**SENTIMENT ANALYSIS AND CATEGORIZATION OF MOBILE APPLICATION REVIEWS**

***Submitted by***

**ISHVARYA. K**

**RESHMA MANOJ**

**RESHMA SARA POTHEN**

***Guided by***

**DR. S. SWAMYNATHAN**

**DEPARTMENT OF INFORMATION SCIENCE AND**

**TECHNOLOGY**

**ANNA UNIVERSITY**

**IT 8811 - FINAL YEAR PROJECT**

**VIII Semester B.TECH IT H – BATCH and G – BATCH**

****

**INFORMATION SCIENCE AND TECHNOLOGY**

**CEG CAMPUS**

**ANNA UNIVERSITY**

**CHENNAI 600 025**

**MARCH, 2017**

**Table of Contents**

**CHAPTER 1**

1. **INTRODUCTION** 
   1. Online Applications
   2. Mobile Applications
   3. Mobile App Stores
      1. Android App stores
      2. Cross-platform App stores
      3. iPhone/iOS App stores
      4. BlackBerry App stores
      5. Manufacturer-based App stores
      6. Operator/Carrier App stores
   4. Categories of mobile applications
      1. Utility apps
      2. Entertainment apps
      3. Games apps
      4. News apps
      5. Productivity apps
      6. Lifestyle apps
      7. Social networking apps
   5. Feedback about mobile applications
      1. Importance of feedback
   6. Sentimental Analysis
      1. Salient features of sentimental analysis
      2. Importance of sentimental analysis

**CHAPTER 2**

1. **LITERATURE SURVEY** 
   1. Limitations of Existing System
   2. Objectives of Proposed System
   3. Data Collection
   4. Sentiment Analysis of Reviews
      1. Corpus Based Approach
      2. Classifier Based Approach
      3. VADER Based Approach
   5. Categorization of Reviews
   6. Visualization of Results

**CHAPTER 3**

1. **MOBILE APPLICATION SENTIENT ANALYZER ARCHITECTURE** 
   1. Flow Diagram
   2. Architecture for Mobile Application Sentiment Analyzer
   3. Module Description for Mobile Application Sentiment Analyzer
   4. Hardware and Software Requirements

**CHAPTER 4**

1. **IMPLEMENTATION DETAILS**
   1. Collection of Dataset
   2. Creation of Corpus
   3. Review Preprocessing
      1. Stopword Removal
      2. Lemmatization
   4. Review Classification using Sentiment Analyzer
   5. Categorization of Reviews
   6. Output Visualization
   7. Algorithm Used in Corpus Based Approach

**CHAPTER 5**

1. **RESULTS AND DISUCSSION**
   1. Results for the Corpus Based Approach
   2. Results for the Classifier Based Approach
   3. Results for the VADER Based Approach
   4. Results for Bug Report and Feedback Categorization

**CHAPTER 6**

1. **CONCLUSION AND FUTURE WORK**

**CHAPTER 1**

1. **INTRODUCTION**
   1. **Online Applications**

Online applications are programmes, such as paint or word that operate online on a website. Online applications may also be referred to by other names like[SaaS](http://centralhubdublin.com/softwaredevelopmentireland.php) or ‘[Software as a Service](http://centralhubdublin.com/softwaredevelopmentireland.php)‘. It simply means a programme – or something that will carry out some kind of a task.

In the early days of the internet, websites and web pages existed as static content. Pages were online to be read and viewed but little interaction occurred. Interaction online took place by email. Slowly, a need for on site interaction evolved and was developed in all kinds of ways from commenting to booking forms.

Since then the internet has come so far in terms of functionalities and interaction that it came to be referred to by some people as Web 2.0. Web 2.0 refers to the highly interactive, non-static web that we now use in our day to day lives.

**Browser Based**

In a browser-based Web application, JavaScript instructions are contained within the Web page that is retrieved from a website. Combined with the HTML code that determines the visual layout and the CSS style sheet, the HTML, JavaScript and CSS are executed via the browser. In addition, processing at the server side is often widely performed to access databases and other networks. The data for a Web application may be stored locally or on the Web, or in both locations. Common web applications include [webmail](https://en.wikipedia.org/wiki/Webmail), [online retail sales](https://en.wikipedia.org/wiki/Online_shopping), [online auctions](https://en.wikipedia.org/wiki/Online_auction), [wikis](https://en.wikipedia.org/wiki/Wiki), [instant messaging services](https://en.wikipedia.org/wiki/Instant_messaging) and many other functions.

**Client Based**

Web applications may also run without the browser. A client program, which is either installed in the user's computer or mobile device or is downloaded each session, interacts with a server on the Web using standard Web protocols. This is similar to the "client/server" architecture that prevailed in companies before the Internet exploded, except that today the server is often on the Internet rather than the local network. Just like browser-based applications, the data may be stored remotely or locally. Examples include ASP and SAAS.

**Mobile Web App**

Countless mobile apps use the Web for additional information. For example, the iOS and Android versions of this encyclopaedia install all the text locally in the device but retrieve all the images from a server via Web (HTTP) protocols.

* 1. **Mobile applications**

A mobile application, most commonly referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet computer. Mobile applications frequently serve to provide users with similar services to those accessed on PCs. Apps are generally small, individual software units with limited function.

There are several ways of targeting mobile devices when making a web application:

* [**Responsive web design**](https://en.wikipedia.org/wiki/Responsive_web_design) can be used to make a web application - whether a conventional web site or a single-page application viewable on small screens and work well with touchscreens.
* [**Native apps**](https://en.wikipedia.org/wiki/Native_app)or "mobile apps" run directly on a mobile device, just as a conventional software application runs directly on a desktop computer, without a web browser (and potentially without the need for Internet connectivity); these are typically written in Java (for Android devices) or [Objective C](https://en.wikipedia.org/wiki/Objective_C) or [Swift](https://en.wikipedia.org/wiki/Swift_(programming_language)) (for iOS devices). Recently, frameworks like [React Native](https://en.wikipedia.org/wiki/React_Native), [Flutter](https://en.wikipedia.org/wiki/Flutter_(company)) and Xamarin allow the development of native apps for all platforms using languages other than each standard native language.
* **Hybrid apps** embed a mobile web site inside a native app, possibly using a hybrid framework like Apache Cordova among others. This allows development using web technologies (and possibly directly copying code from an existing mobile web site) while also retaining certain advantages of native apps (e.g. direct access to device hardware, offline operation, [app store](https://en.wikipedia.org/wiki/App_store) visibility).
  1. **Mobile App Stores**

Mobile applications can be downloaded from the following mobile application stores. Some of these applications can be free of cost and some of them are paid applications.

* + 1. **Android App Stores**

[**Appia**](http://landing.appia.com/mobyaffiliates) – runs an app store service The Appia Network alongside it is cross promotion platform.

[**Codengo**](http://www.mobyaffiliates.com/mobile-app-marketing/codengo/) – submit to over 20 app stores from a single form fill

[**AppBrain**](http://www.appbrain.com/) – Website that helps users discover Android apps across the official Google Play store and other indie stores.

[**SlideMe**](http://slideme.org/)– Provides ‘on device’ application store for device manufactures, as well as a web-based platform. Also offers app store solutions to niche markets.

[**Google Play**](https://play.google.com/store) – Google’s official Android app store, which features over 500,000 apps, as well as music and video streaming services.

