Evaluation of User Perceived QoE in Mobile Systems
Using Social Media Analytics

By
Ahmed Yaseen Mjhool

Bachelor of Science
In Computer Engineering and Information Technology
University of Technology – Baghdad
2008

A thesis submitted to the College of Engineering at
Florida Institute of Technology
In partial fulfillment of the requirements for the degree of
Master of Science
In
Computer Engineering

Melbourne, Florida
December 2016
We the undersigned committee hereby approve that the attached thesis,

Evaluation of User Perceived QoE in Mobile Systems Using Social Media Analytics,

by Ahmed Yaseen Mjhool.

Dr. Carlos Otero
Advisor
Associate Professor
Electrical and Computer Engineering

Dr. Ivica Kostanic
Committee Member
Associate Professor
Electrical and Computer Engineering

Dr. Munevver Subasi
Committee Member
Associate Professor
Mathematical Sciences

Dr. Samuel Kozaitis
Department Head
Professor
Electrical and Computer Engineering
Abstract

Title: Evaluation of User Perceived QoE in Mobile Systems Using Social Media Analytics
Author: Ahmed Yaseen Mjhool
Advisor: Carlos Otero, Ph. D.

In mobile system environments, the quality perceived by users is inconstant and reliant on many factors such as cellular network, data connection, cost, coverage area, etc. Even though Quality of Service (QoS) management is enabled in most modern telecommunication systems, it does not guarantee the actual user’s perceived Quality of Experience (QoE) level. Many cellular networks rely on engineering test research, such as drive testing or smart mobile applications, to collect the required parameters in order to provide better service quality to users. However, this approach does not always yield customer satisfaction. Hence, user opinions should be considered. These opinions can be found via social media, and collected and processed via social media analytics models.

In this thesis research, a Rule-based algorithm is implemented. Based on this Rule-based algorithm, a sentiment analyzer is designed and tested. The results from testing the Rule-based algorithm are compared with results from a Naïve Bayes analyzer. In this thesis, the carrier Verizon is considered as the main topic and Twitter is considered the data source. This Rule-based algorithm and analyzer introduces a new method for generating datasets to easily design sentiment models. These models will
analyze users’ opinions to make better decisions and recommend the optimal QoE solutions.

The results of research conducted in this thesis show that the Rule-based analyzer performs better than the Naïve Bayes analyzer.
# Table of Contents

Table of Contents ........................................................................................................... v

List of Figures .................................................................................................................. vii

List of Tables .................................................................................................................... viii

Acknowledgement .......................................................................................................... ix

Dedication ........................................................................................................................ x

Chapter 1 Introduction ................................................................................................... 1
  1.1 Background and Motivation ..................................................................................... 1
  1.2 What is Sentiment? ................................................................................................. 1
  1.3 Machine Learning .................................................................................................. 3
  1.4 Natural Language Processing .................................................................................. 5
  1.5 Customer Sentiment ............................................................................................... 5
  1.6 Quality of user experience ..................................................................................... 6
  1.7 Project Objective ................................................................................................... 7
  1.8 Project Scope ........................................................................................................ 8
  1.9 Thesis Structure .................................................................................................... 8

Chapter 2 Literature Review ......................................................................................... 9
  2.1 Background Introduction ......................................................................................... 9
  2.1.1 Customer's Requirements of QoS ................................................................. 10
  2.1.2 QoS Offered by the Service Provider ............................................................ 11
  2.1.3 QoS Achieved or Delivered by the Service Provider ...................................... 11
  2.1.3 QoS Perceived by the Customer ...................................................................... 11
  2.2 Engineering Drive Test ........................................................................................ 12
  2.3 Artificial Neural Network Approach ..................................................................... 13
  2.4 QoE in Mobile Applications ................................................................................ 14
  2.5 QoE in Mobile Broadband Networks .................................................................. 16
  2.6 Social Media Analytics Approach ....................................................................... 17

Chapter 3 Naïve Bayes Classifier ............................................................................. 18
  3.1 What is Naïve Bayes? ............................................................................................ 18
  3.2 Why Naïve Bayes Classifier Assigned to this Work? ......................................... 19
  3.3 How does Naïve Bayes Classification Algorithm Work? ................................... 19
  3.3.1 Statistical Learning Background .................................................................. 20
  3.3.1.1 Product Rule ......................................................................................... 20
  3.3.1.2 Sum Rule ............................................................................................. 22
  3.3.1.3 Theorem of Total Probability .................................................................. 22
  3.3.2 Bayes Theorem ............................................................................................... 23
  3.3.3 Naïve Bayes Classification Algorithm ......................................................... 26
  3.3.4 Using Naïve Bayes to Classify ..................................................................... 31
  3.3.4.1 Being Naïve ......................................................................................... 32
  3.3.4.2 Train Naïve Bayes Classifier ................................................................. 37
### Chapter 4 Proposed Rule-Based Classifier

- **4.2 Rule Base Algorithm** .................................................. 38
- **4.3 Determining Topic and Dictionaries** .............................. 38
- **4.4 Collecting Data** ......................................................... 39
- **4.5 Accessing the Data** ..................................................... 40
- **4.6 Type of Opinions** ....................................................... 42
  - 4.6.1 Types of Evaluation ............................................... 42
  - 4.6.2 Types of Context ..................................................... 42
  - 4.6.3 Level of Interest ..................................................... 42
  - 4.6.4 Querying Formula .................................................. 42
- **4.7 How Rule-Based Classifier Works?** ............................... 43

### Chapter 5 Performance Evaluation

- **5.1 Evaluation Matrices** ................................................... 48
- **5.2 Cross Validation** ....................................................... 49
- **5.3 Evaluating Rule-Based Algorithm** .................................. 50
- **5.4 Performance Evaluation of Rule-Based vs Naïve Bayes** ........... 54

