against a															
25	06/07/2018	Iustin Coerswell	1	The customer service is terrible... Any Discrepancy you have is never fixed and it always vnur fault.. I've had matches the was a draw am they made it a win for the nfl															

A.3. First screen of mobile application



A.4. Result screen of mobile application