* + 1. **Cross-platform App Stores**

[**OpenAppMkt**](http://www.openappmkt.com/app/4c3e754dea754f1de1000012/YouTube) – Android and iPhone-compatible market that distributes web-apps built in HTML5. Unlike the traditional 70/30 split, developers keep 80% of app sale revenue.

[**GetJar**](http://www.getjar.com/) – One of the biggest 3rd party app stores currently in operations. Claims to distribute more than 350 thousand mobile apps, with 395 registered developers. Also offers a pay per download system to help developers get visibility.

**Handmark** – App developer behind popular Tweet cast app, which also runs an app store for Android, BlackBerry, Windows and Palm devices.

[**Handster**](http://www.handster.com/)– Claims to features over 30,000 apps from 4000 vendors across Android, Symbian, Java and BlackBerry platforms. Also offers white label platform to manufacturers, distributors and carriers. Owned by Opera.

[**WAC**](http://www.wacapps.net/home) – The app store for the Wireless Application Community (a consortium backed by a number of telcos). It focuses on distributing cross-platform web-apps using mainly HTML5.

* + 1. **iPhone/iOS App Stores**

[**App Store**](http://www.apple.com/iphone/from-the-app-store/) – Apple’s official iOS application store and is still thought to be the biggest app store available across all smartphone platforms, though Android is catching-up fast.

[**Cydia**](http://cydia.saurik.com/) – Unofficial iOS app store that is only available to users with ‘jailbroken’ iOS devices. Most apps are offered for free, but there are also paid apps. Claims over 4.5 million weekly users.

[**Lima**](http://infini-dev.com/blog/?/home)– Browser-based application installer for jailbroken iPhones. Works similar to Cydia, allowing you to download apps not available in the official App Store, but within the Safari browser.

[**PremierAppShop**](http://premierappshop.com/) – Legal iPhone application store that delivers downloadable apps, which can be used offline via a browser-based shopfront.

**1.3.4. BlackBerry OS App Stores**

[**BlackBerry App World**](http://appworld.blackberry.com/) – RIM’s official BlackBerry app store, which features more than 40,000 apps.

[**CrackBerry Store**](http://software.crackberry.com/homeSoftware.asp) – Independent BlackBerry-specific app store run by the online blog CrackBerry. Takes advantage of the large CrackBerry community to promote apps. Powered by MobiHand.

**BBNation**– BlackBerry-focused content portal with apps, themes, ringtone and games. Owned by Motek Americas.

**1.3.5. Manufacturer-specific App Stores**

[**Samsung Apps**](http://www.samsungapps.com/) – The official Samsung App store for its  Bada operating system, made popular on devices such as the Samsung Wave. Bada is expected to be phased out soon, so Samsung Apps’ future is unclear.

[**LG Smart World**](http://us.lgworld.com/web.main.dev) – The official app store run by manufacturer LG. Offers Android apps to US users and Windows Phone apps to users in Asia.

**Motorola Shop4Apps** – Motorola’s own-branded Android app store aimed exclusively at the Chinese market.

[**Dell Mobile App Store**](http://dellmobileappstore.com/DellLanding.jsp?siteId=658) – The official app store of Dell. Run by PocketGear and contains over 40,000 apps, across Android, BlackBerry and Symbian.

**1.3.6**. **Operator/Carrier App Stores**

[**Verizon App Store**](http://developer.verizon.com/content/vdc/en/verizon-platforms.html) – App store run by US carrier Verizon. Works across Android, BlackBerry and Brew platforms.  Partners with UK operator Vodafone’s app store.

[**Vodafone AppSelect**](https://developer.vodafone.com/) – Official app store for UK operator Vodafone. Focuses on Android for European users. BlackBerry/Symbian are supported for African users. Offers developers free testing.

[**China Mobile**](http://dev.10086.cn/en/supesite/?action-singlepage-name-aboutus) – The app store of China’s biggest mobile operator. Offers apps across Android and Symbian platforms. Claims a user base of over 500 million.

[**T-Mobile Mall**](http://developer.t-mobile.com/site/global/home/p_home.jsp) – Official app portal of UK carrier T-Mobile. T-Mobile Mall doesn’t host applications, but rather acts a portal that highlight different apps on BlackBerry AppWorld, Google Play and Windows Marketplace.

In our project we will be focussing on exclusively on Google’s PlayStore.

* 1. **Categories of Mobile Applications**

The various categories of mobile apps are:

1. Utilities
2. Entertainment
3. Games
4. News
5. Productivity
6. Lifestyle
7. Social Networking

Each of these app types are quite different and require a different style of programming from the other.

* + 1. **Utilities apps**

These are handy tools that you carry with you that help you perform simple tasks.

Examples of utility apps are:

* Calculators
* Note-pads
* Communication apps
* Weather apps

Utility apps are typically used frequently (4-5 times a week) but have shorter session times.

This is because the user doesn’t want to spend time playing around in the app – they came to solve a problem quickly so they are in an out in a matter of seconds.

**1.4.2. Entertainment apps**

Examples of entertainment apps are:

* Face Juggler
* [Ice Effex](http://www.buzinga.com.au/tech-trends/confronting-photo-app-tackles-the-australian-ice-epidemic-press-release/)
* Duolingo
* DubSmash

Entertainment apps are distinguished from games apps because they often have quite different goals, even though both of them seek to entertain the user.

Entertainment apps are often educational, and seek to ‘gamify’ an already existing process or activity.

* + 1. **Games apps**

Game apps have been a favourite among app developers for years. Examples of games apps are:

* Angry Birds
* Sudoku
* Trivia Crack

Games apps are still a hugely popular subset of app development today.

* + 1. **News apps**

Having a news app has become a necessity for media and publishing companies to stay relevant in the ‘*digital age of journalism*‘.

The most important feature of a news app are the sharing functions.

These apps [spread virally](http://www.buzinga.com.au/buzz/how-to-build-a-viral-app/) social media and spread fresh, relevant content to a mass audience.

Examples of news apps are:

* The New York Times
* Buzzfeed
* The Hindu
* Flipboard

Essentially, the goal of any publishing app is to encourage greater content consumption, so that the publisher can generate more revenue from advertisers.

* + 1. **Productivity apps**

Productivity apps are developed to help us be more productive.

Examples of productivity apps are:

* Finance apps
* Calendars
* Translators
* Grocery list makers

 A [seamless user experience](http://www.buzinga.com.au/buzz/mobile-user-experience-design/) is crucial for productivity apps.

* + 1. **Lifestyle apps**

Lifestyle apps are used to solve everyday problems we have in our lives.

We use these apps to enhance our lifestyles and to make everyday living ‘easier’.

Examples of lifestyle apps are:

* Fitness apps
* Travel Apps
* Food & Drink apps

Lifestyle apps tend to complement other products and services very well. It’s for this reason that sponsored advertising partnerships are incredibly successful in lifestyle apps.

For example, RunKeeper offers its exercise fanatics exclusive deals on Nike products. They then take a cut of the sales attributed to this ad.

* + 1. **Social networking apps**

Nearly all of us use social networking apps.

Social media is so pervasive in our society that we are now seeing apps that wouldn’t traditionally be considered a social network add social as a core part of their product.