### Chapter 6 Summary, Conclusion and Future Work

- **6.1 Summary** ................................................................. 56
- **6.2 Conclusion** ............................................................... 56
- **6.3 Future Work** ............................................................. 57

### References

- .................................................................................................. 58
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sentiment Analysis Concept</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Natural Language Interface to a knowledge base [36]</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>The 4 Viewpoints of QoS [14]</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>Drive Testing Overview [15]</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>Facebook and YouTube differentiation [21]</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>QoS and QoE in a Mobile Systems [22]</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>QoE evaluation methodology layers in networking apps [21]</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>Venn diagram of product rule</td>
<td>21</td>
</tr>
<tr>
<td>9</td>
<td>Venn diagram of a zero-product rule</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>Example of Theorem of Total Probability</td>
<td>23</td>
</tr>
<tr>
<td>11</td>
<td>NB classifier example</td>
<td>27</td>
</tr>
<tr>
<td>12</td>
<td>NB classifier example with new white object</td>
<td>28</td>
</tr>
<tr>
<td>13</td>
<td>Probability of Class of positive or negative tweet</td>
<td>31</td>
</tr>
<tr>
<td>14</td>
<td>Rule-Based Algorithm Flowchart</td>
<td>43</td>
</tr>
<tr>
<td>15</td>
<td>Rule-Based Processing Steps</td>
<td>45</td>
</tr>
<tr>
<td>16</td>
<td>The Main Functions of Rule-Based Algorithm</td>
<td>46</td>
</tr>
<tr>
<td>17</td>
<td>Precision and Recall [36]</td>
<td>49</td>
</tr>
<tr>
<td>18</td>
<td>Four k-fold stratification method</td>
<td>50</td>
</tr>
<tr>
<td>19</td>
<td>STS-Test dataset statistics</td>
<td>51</td>
</tr>
<tr>
<td>20</td>
<td>Classification of STS dataset according to Rule-Based</td>
<td>51</td>
</tr>
<tr>
<td>21</td>
<td>Rule-Based Metrics</td>
<td>53</td>
</tr>
<tr>
<td>22</td>
<td>Rule-Based Metrics</td>
<td>54</td>
</tr>
<tr>
<td>23</td>
<td>Naive Bayes Matrices</td>
<td>55</td>
</tr>
</tbody>
</table>
List of Tables

Table 1 - A handful of tweets classified manually..................................................34
Table 2 - Tweet classification example [25].................................................................36
Table 3 - Public datasets that available to the research community [32]..................37
Table 4 - Number of Collected Data.................................................................39
Table 5 - Performance Evaluation Metrics [26]..................................................48
Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor Dr. Carlos Otero for the continuous support of my master study, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my master study!

Furthermore, my warm sincere thanks also go to Dr. Nidhal El Abbadi for his endless encouragements and supports to gain full benefits of Academic Training experience of my master degree. More, I am very thankful to my close friends, Ahmed Al-Jobouri, and Raoof Hazim, for their generous support and for the hard paper work they been doing since the first day of my master study. They are amazing and absolutely proved the concept of friendship.

Last but not the least; I am very grateful to my parents, my sisters, and my brothers, for their infinite spirit support and patience since I started my study abroad.
Dedication

To my parents

To my sisters

To my brothers

To my future wife

I dedicate my master’s thesis work.
Chapter 1

Introduction

He who molds the public sentiment ... makes statues and decisions possible or impossible to make. —Abraham Lincoln

1.1 Background and Motivation

For many years, researchers have used survey methods to get data for their research. This data sometimes target the way people think toward some cases. Survey adds many preparations and waste resources for the sponsor and it can be boring and time consuming for the responder. Till date, old-fashioned tools such as surveys and polls are still used by sponsors, and if they want to collect customers’ opinion they may have to give some encouragements to the responders like coupons or gifts in order to get their feedback. Normally, this is not an efficient way when we need to gather very high number of responses in short time. Nowadays, availability and popularity of public opinion-rich resources, like social media and internet products’ reviews, have grown dramatically which give researchers the opportunity to use information technology and to seek out a smart algorithm that could help simplifying people’s opinions. According to V. Sharma [1], motivation for smart opinion mining which is also called “Sentiment Analysis” has two-folds, consumers and producers, this two-folds approach highly values “customer’s opinion” about products and services. Hence, Sentiment Analysis has seen a significant effort from business as well as academia.

1.2 What is Sentiment?

The work of Sentiment Analysis (SA) started in early 2000s with a seminal paper by Bo Pang & colleagues [2]. SA and opinion mining both mostly used in
academia while the term SA is commonly used in industry. The name sentiment analysis possibly first appeared in [3] while the term opinion mining was first written in [4]. Many other terms with slightly different task are categorized under the umbrella of SA such as sentiment mining, opinion extraction, opinion mining, subjectivity analysis, emotion analysis, affect analysis, review mining, etc. [5].

SA is one of the most broadly studied applications of Natural Language Processing (NLP) and Machine Learning (ML). This pitch has developed massively with the beginning of the Web 2.0 technology. According to V. Sharma [1], the goal of SA is to harness random big data in order to obtain important information regarding public opinion, which would help build an accurate mathematical model for smarter business decisions, political campaigns and better product consumption. It focuses on identifying whether a given piece of text is subjective or objective and if it is subjective, then whether it is negative or positive.

The crucial concern in SA is to find how sentiments are expressed in texts and whether these expressions are categorized as positive or negative opinions toward the subject. Thus, SA involves identification of Sentiment expressions, polarity of the expressions and their relationship to the subject. These features are connected to each other. For instance, in this sentence, “@verizon better than @tmobile”, the expression “better than” denotes a positive sentiment toward verizon and a negative sentiment toward tmobile.

*Verizon and T-Mobile are among the large mobile communication technology companies in USA.*
Jigsaw Academy, the online school of analytics, gave a general understanding of SA as seen in Figure 1.

Figure 1 - Sentiment Analysis Concept

1.3 Machine Learning

Machine Learning is an Artificial Intelligence (AI) subfield that is letting computers to learn and build a smart system to help human to predict the results based on trained model. In other word, algorithm is inserted within data and this leads to conclude information about the properties of that data. This information, is called patterns if nonrandom, lets algorithm create predictions about new data that might come in the future. Since almost all nonrandom data contains patterns, this patterns is used to train a model with what it decides are the significant features of the data [6].

For instance, to understand how the model works, think about a simple example of email filtering. Suppose you are receiving many spam emails that contain the words “Free data!”. As Human, you easily would figure out that any email containing these words “Free data!” is a spam and should move to spam folder or
trash. This way is considered as a generalization—quickly a mental model of what is spam generated in our mind. Same concept, ML algorithm allows computers to learn from our actions while we reporting spam emails and as a result it would be improved in preventing any other new spam emails that might come and should create same generalization [6].

Basically, ML algorithms rely heavily on mathematics and statistics. To understand them easily you need good knowledge in math. Broadly, there are three types of ML algorithms such as [7]:

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

More, these algorithms can be used to design solution model to almost any data problem. Those algorithms commonly used are:

1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. Support Vector Machine (SVM)
5. Naive Bayes
6. K-Nearest Neighbors algorithm (KNN)
7. K-Means
8. Random Forest
9. Dimensionality Reduction Algorithms
10. Gradient Boost & Adaboost
11. Neural Networks
1.4 Natural Language Processing

The concept NLP incorporates a wide set of techniques for computerized generation, manipulation and analysis of human languages. Although most NLP techniques inherit mostly from Linguistics and AI, they are also affected by relatively newer technologies such as ML, Cognitive Science and Computational Statistics [8].

![Image](image.png)

**Figure 2 - Natural Language Interface to a knowledge base [36]**

1.5 Customer Sentiment

The latest role of social media in improving customer experience actions in US and around the world has given an extent of studies on mining of customer speech online. Automated SA, developing technology that overlaps with many others, is hard to measure such as business intelligence, customer service, brand reputation management, and market. Various styles of sentiment software use a technology
commonly known as text analytics, which extracts intuition from text, such as in social media, news articles, etc. The report by R. King [9] states that the market for text analytics alone may rise to $978 million in 2014 from $499 million in 2011, according to an October 2009 report by Forrester Research (FORR).

Just to clear the confusion between text analytics and SA, L. Sigler in his article [10] explains the differences and said, they are both ways to derive meaning from customer data, and they are both critical components of a successful customer experience management program. However, they are not the same thing. Text analytics is the process of analyzing unstructured text, extracting relevant information and transforming it into useful business intelligence. SA determines if an expression is positive, negative, or neutral and also determines its polarity. In other words, text analytics studies the face value of the words, including the grammar and the relationships among the words. Simply put, text analytics gives you the meaning. Sentiment analysis gives insight into the emotion behind the words.

1.6 Quality of user experience

If we need to give an accurate definition to the term “Quality of user experience”, which is called (QoE), we need to break down the term into three parts Quality, Experience and User Experience to then understand their meaning separately. Diepold in his report [11] showed a professional quest; thus, he wrote a professional definition for Quality of Experience term with its parts as below:

Quality: is the outcome of an individual’s comparison and judgment process. It includes perception, reflection about the perception and the description of the outcome.

Experience: is an individual’s stream of perception and interpretation of one or multiple events.
**QoE**: is the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the user’s personality and current state.

Furthermore, the document ISO 9241-210 defines user experience as “a person’s perceptions and responses that result from the use or anticipated use of a product, system or service”. According to the ISO definition, user experience includes all the users’ emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviors and accomplishments that occur before, during and after use. The ISO also list three factors that influence user experience: system, user and the context of use.

Besides, Nielsen and Norman in their Group [12] shared article about User Experience definition summarized "User Experience" as an experience that encompasses all aspects of the end-user's interaction with the company, its services and its products. However, true user experience goes far beyond giving customers what they say they want, or providing checklist features. In order to achieve high-quality user experience in a company's offerings there must be a seamless merging of the services of multiple disciplines, including engineering, marketing, graphical and industrial design and interface design.

### 1.7 Project Objective

The main objective of this research work is to determine how customers on Verizon services are happy in term of quality of user experience. More, this research aimed to improve QoE from user perspective. In addition, this work hopes to develop an algorithm that could have a better accuracy to generate a dataset from twitter platform to any specific machine-learning algorithm to build SA model to any topic easily. Parsing user’s tweets on Twitter in a specific topic is used for generating labeled dataset.
1.8 Project Scope

This project consists of creating Rule-Based classifier based upon user’s opinion. More, this approach considers tweets that mentioning Verizon official accounts and in English language only. The classification of these tweets under Rule-Based classifier is limited into four categories: positive, neutral, negative, advertisement or retweet. As a process of feature selection, feature will be considered in the form of ‘unigrams’. Moreover, this project is compared with Naïve Bayes classifier and tested with Stanford university dataset. This project will help feedback analyzer to categorize opinions and give a chance of improving QoE quickly and easily.

1.9 Thesis Structure

This thesis structured as the following chapters: (1) Introduction; (2) Literature Review; (3) Naïve Bayes Classifier; (4) The Proposed Rule-Based Classifier; (5) Performance Evaluation and lastly (6) Summary, Conclusion and Future Work.
Chapter 2
Literature Review

Many carrier companies try to improve their services based on data collection and analysis of coverage area. However, the techniques those companies are using never reflects the real customer satisfaction. There are some hidden problems that carriers and customers face such as drop calls, data limitation, poor voice quality, etc., which affects QoS. It essential to browse what researchers have done for QoS from the customer’s perspective. In this literature review, most recent researches on evaluation of QoS and QoE for mobile carrier companies will be discussed.