For example, FitBit is a fitness app with the added in-app functionality of following your friends and engaging in competitions with and against them.

Examples of social networking apps are:

* Facebook
* Circles
* Path
* Instagram

Whatever the type of app is, as a mobile app developer, you need to understand that the purpose of an app is to make people’s lives easier.

* 1. **Feedback about Mobile Applications**

Creating a mobile app takes a lot of hard work and success is never guaranteed. Even with ample time and money invested, app developers can still fail to build a successful mobile app. Commonly, app developers are never able to fully understand what their potential app customers want in a mobile app.

Instead of creating a mobile app you think people will want to use, you need to build one you know people will want to use. The idea doesn’t have to be perfect from the start, but you need to be open to improvement and making fixes to your app.

So how do you know what to include in your mobile app, what needs to be fixed, or what your customers want? The answer is feedback. Getting feedback from your customers will help you build a successful mobile app and is an important part of [marketing your mobile app](http://www.apptamin.com/?p=2840).

* + 1. **Importance of Feedback**

Knowing exactly what your customers will want in a mobile app is challenging. No matter what you think your customers want the only way you can ever know for sure is by listening to them. Only your customers can tell you what could be better or what doesn’t work. Companies that don’t listen to their customers often make uneducated guesses as to what their customers need and want.

Feedback lets your customers help drive your product roadmap to make the changes that matter most to your customers with confidence and eliminate time figuring out what features you think are most important. Mobile moves fast so limiting the number of guesses you make throughout development can make a key difference.

You can spend as much time looking at your analytics to [understand your customer behaviour](http://www.apptamin.com/?p=1701), and although the analytics part is important nothing can provide more proof of what your customers want than what they actually say. The more customer feedback you receive the better you can understand your customer base and how your app is being used.

Asking for and responding to feedback is also a great way to foster relationships between you and your mobile customers and develop a community. When customers feel that their feedback is taken seriously they are more willing to give it and feel more connected to the app.

* 1. **Sentiment Analysis**
     1. **Salient Features of Sentiment Analysis**

Sentiment: A sentiment is an attitude, thought, or judgment prompted by feeling.

Sentiment analysis: It is also known as opinion mining. It studies people’s sentiments towards certain entities. Sentiment analysis can be done on just about any text and it helps reveal the writer’s opinion on the subject matter. Sentiment analysis is widely applied to the voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.

* + 1. **Importance of Sentimental Analysis**

Sentiment analysis is critical because helps you see what customers like and dislike about you and your product. Sentiment analysis can let you know if there has been a change in public opinion toward any aspect of your business. Peaks or valleys in sentiment scores give you a place to start if you want to make product improvements, train sales or customer care agents, or create new marketing campaigns. Sentiment analysis is not a once and done effort. By reviewing your customer’s feedback on your business regularly you can be more proactive regarding the changing dynamics in the market place.

Sentiment analysis can even be applied on mobile application reviews. This helps the developers to find out what the customers feel about the application and what features need to be modified, newly incorporated etc.

**CHAPTER 2**

1. **LITERATURE SURVEY**

One fundamental problem in sentiment analysis is categorization of sentiment polarity. Given a piece of written text, the problem is to categorize the text into one specific sentiment polarity, positive or negative (or neutral). Based on the scope of the text, there are three levels of sentiment polarity categorization, namely the document level, the sentence level, and the entity and aspect level. The document level concerns whether a document, as a whole, expresses negative or positive sentiment, while the sentence level deals with each sentence’s sentiment categorization; The entity and aspect level then targets on what exactly people like or dislike from their opinions.

1. Springer, Journal of Big Data 2015, Sentiment Analysis using product reviews.

The paper details the following:

* Data collection and Part of speech tagging.
* Sentiment Phrase Identification and score computation.
* Categorization and result interpretation.

1. Adjective-Verb-Adverb Combinations for Sentiment Analysis, IEEE Intelligent Systems (Volume: 23, Issue: 4 , July-Aug. 2008).

This approach can identify the intensity of opinion on any topic using a comprehensive framework for analyzing the sentiments expressed by combinations of adjectives, verbs, and adverbs. For sentiment classification, a document is searched for sentiment bearing words like adjectives. By means of SentiWordNet, scores for positivity and negativity are determined for these words. An interpretation of the scores then leads to the document polarity.

1. Kim S-M, Hovy E (2004) Determining the sentiment of opinions. In: Proceedings of the 20th international conference on Computational Linguistics, page 1367. Association for Computational Linguistics, Stroudsburg, PA, USA
2. Liu B (2010) Sentiment analysis and subjectivity. In: Handbook of Natural Language Processing, Second Edition, Taylor and Francis Group, Boca
3. Pang B, Lee L (2008) Opinion mining and sentiment analysis.

1. N. Bettenburg, S. Just, A. Schröter, C. Weiss, R. Premraj, and T. Zimmermann.
2. What makes a good bug report? In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, page 308. ACM Press, Nov. 2008.
3. Gartner. Number of mobile app downloads worldwide from 2009 to 2017 (in millions). Technical report, Gartner Inc., March 2015.

# Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification, Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference.

# [http://download.cnet.com/WhatsApp-Messenger/3000- 12941\_4-10972991.html](http://download.cnet.com/WhatsApp-Messenger/3000-%2012941_4-10972991.html). Reviews are taken from mobile app stores like Google play stores and iOS app stores for WhatsApp and other mobile apps to explore its features and defects of mobile applications and user experiences of the application.

* 1. **Limitations of existing system**

The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. For example, instead of sharing topic-related opinions, online spammers post spam on forums, making it difficult to identify any useful content in a review.

The second flaw is that ground truth of such online data is not always available. A ground truth is more like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral.

These problems are addressed in our application.

* 1. **Objectives of the proposed system**

As a first step towards a tool support for analyzing app reviews, this system performs sentiment analysis on them and then tries to automatically classify them according to the type of information they include. Sentiment analysis is performed on the reviews using 3 methods descried below. Following this, classification takes place. Broadly speaking, a review can be categorised into 2 basic types. Bug reports describe problems with the app which should be corrected, such as a crash, an erroneous behaviour, or a performance issue. In feature requests, users ask for missing functionality (e.g., provided by other apps) or missing content (e.g., in catalogues and games), share ideas on how to improve the app in future releases by adding or changing features. The complete procedure is detailed below.

* 1. **Data Collection**

Here data is collected for various applications from mobile app stores, primarily Google Play Store. For example from various domains like games, entertainment, utility and social media applications etc. The apps are selected for their overall popularity, diverse users and domain.

The below table shows the details of the reviews collected.

|  |  |  |
| --- | --- | --- |
| APPLICATION NAME | CATEGORY | NUMBER OF REVIEWS |
| Candy Camera | Photo Editing | 635 |
| Candy Crush | Games | 559 |
| Cricbuzz | Sports | 500 |
| Gaana Music | Music | 520 |
| Instagram | Social Media | 1023 |
| NDTV | News | 516 |
| Saavn Music | Music | 501 |
| Snapchat | Social Media | 998 |
| Sportskeeda Official | Sports | 514 |
| Subway Surfers | Games | 798 |
| Temple Run | Games | 909 |
| Twitter | Social Media | 1129 |
| The Hindu | News | 500 |
| Whatsapp | Social Media | 823 |
| YouCam Perfect | Photo Editing | 535 |

* 1. **Creation of Corpora**

Various corpora are available online. The corpus used in our approach is the AFINN word list. AFINN is a list of English words rated for valence with an integer between minus five (negative) and plus five (positive). The words have been manually labeled by Finn Årup Nielsen. It contains 2477 words and phrases. We modified this word list to account for abbreviations and emoticons which are frequently used in reviews online.