2.1 Background Introduction

Although a brief definition of QoE was stated in chapter 1, a redefinition of QoE with another concept will shortly be covered. Since we are going to study the most relevant researches about mobile cellular network, the quality concept will be defined based on International Telecommunication Union’s definition (ITU).

According to ISO 8402, quality is defined as “the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs”. The ITU-T Rec. E.800 [13] defined the QoS as “The collective effect of service performances, which determine the degree of satisfaction of a user of the service”. More from ITU-T Rec. G.1000, QoS viewpoint can be categorized into four terms (see figure 3):

1. QoS offered/planned by provider: the quality that is expected to be offered to the users.

2. QoS delivered/achieved by provider: the quality that is expected to be delivered to the users.
3. QoS perceived by user/customer: the quality that users believe they have experienced.

4. QoS requirements of user/customer: the level of quality required by a service of users’ applications.

![Diagram showing four viewpoints of QoS](image)

**Figure 3 - The 4 Viewpoints of QoS [14]**

As seen from the Figure (3) above, we have two main parts (customer and service provider (SP)) and four features that connect in a life cycle, it starts from customer’s QoS requirements which is basically what encourage the provider to offer the service and this service should achieve QoS before it is delivered to the user to reach quality satisfactions.

### 2.1.1 Customer's Requirements of QoS

The user does not really care about how particular services are provided or any other internal support design, but only care about the resulting end-to-end service quality.
2.1.2 QoS Offered by the Service Provider
The primary use of this form of QoS is for planning documents and for Service Level Agreements. In the example mentioned in [14], the SP may state that the availability of basic telephone service is intended to be 99.95% in a year with no more than a 15 minutes’ break at any one incident, and no more than 3 breaks over the year.

2.1.3 QoS Achieved or Delivered by the Service Provider
This QoS achieved is usually used by industry. It should be like what is stated for the offered QoS so both of those achieved and offered of QoS can be compared to decide what was truly achieved to evaluate the level of performance reached. For example, the SP may have specified that the achieved availability for the previous quarter was 99.95% with 5 breaks of service, one of which lasted 65 minutes.

2.1.3 QoS Perceived by the Customer
Perceived QoS can be expressed in term of user’s satisfaction other than being expressed by a technical term. It is measured by user surveys and from user's own comments on levels of service. It can be used by the SP to measure user’s satisfaction of the service quality. For instance, a user may state that on an unacceptable number of cases, there was problem with the network and it was not possible to make calls, thus, the customer may choose to give a rating of 2 out of 5 of which 5 indicates an excellent service. Ideally, there would be a 1:1 correspondence between delivered and perceived QoS.

User satisfaction is the main goal that every company is trying to achieve. Recently, cellular operators have tackled the need for a major shift from technical quality requirements to customer experience (CX) guarantees. In telecommunications networks, QoS is studied by many researchers in several aspects such as engineering drive test, tracking tools, end-user application, neural
network approach and social media QoE analytics. Next, we are going to refer to some research that focuses on improving QoS or QoE in mobile systems.

2.2 Engineering Drive Test

Drive test approach is used for cellular network enhancements, network performance monitoring and Radio Access network (RAN) optimization. This approach is commonly used in mostly mobile carrier and deployed in logistical area that QoS need to be proven. The Reference Signal Received Power (RSRP) is a fundamental indicator of the coverage and QoS in the LTE cellular communication networks. See drive test setup overview in figure 4.

![Drive Testing Overview](image)

**Figure 4 - Drive Testing Overview [15]**

In his thesis study 2014 [16], Egi conducted a drive test experiment in a specific Melbourne area with receiver and phone in order to rank the measurement equipment based on their ability in collecting RSRP. Drive test he used was based on major and detailed roads and comparison between equipment. Various mathematical techniques such as interpolation and other statistical methods were
used to characterize the quality of the drive test equipment based on the RSRP data collection. According to his RSRP measurement he figured out that AT&T has a better QoS and QoE as well.

Another thesis study [17], Eyceyurt studied and statistically analyzed the performance of two US mobile carriers (T-Mobile and AT&T) in Melbourne Florida in term of video streaming in LTE technology using drive test approach. The two carriers were scanned according to particular parameters such as RSRP, Reference Signal Received Quality (RSRQ), Signal to Interference & Noise Ratio (SINR) and spectral efficiency values. Eyceyurt observed that T-Mobile has better QoS across all other parameters.

More, Saleh in his thesis experiment [18], radio frequency driving test, evaluated the downlink spectral efficiency and the Multiple Input Multiple Output performance and its different transmission modes in T-Mobile cellular network around Melbourne FL area. This study intends to prove QoS of SP for internet data, video streaming and other related parameters to get better user QoE.

### 2.3 Artificial Neural Network Approach

The paper written by Anchuen, Uthansakul, et al. [19] proposed the estimation model of user satisfaction in terms QoE by using neural network approach. This paper intends to evaluate QoE using QoS parameters rather than evaluating by the users. The factor of service is computed from the QoS measuring tool which is nPerf® and the opinion score is computed by the AMOS** application.

---

**nPerf** is license-free monitoring program used to collects all QoS parameters.

**AMOS** is popular application or measuring tool on smart phone that used to help users to train their QoE model.
The results of AMOS and QOS parameters were used for learning in the neural networks in order to create QOE model. The experiment results revealed a good agreement between QOE value from the AMOS applications and QOE value from the proposed model. This is a very supportive for SP to understand the satisfaction of users based on viewpoint of network performances.

In their conference paper [20], Rivera et.al. considered the use of neural networks to reduce the gap between the user’s perception of the service and the SP. They proposed a methodology to support the network performance decision to provide mobile internet services based on users’ perception. Simply what is done in this study, a neural network model was built to construct the correlation between QoS and QoE in order to create making decision tool that support SP’s decision. This model allows the SP to collect data constantly and prevent rejections by the users in their services or products.

2.4 QoE in Mobile Applications

Consider data internet service as example, not only the SP is responsible of user’s experience but also the application user installed in his/her phone play big role in performance topic. For instance, as proven in [21], as seen in figure 5 YouTube QoE is highly sensitive to downlink-encoding bottlenecks, while Facebook generate the low traffic volume. Facebook application is very robust to changing network conditions due to the different optimization techniques they used.
The paper [22] gives a better understanding about factors that influenced QoE and focused on evaluating qualitative users’ perceived experience on users’ phones, based on the set of mobile applications they use in their daily life. Figure 6 shows the subjective and objective quality of mobile services. The authors employed 30 users using Android OS and with 3 sorts of phones (Samsung, HTC and Motorola) subscribed to 4 SP such as (Verizon, Sprint, T-Mobile and AT&T).

According to this work [22], one user interviewed and said:” My phone can operate on a 4G network, but I usually keep it set to 3G because in my experience,
the 4G is not considerably faster and just eats up my battery. Generally, I keep 4G turned off unless I am doing something network intensive and I know it is available”. This is not as what authors was expected to said, because the acquired performance measurements showed better QoS in 4G than 3G. presumably, the user using applications worked perfectly well on 3G mobile network.

Procera networks declared its new ScoreCard solution [23], it evaluates QoE indicators from different way such as Internet browsing and streaming-video applications, as well as gaming, social media, and uploads and downloads.

2.5 QoE in Mobile Broadband Networks

This paper [21] analyzed QoE as assessed with a specific scenario. The authors used a group of 33 mobile broadband users in a field trial. Crossing an evaluation period of 31 days, users using their own 3.5G-connected devices, they frequently stated their perceived experience on surfing their favored YouTube and Facebook contents under changing network conditions. Authors in this scenario targeting a three-layered evaluation methodology as seen in figure 7, including monitoring of the network-layer QoS, monitoring network and application layers, and QoE assessment at the user-layer. In conclusion, this paper offers the consequences on the evaluation of different network conditions on YouTube and Facebook from the end-user perspective, considering daily life Web usage scenarios.
2.6 Social Media Analytics Approach

From what is listed above we can see how engineering approaches evaluate the QoS and QoE in different methodologies with subjective and objective directions. Many factors and parameter should be considered especially in QoE measurements and that vary per user mode. However, does those studies really reflects user experience? How about the feedback user post on social media platforms, can we use those platforms to develop an easy smart way such as an algorithm to evaluate perceived user QoE matrices? This idea may be totally new for researcher, no papers or work done in this field except Andrew Moss & Dr. Eric Jensen titled it in their workshop [24] and discussed QoE from analyzing social media for feedback and evaluation an organization and its activities.

In this thesis approach, the answer of these questions will be found and rule base algorithm implemented to classify positive and negative feedback tweets that are coming from user using twitter as a source of collecting data.
Chapter 3
Naïve Bayes Classifier

3.1 What is Naïve Bayes?

Naïve Bayes (NB) is a machine learning algorithm and it is considered one of the simplest learning algorithms. There are many machine learning algorithms (as seen in the list below) that deal with mining sentiment in social media or any text processing and each has its own characteristics.

1. Support Vector Classifiers (LinearSVC, PolynomialSVC, RbfSVC, NuSVC)
2. Maximum Entropy Model (GIS, IIS, MEGAM, TADM)
3. Gradient Boosting Classifier
4. Random Forest Classifier
5. Logistic Regression
6. Bernoulli NB
7. Gaussian NB
8. Multinomial NB
9. K Neighbors Classifier
10. Linear SVC
11. NuSVC
12. Decision Tree Classifier

Why is it called naive? The naive was given to classifier for one supposition that is needed for Bayes to work in excellent way: all features must be independent of each other. This, however, is rarely the case for real-world applications. Nevertheless, it still proceeds with very good accuracy in practice even when the independent assumption does not hold [25].
3.2 Why Naïve Bayes Classifier Assigned to this Work?

The NB used in this thesis is based on Bayes Theorem to predict the probability of a particular label. The NB classifier is commonly used because of its simplicity, it proves to be robust to irrelevant feature, and it learns fast and does not need lots of storage.

In addition, [26] cite that “When the amount of training data is low, MaxEnt and SVM tend to produce better performance than NB. However, these reported performance gains may tend to dissipate as NB’s performance increases with very large amounts of training data”.

Here are some advantages of NB [27]:

- Fast to train (single scan). Fast to classify.
- Not sensitive to irrelevant features.
- Handles real and discrete data.
- Handles streaming data well.

And disadvantages:

- Assumes independence of features.

3.3 How does Naïve Bayes Classification Algorithm Work?

Before discussing the way of SA works with NB classifier and before behind NB, some statistical math functions should be addressed.
3.3.1 Statistical Learning Background

In this background, a quick and brief review of statistical math to learn ML techniques of NB that is commonly used in data analysis will be discussed. NB is very popular ML algorithm for sentiment classification. As mentioned before NB classifier algorithm is based on Bayes Theorem with conditional independence expectations. In another word, NB classification model calculates the posterior probability of a class based on the words distribution in the particular document. The model works with the bag of words feature extraction which ignores the position of the word in the document.