Apart from this, separate corpora have been created comprising of words that indicate either a Feature Request/Suggestion or a Bug Report. These two are the two categories that the reviews will be placed in once sentiment analysis is complete.

* 1. **Sentiment Analysis of Reviews**

Sentiment analysis is performed on the reviews. This classifies the reviews in positive, negative or neutral based on the content of the review. 3 methods have been explored in this system. They are detailed below.

* + 1. **Corpus Based Approach**

The computational speed and efficiency of dictionary-based approaches to sentiment analysis, together with their intuitive appeal, make such approaches an attractive alternative for extracting emotional context from text. At the same time, both types of dictionary or corpus based approaches offer potential limitations as well. Pre-constructed dictionaries for use with modern standard U.S. English have the advantage of being exceptionally easy to use and extensively validated, making them strong contenders for applications where the emotional content of the language under study is expressed in conventional ways. At the same time, the validity of such dictionaries rests critically on such conventional usage of emotional words and phrases. Conversely, custom dictionaries developed for specific contexts are sensitive to variations in word usage, but come with a high cost of creation and limited future applicability.

What we term specialized vocabularies arise in situations when the standard emotional valences associated with particular words are no longer correct, either because words that typically convey emotional content do not do so in the context in question or vice versa. For example, in colloquial English the word “love” almost always carries a positive valence (and its inclusion in pre-constructed sentiment dictionaries reflects this fact) while the word “bagel” does not. For professional and amateur tennis players, however, the two words might mean something very different; “love” means no points scored (a situation 3 which has, if anything, a negative valence) and the word “bagel” refers specifically to the (negative) event of losing a set 6-0 (e.g., “putting up a bagel in the first set”). It is easy to see how the application of a standard sentiment dictionary to a body of text generated from a discussion of tennis could easily lead to inaccurate inferences about its content. In such circumstances, an ideal approach is to develop an approach that considers more than a single word in a statement before categorising it as positive or negative. This is a challenge that must be iterated through coding.

This approach relies on a corpus of emotive words. The system compares the review text with the corpus and scores the review accordingly. Thus, the following takes place. First, the candidate terms are extracted from the reviews. Second, the terms are matched against our specialized corpus. Third, the sentiment score is calculated based on number and type of words.

* + 1. **Classifier Based Approach**

Natural Language Processing (NLP) is a vast area of Computer Science that is concerned with the interaction between Computers and Human Language.   
Within NLP many tasks are – or can be reformulated as – classification tasks. In classification tasks we are trying to produce a classification function which can give the correlation between a certain ‘feature’ and a class. This Classifier first has to be trained with a training dataset, and then it can be used to actually classify documents. Training means that we have to determine its model parameters. If the set of training examples is chosen correctly, the Classifier should predict the class probabilities of the actual documents with a similar accuracy. A dataset is preprocessed and from that preprocessed data, a training and a testing set is created and therefore they are classified.

Here the purpose is to determine the subjective value of a review, i.e. how positive or negative is the content of a review. This is done with the subtleties of human language; sarcasm, irony, context interpretation, and the different ways in which opinion can be expressed (subjective vs comparative, explicit vs implicit).

Naïve Bayesian algorithm is used to classify the reviews. Naive Bayes seems to be an appropriate classification algorithm as it can achieve high accuracy with a small training set and less than half of the time needed by the other studied classifiers (about 10 minutes on a machine with 2 GHz Intel Core i7 and 8 GB RAM).

* + 1. **VADER Based Approach**

VADER (for Valence Aware Dictionary for sEntiment Reasoning was created using \a combination of qualitative and quantitative methods to produce, and then empirically validate, a *gold-standard* sentiment lexicon that is especially attuned to microblog - like contexts. These lexical features were chosen with consideration for five generalizable rules that embody grammatical and syntactical conventions that humans use when expressing or emphasizing sentiment intensity. Incorporating these heuristics improves the accuracy of the sentiment analysis engine across several domain contexts (social media text, NY Times editorials, movie reviews, and product reviews). Interestingly, the VADER lexicon performs exceptionally well in the social media domain. The correlation coefficient shows that VADER performs as well as individual human raters at matching ground truth (aggregated group mean from 20 human raters for sentiment intensity of each tweet). Surprisingly, when further inspected for classification accuracy, it is seen that VADER even outperforms individual human raters at correctly classifying the sentiment of tweets into positive, neutral, or negative classes.

VADER retains (and even improves on) the benefits of traditional sentiment lexicons like LIWC: it is bigger, yet just as simply inspected, understood, quickly applied (without a need for extensive learning/training) and easily extended. Like LIWC (but unlike some other lexicons or machine learning models), the VADER sentiment lexicon is *gold-standard* quality and has been validated by humans. VADER distinguishes itself from LIWC in that it is more sensitive to sentiment expressions in social media contexts while also generalizing more favorably to other domains. VADER is freely available for download and use.

This library returns the positive, negative and neutral score per review. In addition to that, it also gives a compound score, which is a signed value indicating the degree of positivity or negativity in a review.

* 1. **Categorization of Results**

Once sentiment analysis has been performed, the next phase of the project begins. Each review is now compared to one of the two specialized corpora to categorize them as either

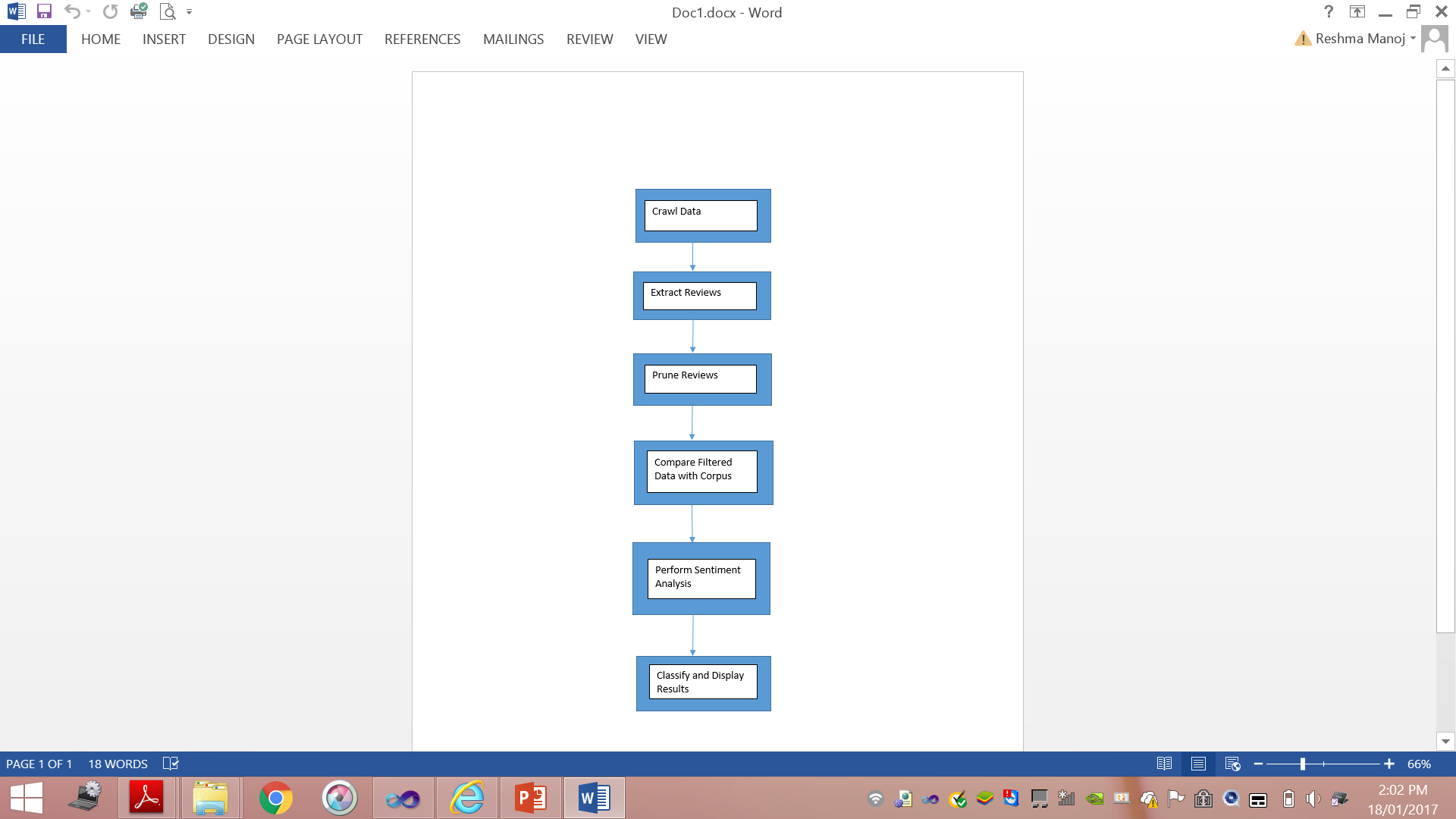
* + Feature Request or Suggestion
  + Bug Report
  1. **Visualization of Results**

The final output of the system visualizes the results in the form of pie charts. For each application, the system will present 2 pie charts.

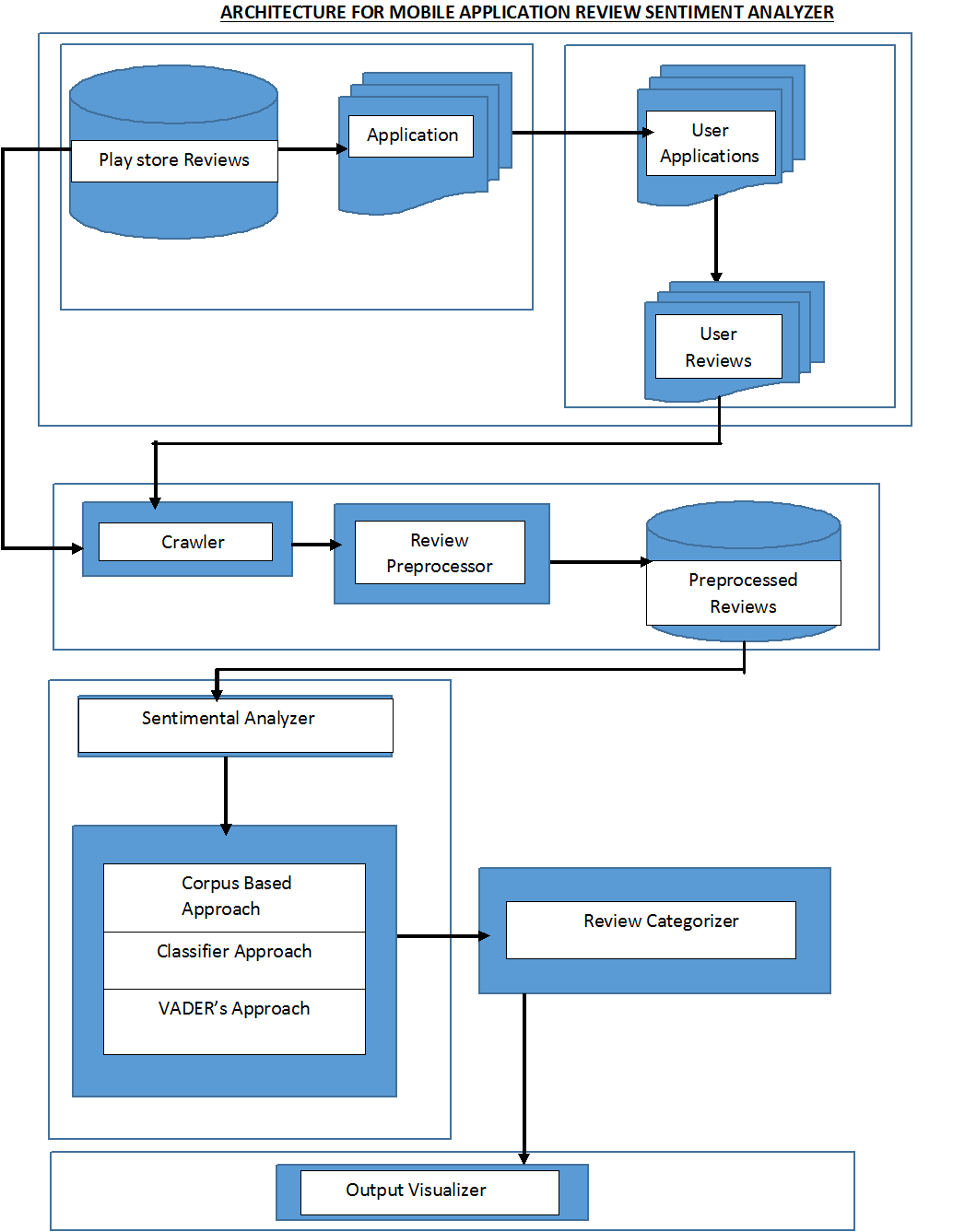
One will show the sentiment for that app. It will display the overall positive, negative and neutral sentiments based on total count of reviews. The other will display the bug reports and feature requests for the app. It will show the number of bug reports and the number of feature requests per app.

**CHAPTER 3**

1. **ARCHITECTURE FOR MOBILE APPLICATION SENTIMENT ANALYZER**
   1. **Flow Diagram**



* 1. **Architecture for Mobile Application Sentiment Analyzer**



* 1. **Modules**
     1. **Crawler**

Initially the crawler crawls and collects reviews from Google Play Store. Meta data is removed and each of the review is extracted separately.

* + 1. **Review Preprocessor**
       1. **Pruning the text**

The preprocessor module removes stop words like would, shall, do, will and is among others, essentially pruning the text.

* + - 1. **Lemmatization**

Here, the words in the text are converted into their root forms.

* + 1. **Sentimental Analyzer**

Then the preprocessed reviews are sent to the sentimental analyzer module. The analyzer was constructed to reflect the following 3 approaches:

1. **Corpus Based Approach**

In this approach, the reviews are compared to a specialized corpus which contains emotive words. These words have scores associated with them which are then tallied and presented as an overall score for each review.