NB relies on the following probability theory:

1. Product rule,
2. Sum rule,
3. And the theorem of total probability.

3.3.1.1 Product Rule

When more than two events occurring at same time, the product rule can be used. Suppose you have event A and event B with the probability P(A) and P(B) respectively, the probability of events A and B occurring at the same time is P(A ∩ B) = P(A | B) P(B). To demonstrate this equations, see Venn diagram in the Figure (8) below, where P(A) is represented by green circle on the left, P(B) is represented by red circle on the right, and the intersection of circle A and circle B denoted to the both probabilities P(A ∩ B).
As shown in the intersection of event A and event B the probability of both events A and B are occurring. That is, the P(A) and P(B) occurring is the product of the probability of A occurring given that B occurred and the P(B) occurring. Mathematically, the shared area between the two circles could be written where 
\[ P(A \cap B) = P(B \cap A) = P(B \mid A) \cdot P(A). \]

For independent events, when, for instance, the result of A does not affect the result of B and vice versa, it is clearly gotten that the P(A) occurring given that B occurred is equal to P(A) and P(B) occurring given that A occurred is equal to P(B). Therefore, the probability in the intersection of the events P(A ∩ B) in figure 3 can be written as 
\[ P(A \cap B) = P(A) \cdot P(B). \]

Lastly, the joint probability of events A and B P(A ∩ B) = 0 when mutually exclusive events A and B events that cannot occur at the same time, as shown in Figure (9) below.
3.3.1.2 Sum Rule
This can be used when the probability at least one of more events are computing. In other words, the probability of event A donated as P(A) and probability of event B donated as P(B), the probability of at least one event occurring is written as \( P(A \cup B) = P(A) + P(B) - P(A \cap B) \). To demonstrate this model, suppose again the Venn diagram in Figure (8). The area at the intersection of A and B represents a double-counting when calculating the probability of event A occurring or event B occurring, thus it must be taken from the sum of both probabilities. Since \( P(A \cap B) = 0 \) in mutually exclusive events, the sum rule can be used as \( P(A \cup B) = P(A) + P(B) \).

3.3.1.3 Theorem of Total Probability
It is a fundamental law relating marginal probabilities to conditional probabilities. Also, it applies to set of events that are mutually exclusive with exhaustive results. For instance, suppose events \( \{A_1, A_2… A_n\} \) thus \( \sum_{k=1}^{n} P(A_k) = 1 \), at that point the theorem of total probability can be used to calculate the probability of B as the following equation \( P(B) = \sum_{k=1}^{n} P(B \setminus A_k)P(A_k) \) to clarify this equation consider Figure 5 where the \( A_k \) events represented as the rectangle area as seen \( A_1,\ldots,A_5 \). And the probability of B is donated to the red ellipse B. From the
shown Figure (10) below, easily we can realize the P(B) can be computed as a function of the sum of the probability of each event $A_k$ multiplied by the P(B) occurring given event $A_k$.

![Figure 10 - Example of Theorem of Total Probability](image)

### 3.3.2 Bayes Theorem

Bayes Theorem is primary rule to all Bayesian learning approaches. It provides a framework for statistical assumption in many ML applications. A key feature of the Bayes Theorem is its capability to account for prior beliefs when making assumptions about specific events from a given event space.

To exemplify the Bayes Theorem, the common example of concluding whether a patient has cancer or not will be used [28]. Suppose an imperfect process when determining lab results, where:

The symbol (+) reflects a positive test result,

The symbol (-) reflects a negative test result,

And the symbol (¬) reflects negation.
The Lab test results of a patient with cancer can be summarized as below:

\[ P(\text{cancer}) = .008, \quad P(\neg\text{cancer}) = .992 \]
\[ P(\oplus|\text{cancer}) = .98, \quad P(\oplus|\neg\text{cancer}) = .02 \]
\[ P(\negoplus|\neg\text{cancer}) = .03, \quad P(\negoplus|\text{cancer}) = .97 \]

With the results above, the assignment now is to check whether a patient has cancer (or not) assuming the Lab result has given positive result. (The lab-result process is not hundred percent accurate, so there might be some error on the results.) In this example, there is a hypothesis space named as \((H)\) containing two hypotheses, one indicating that the patient has cancer represented \((h1)\) and other indicating that the patient does not have cancer represented \((h2)\). These hypotheses belong to the set of hypothesis \((H)\), where, \(h \in H\).

The listed data results as shown above, provides the probability of observing a particular event \(h\) from event space \(H\). This means, \(P(h)\) represent the prior knowledge based on old data and it is captured as \(p(h)\). We also have access to the probability of observing a particular lab result which represent \((D)\) so that probability of \(D\) as \(P(D)\) reflects our prior knowledge of observing a particular lab result. To calculate \(P(D)\), we use the previous theorem of total probability equation as follows.

\[ p(D) = \sum_{i=1}^{n} p(D|h_i)p(h_i) \]

The summarized Lab test results provides information about the probability of observing a particular lab result given some hypothesis \(i\). Thus, the probability of observing lab result \((+)\) given that a patient actually has cancer is generically represented by the conditional probability \(P(D | h)\). The \(P(D | h)\) represents the probability of \(D\) given \(h\). Therefore, a summary of the data can provide us with \(P(h)\) and \(P(D | h)\); however, the question that is being inquired is to decide if the patient should be infected with cancer given the test results \(P(h | D)\).
To solve this problem, Bayes theorem provides the framework for relating these probabilities. See below.

\[ p(h|D) = \frac{P(D|h)P(h)}{P(D)} \]

Where,

- P(h) is the probability of the hypothesis before seeing the data, or prior probability.
- P(h|D) is the probability of the hypothesis after seeing the data, or posterior probability.
- P(D|h) is the probability of D given h.
- P(D) is the probability of the data under any hypothesis, which is a normalizing constant.