1. **Classifier Based Approach**

In this approach, the reviews are used to train the Naïve Bayes Classifier which then further classifies the testing data into two categories of sentiment, positive and negative.

1. **VADER Based Approach**

In this approach, the reviews are run through VADER, or VADER (Valence Aware Dictionary and Sentiment Reasoner). VADER is a lexicon and rule-based sentiment analysis library. Based off the Natural Language Processing Toolkit in Python, this library returns the positive, negative and neutral score per review. In addition to that, it also gives a compound score, which is a signed value indicating the degree of positivity or negativity in a review.

* + 1. **Review Categorization**

After calculating the sentimental polarity of each review as a whole, the overall positivity, negativity and objectivity is displayed.Now, each review is compared to the pre-defined corpus to identify whether its text contains a Feature Request/Suggestion or a Bug Report.

* + 1. **Output Visualization**

Finally, the overall output is visualized in the form of pie charts. For each application, the system will present 2 pie charts.

One will show the sentiment for that app. It will display the overall positive, negative and neutral sentiments based on total count of reviews. The other will display the bug reports and feature requests for the app. It will show the number of bug reports and the number of feature requests per app.

**3.4. Hardware and Software Requirements**

**HARDWARE REQUIREMENTS**

Processor : Pentium dual core or above

RAM : 1 GB of RAM

Hard Disk Drive : 2 GB of hard disk space

**SOFTWARE REQUIREMENTS**

Front End/GUI : Microsoft Visual studio 2010

Operating System: Windows 7, Windows 8

Language : C#.NET 4.0 (ASP.net), Python

Back End : SQL Server2008

# CHAPTER 4

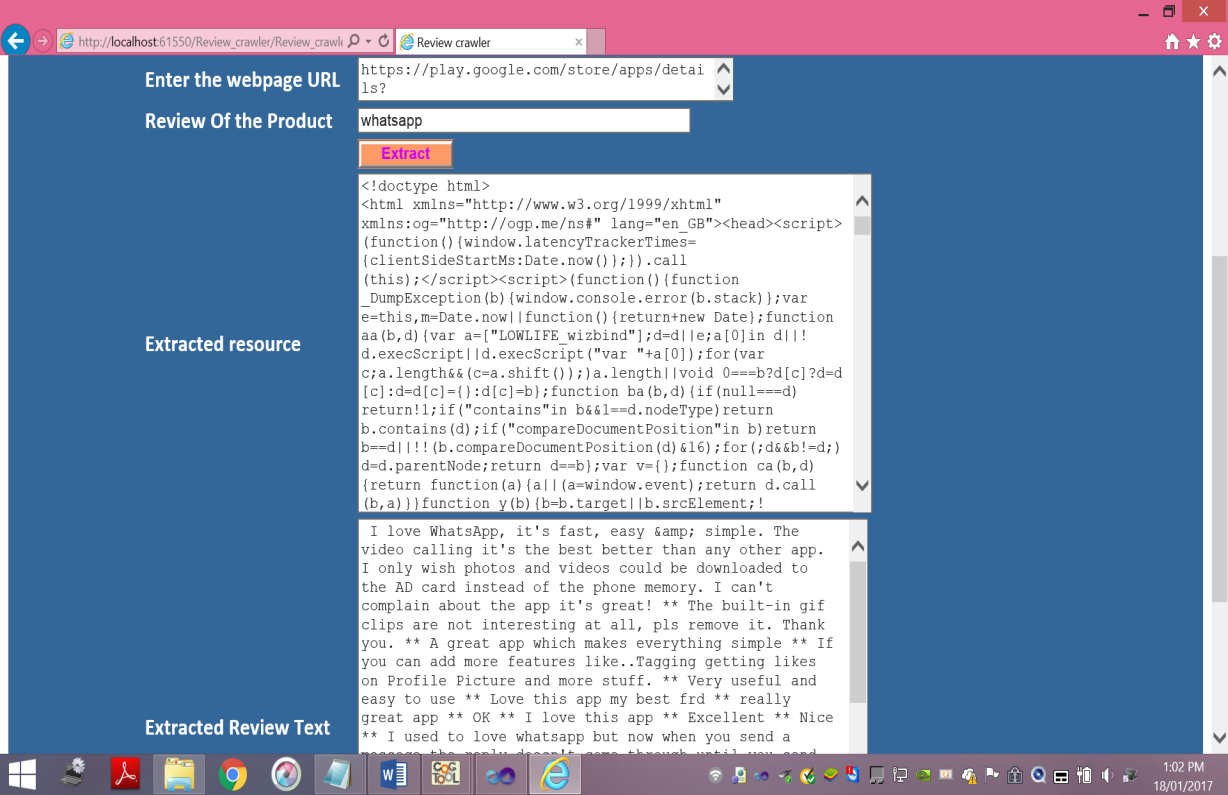
# IMPLEMENTATION DETAILS

* 1. **Collection of Data**

Initially reviews are collected from app stores (Google Play store, App store). Meta data is removed and each of the review is extracted separately. For example from various domains like games, entertainment, utility and social media applications etc. The apps are selected for their overall popularity, diverse users and domain.

The below table shows the details of the reviews collected.

|  |  |  |
| --- | --- | --- |
| APPLICATION NAME | CATEGORY | NUMBER OF REVIEWS |
| Candy Camera | Photo Editing | 635 |
| Candy Crush | Games | 559 |
| Cricbuzz | Sports | 500 |
| Gaana Music | Music | 520 |
| Instagram | Social Media | 1023 |
| NDTV | News | 516 |
| Saavn Music | Music | 501 |
| Snapchat | Social Media | 998 |
| Sportskeeda Official | Sports | 514 |
| Subway Surfers | Games | 798 |
| Temple Run | Games | 909 |
| Twitter | Social Media | 1129 |
| The Hindu | News | 500 |
| Whatsapp | Social Media | 823 |
| YouCam Perfect | Photo Editing | 535 |



* 1. **Creation of Corpus**

Various corpora are available online. The corpus used in our approach is the AFINN word list. AFINN is a list of English words rated for valence with an integer between minus five (negative) and plus five (positive). The words have been manually labeled by Finn Årup Nielsen. It contains 2477 words and phrases. We modified this word list to account for abbreviations and emoticons which are frequently used in reviews online.

Apart from this, separate corpora have been created comprising of words that indicate either a Feature Request/Suggestion or a Bug Report. These two are the two categories that the reviews will be placed in once sentiment analysis is complete.

* 1. **Review preprocessing**
     1. **Pruning the text**

Pruning is the process of removing the stop words like would, shall, do, will and is among others.

* + 1. **Lemmatization**

Here, the words in the text are converted into their root forms.

* 1. **Review Classification Using Sentimental Analyzer**

Then the preprocessed reviews are sent to the sentimental analyzer through the review pre-processor and the reviews are classified using the following 3 approaches:

* + 1. **Corpus Based Approach**

This approach relies on a corpus of emotive words. The system compares the review text with the corpus and scores the review accordingly. Thus, the following takes place. First, the candidate terms are extracted from the reviews. Second, the terms are matched against our specialized corpus. Third, the sentiment score is calculated based on number and type of words.

* + 1. **Classifier Based Approach**

Natural Language Processing (NLP) is a vast area of Computer Science that is concerned with the interaction between Computers and Human Language.   
Within NLP many tasks are – or can be reformulated as – classification tasks. In classification tasks we are trying to produce a classification function which can give the correlation between a certain ‘feature’ and a class. This Classifier first has to be trained with a training dataset, and then it can be used to actually classify documents. Training means that we have to determine its model parameters. If the set of training examples is chosen correctly, the Classifier should predict the class probabilities of the actual documents with a similar accuracy. A dataset is preprocessed and from that preprocessed data, a training and a testing set is created and therefore they are classified.

Here the purpose is to determine the subjective value of a review, i.e. how positive or negative is the content of a review. This is done with the subtleties of human language; sarcasm, irony, context interpretation, and the different ways in which opinion can be expressed (subjective vs comparative, explicit vs implicit). Naïve Bayesian algorithm is used to classify the reviews.

* + 1. **VADER Based Approach**

VADER (Valence Aware Dictionary and Sentiment Reasoner). VADER is a lexicon and rule-based sentiment analysis library. Based off the Natural Language Processing Toolkit in Python, this library returns the positive, negative and neutral score per review. In addition to that, it also gives a compound score, which is a signed value indicating the degree of positivity or negativity in a review.

* 1. **Review Categorization**

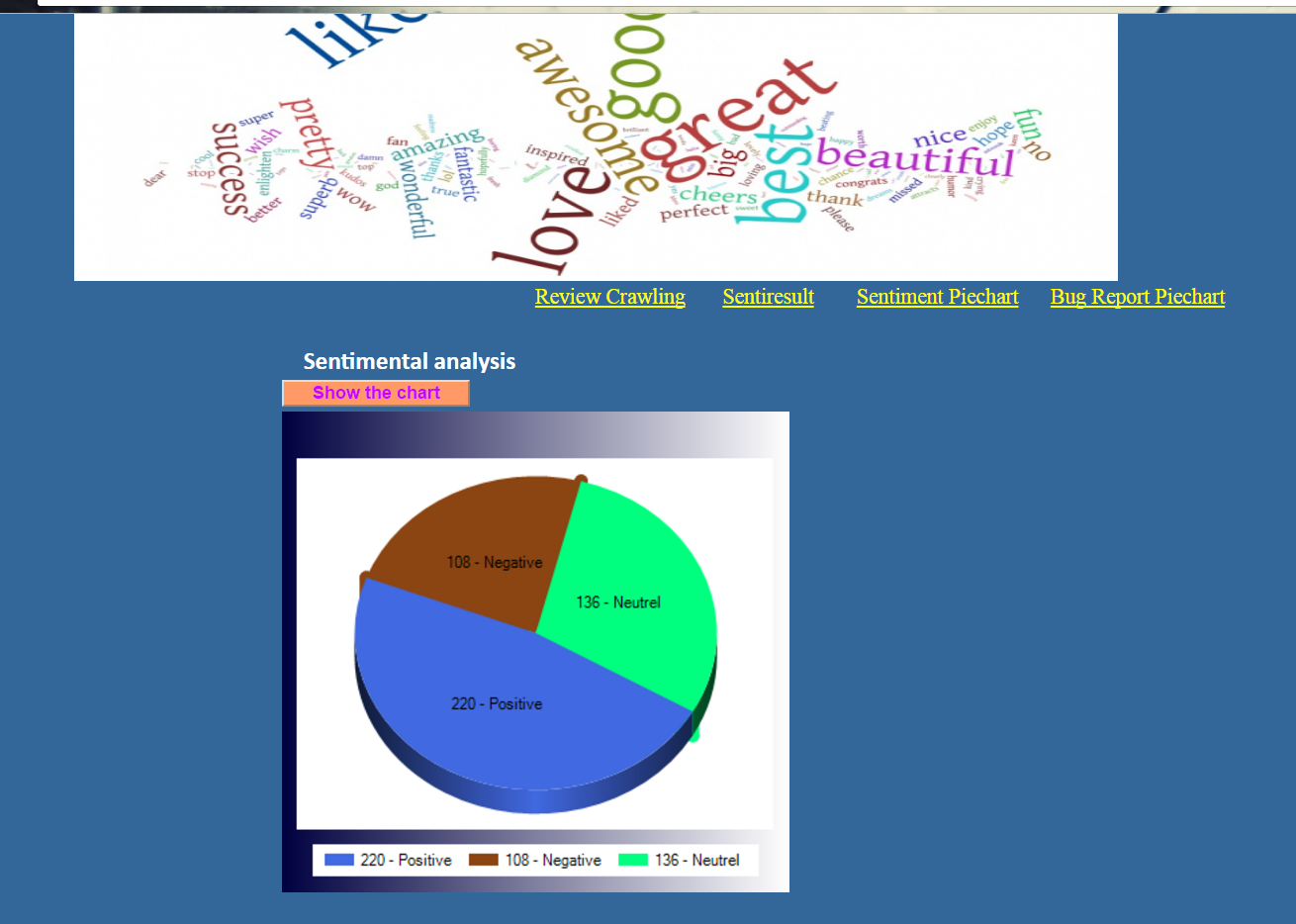
After calculating the sentimental polarity of each review as a whole, the overall positivity, negativity and objectivity is displayed.Now, each review is compared to the pre-defined corpus to identify whether its text contains a Feature Request/Suggestion or a Bug Report

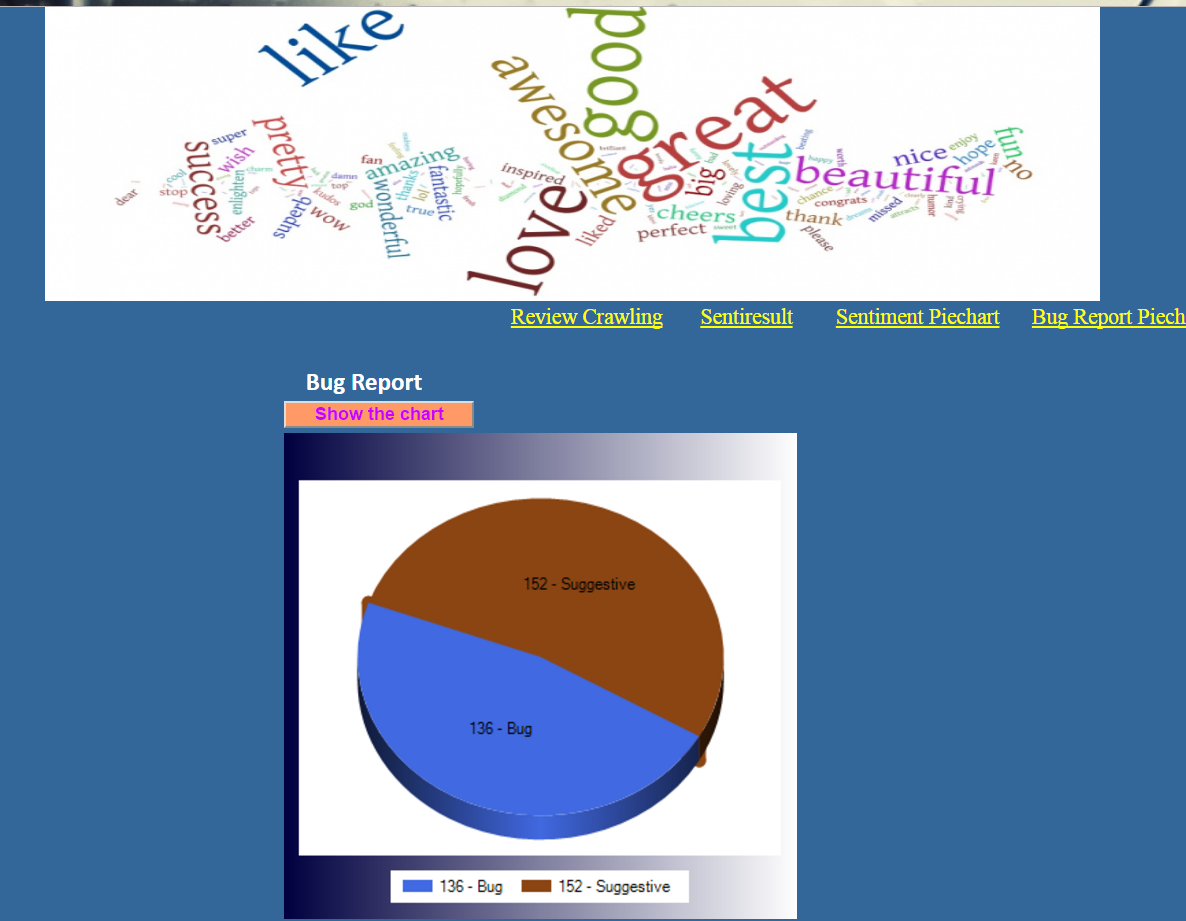
. 