Bayes theorem also provides the means for updating the probability of a hypothesis (h) in light of some body of data. This happens when the probability of hypothesis changes over time [29]. Bayes theorem can be proven by multiplying each side by P(D); this produces the following property, \( p(h\nD) \ p(D) = p(D|h) \ p(h) \) which (as mentioned on the Product Rule) is basically the probability of a conjunction of events. From this book [30], we can expect whether the positive lab report observed should lead to a positive cancer diagnosis, as written below.

\[
p(h = \text{"cancer"}| D = \text{"positive"}) = \frac{p(D = \text{"positive"}|h = \text{"cancer"})p(h = \text{"cancer"})}{p(D = \text{"positive"})}
\]

\[
p(h = \text{"not cancer"}| D = \text{"positive"}) = \frac{p(D = \text{"positive"}|h = \text{"not cancer"})p(h = \text{"not cancer"})}{p(D = \text{"positive"})}
\]
After inserting appropriate probabilities from summarized Lab test results, we can get the results below.

\[
p(h = "cancer" | D = "positive") = \frac{(0.98)(0.008)}{0.0376} = \frac{0.00784}{0.0376} = 0.21
\]

\[
p(h = "not cancer" | D = "positive") = \frac{(0.03)(0.992)}{0.0376} = \frac{0.02976}{0.0376} = 0.79
\]

Hence, based on Bayes theorem, in this case, we can predict that the patient does not have cancer [26].

3.3.3 Naïve Bayes Classification Algorithm

The NB Classifier technique is based on the Bayesian theorem and is particularly appropriate when the dimensionality of the inputs is high. Despite its simplicity, NB can often outperform more sophisticated classification methods. It takes into account prior knowledge to estimate the conditional probabilities of each feature set.

To explain the concept of NB Classification, we are going to refer to the example in [31]. Consider the sample data that is displayed in Figure (11) below. As shown, the objects can be classified as either red or green. Our job is to classify new data as they arrive, that is, choose to which class label they belong, based on the currently existing objects.
Since green objects are double red objects, it is sensible to believe that a new future case is twice as probable to have membership green rather than red. In the Bayesian theorem, this belief is the prior probability and these prior probabilities are based on previous experience, here in this situation the percentage of green and red objects are often used to expect outcomes before they really happen. Hence, we can say:

$$\text{Prior probability for } \text{GREEN} \propto \frac{\text{Number of GREEN objects}}{\text{Total number of objects}}$$

$$\text{Prior probability for } \text{RED} \propto \frac{\text{Number of RED objects}}{\text{Total number of objects}}$$

We have 60 objects, 40 of which are green and 20 red, hence, our prior probabilities for class membership are:

$$\text{Prior probability for } \text{GREEN} \propto \frac{40}{60}$$

$$\text{Prior probability for } \text{RED} \propto \frac{20}{60}$$
We expressed our prior probability, then we are ready to classify a new white object as seen in Figure (12). The objects are clustered well, this make it easy to assume that the more (green or red) objects in the neighborhood of X, the more expected that the new cases belong to that specific color. To measure this probability, A circle around X was drawn which includes a number (to be chosen a priori) of points regardless of their class labels. Then we determine the number of points in the circle belonging to each class label. From this we compute the probability:

\[
\text{Likelihood of } X \text{ given } \text{GREEN} \propto \frac{\text{Number of GREEN in the vicinity of } X}{\text{Total number of GREEN cases}}
\]

\[
\text{Likelihood of } X \text{ given } \text{RED} \propto \frac{\text{Number of RED in the vicinity of } X}{\text{Total number of RED cases}}
\]

From the illustration above, it is clear that Probability of X given green is smaller than Probability of X given red, since the circle includes one green object and three red ones. Hence:

\[
\text{Probability of } X \text{ given } \text{GREEN} \approx \frac{1}{40}
\]
Even if the prior probabilities indicate that X may belong to green (given that there are twice as many green compared to red) the probability indicates otherwise; that the class membership of X is red (given that there are more red objects in the neighborhood of X than green). In the Bayesian, the final classification is created by combining both sources of information to form a posterior probability using the so-called Bayes' rule (named after Rev. Thomas Bayes 1702-1761).

\[
\text{Probability of } X \text{ given RED } = \frac{3}{20}
\]

Finally, in this example we mentioned, we classify X as red since its class membership reaches the biggest posterior probability.

One more thing we need to know is that the above probabilities are not normalized. However, this does not affect the classification outcome since their normalizing constants are the same.

Another simple example, consider a set of features \(X = \{x_1, x_2, x_3, x_4\}\) where \(x_i\) is a specific feature that can be used to classify some object based on class \(C = \{c_1, c_2, \ldots, c_n\}\). Using the Bayes Theorem, the problem can be expressed as follow:

\[
p(c|x_1, x_2, x_3, x_4) = \frac{p(x_1, x_2, x_3, x_4|c)p(c)}{p(x_1, x_2, x_3, x_4)}
\]
A significant component of the NB classifier algorithm is the assumption of conditional independence between the features $x_1$, $x_2$, $\ldots$, $x_n$. That is, NB classification algorithm makes the simplifying assumption that the presence or absence of a particular feature $x_i$ is not associated to the presence or absence of any other feature $x_j$, given a target class. Supposing conditional independence and after applying the theorem of total probability, we can re-write previous equation as follow:

$$p(c|x_1, x_2, x_3, x_4) = \frac{P(x_1|x_2, x_3, x_4 | c)P(c)}{\sum_{i=1}^nP(x_1, x_2, x_3, x_4 | c_i)P(c_i)} = \frac{P(x_1 | c)P(x_2 | c)P(x_3 | c)P(x_4 | c)P(c)}{\sum_{i=1}^nP(x_1, x_2, x_3, x_4 | c_i)P(c_i)}$$