* 1. **Output Visualization**

Finally, the overall output is visualized in the form of pie charts. For each application, the system will present 2 pie charts.

One will show the sentiment for that app. It will display the overall positive, negative and neutral sentiments based on total count of reviews. The other will display the bug reports and feature requests for the app. It will show the number of bug reports and the number of feature requests per app.





* 1. **Algorithm used in Corpus Based Approach**

We are using SentiStrength algorithm to classify the review as positive, negative and neutral based on the polarity score calculated for the review.

**SentiStrength** is an algorithm used for sentiment analysis, especially for short text and informal language.

**Working of SentiStrength**

There are a list of words in the corpus with a corresponding sentiment score assigned to it. Each sentence in the review is extracted and all the words in the corpus are checked against the words in the sentence. If a word in the corpus is found in the sentence, its score is added to the sentiment score of that sentence. If a booster word is found, for example, really, + or – 1 is added depending on the word after it. Emoticons and abbreviations are similarly scored.

This process is repeated for all the sentences in the review. Overall sentiment score of the review is found by aggregating sentiment scores of individual sentences in the review.

* The review is classified as ‘positive’ if the overall sentiment score of the review is greater than 0.
* The review is classified as ‘negative’ if the overall sentiment score of the review is less than 0.
* The review is classified as ‘neutral’ if the overall sentiment score of the review is 0.

For example, below are sample reviews of the NDTV news application and the corresponding sentiment and sentiment score.

|  |  |  |
| --- | --- | --- |
| Review Text | Sentiment Score | Sentiment |
| Notification is not good. Every notification flashes only half a sentence of news. Instead of giving a big photo in notification give the full headline of the news in it. | -2 | Negative |
| Love the new design, this update makes the application look good, feel good. But consuming much data compared to other news app and size is too big than other news app. But i like this update. Thank you developers. | 4 | Positive |

**CHAPTER 5**

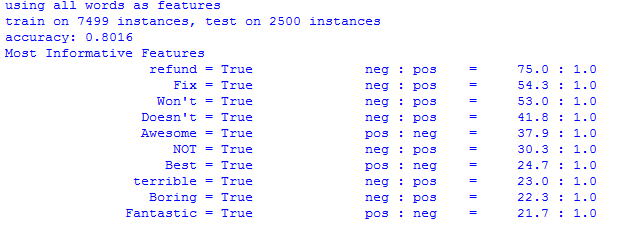
1. **RESULTS AND DISCUSSION**

There were 3 approaches used to perform sentiment analysis. The results for each are discussed below.

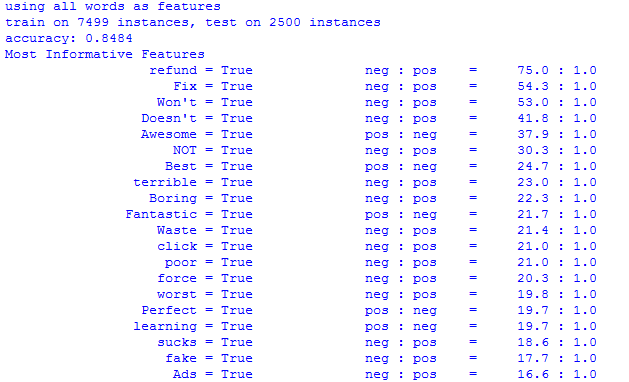
* 1. **Results for Corpus Based Approach**
  2. **Results for Classifier Based Approach**

The naïve Bayes classifier was used to classify the reviews as positive or negative. To do this, the total review dataset was split into training and testing sets which were used to train and test the classifier respectively.

The system showed a 80.16% accuracy when using all words as features. This was using a representative sample of 9999 reviews. The training set consisted of 7499 reviews and the testing set consisted of 2500 reviews.



Removing stopwords from the review was also performed. This was done using the stopword corpus which is available as part of the Natural Language Processing Toolkit. This further ratcheted up the accuracy to 84.84%. Again, the training set consisted of 7499 reviews and the testing set consisted of 2500 reviews.



* 1. **Results for VADER Based Approach**

The VADER based approach utilizes VADER which can be used as part of the Natural Language Processing Toolkit in Python.

**CHAPTER 6**

# CONCLUSION AND FUTURE WORK

Sentiment analysis or opinion mining is a field of study that analyzes people’s sentiments, attitudes, or emotions towards certain entities. Future work could include credibility checking of the data set by utilizing other sources of reviews and comparing how well the system responds to them.

The future work will aim to discover more linguistic patterns, making use of collocations for one. Also, using diverse datasets, perhaps with different languages will also enhance the system. Different app stores can also be incorporated to allow for a more holistic approach to the problem. Textual language also uses sarcasm – for example, ‘This app doesn’t even load. Good work guys.’ Sarcasm detection plays an important role in sentiment analysis: irony usually inverts the polarity of the sentence. A related question is dictionary size. At present there is little clear guidance about the optimal dictionary size for sentiment analysis. There are also a number of future directions for this research. One key question is the generalizability of our methods: To what extent do our approaches “travel well,” yielding valid results for widely-varying types of specialized vocabularies? A concern on this front has to do with variation in the usage of sentiment-laden words in different specialized contexts. To address this, the plan to apply existing methods to a range of different corpuses from various scientific, literary, and artistic fields of endeavor, and to texts drawn from both formal (e.g., official documents) and informal (message boards, blog posts) sources. Similarly, because the initial findings provide some evidence that our approaches may have advantages when applied to texts from earlier eras, we also plan to compare the performance of dictionaries constructed using our methods to standard ones as applied to older corpuses.

In conclusion, our work helps to filter reviews of interest for certain stakeholders as developers, analysts, and other users. Complementary within-app analytics such as the feature extraction, opinion mining, and the summarization of the reviews, will make app store data more useful for software and requirements engineering decisions.