In the last two equations, $p(c \mid x_1, x_2, x_3, x_4)$ reflects the posterior probability of observing a class $c$ given the set of features, since the probability is calculated after we have observed data $X$. To classify an object based on a set of features, the last equation can be used to calculate the probability for each class $c_i$, and the class that returns the maximum posterior probability is selected as the object’s classification. Therefore, assuming two classes $c_1$ and $c_2$, the equation above would be used to calculate the posterior probability of each class and the maximum probability returned would be used as basis for classification. A specific detail to be considered during the calculation of the posterior probability for each class is that the denominator is a constant normalization factor for all classes. Therefore, since it is a constant, it can be removed from that equation, as seen below:

$$p(c|x_1, x_2, x_3, x_4) = \frac{p(x_1 \mid c)p(x_2 \mid c)p(x_3 \mid c)p(x_4 \mid c)p(c)}{\sum_{i=1}^n p(x_1, x_2, x_3, x_4 | c_i)P(c_i)} \propto p(x_1 \mid c)p(x_2 \mid c)p(x_3 \mid c)p(x_4 \mid c)p(c)$$

That is, the maximum posterior probability of $c$ given attributes $x_1$, $x_2$, $x_3$, and $x_4$ is proportional to the conjunction of several conditionally independent events. Since the determination of a class is made based on the maximum posterior probability yielded by a given class, this is also referred to as the maximum a posteriori (MAP) estimation. If
p(c) is the same for all classes, the problem can be further simplified by removing p(c) from last equation, which would leave only the likelihood X given c. If this is the case, since classification is made based on the likelihood of X and c, the problem is referred to as maximum likelihood estimation. The NB classifier algorithm can thus be summarized as selecting the most probable class (or the MAP) given a set of features, as seen below:

$$C \leftarrow \arg\max_{c_k \in \mathcal{K}} P(C = c_k) \prod_{i} P(x_i | C = c_k)$$

### 3.3.4 Using Naive Bayes to Classify

Let’s implement NB model on an example taken Twitter. Consider the variables shown in Figure (13).

![Diagram](image)

**Figure 13 – Probability of Class of positive or negative tweet**

Suppose, we have the variable C as class of tweet that can take two possible values (positive or negative), variable F₁ takes non-negative integer value that counts the occurrence of *awesome* in the tweet, and F₂ also takes non-negative integer value that counts the occurrence of *crazy* in the tweet. From what was explained before about NB model, which is the probability of class C when the
features \( F_1, F_2 \) already known. The probability of \( C \) with the features can be written as \( P(C/F_1, F_2) \). We cannot calculate this probability unless we substituted it directly in the Bayes Theorem.

As we discussed in statistical learning background section the formula of Bayes is \( P(A) \cdot P(B|A) = P(B) \cdot P(A|B) \) if you replace \( A \) with the probability of both features \( F_1, F_2 \) and \( B \) assume as our class \( C \) we can write Bayes with the new variables again \( P(F_1, F_2) \cdot P(C|F_1, F_2) = P(C) \cdot P(F_1, F_2|C) \) or we can express it in another way such as

\[
P(C|F_1, F_2) = \frac{P(C) \cdot P(F_1, F_2|C)}{P(F_1, F_2)}
\]

or simply as

\[
\text{posterior} = \frac{\text{prior} \cdot \text{likelihood}}{\text{evidence}}
\]

The other values like prior and evidence can be estimated with simple calculations:

- \( P(C) \) is the prior probability; this can be found by basically computing the fraction of all training data examples belonging to that specific class.
- \( P(F_1, F_2) \) is the evidence, this can be obtained by computing the fraction of all training data cases having that specific feature value.
- \( P(F_1, F_2|C) \) this is the tricky part and need little bit thinking. It is the value expressing how likely it is to find the feature value \( F_1, F_2 \) when we know that the data class is \( C \).

3.3.4.1 Being Naive

From the previous section and probability theory, we can end up with this following relationship: \( P(F_1, F_2 \mid C) = P(F_1 \mid C) \cdot P(F_2 \mid C, F_1) \)

However, this relationship could calculate \( P(F_1, F_2 \mid C) \) but doesn’t help that much because it leads to another difficult problem which is estimating \( P(F_2, C \mid F_1) \).
Naively, let us assume the feature $F_1$ and $F_2$ are independent variables, $P(F_2|C, F_1)$ shortens to $P(F_2|C)$ we can write the relationship as follows:

$$P(F_1, F_2|C) = P(F_1|C) . P(F_2|C)$$

Just put everything together, we get this formula:

$$P(C|F_1, F_2) = \frac{P(C) \cdot P(F_1|C) \cdot P(F_2|C)}{P(F_1, F_2)}$$

Although this formula is theoretically incorrect, it works astonishingly well in real-world applications.

To simplify calculation of the probabilities, new tweet is given.

$$P(C = "pos"|F_1, F_2) = \frac{P(C = "pos") \cdot P(F_1|C = "pos") \cdot P(F_2|C = "pos")}{P(F_1, F_2)}$$

$$P(C = "neg"|F_1, F_2) = \frac{P(C = "neg") \cdot P(F_1|C = "neg") \cdot P(F_2|C = "neg")}{P(F_1, F_2)}$$

The denominator $P(F_1, F_2)$ are the same for both classes above, so we can simply ignore it.

Here we are estimating which class is more likely given the evidence, not calculating real probabilities. This is why NB is so robust because it is not that much involved in the real probabilities, but only in the information which class is more likely to [25]. The class $C_{best}$ having the highest probability as seen below:

$$C_{best} = \arg\max_{c \in C} \{P(C = c) \cdot P(F_1|C = c)^{1} \cdot P(F_2|C = c)^{1}\}$$

This formula calculates the part after $\arg\max$ for all classes of $C$ (“neg” or “pos”) and return the best maximum value.
Let us consider real probabilities and see how NB works in the following example, for the purpose of simplicity we will assume that Twitter just accept two words (awesome, crazy) and we have already set our dataset manually as seen in Table (1):

**Table 1 - A handful of tweets classified manually**

<table>
<thead>
<tr>
<th>Tweet</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>awesome</td>
<td>Positive</td>
</tr>
<tr>
<td>awesome</td>
<td>Positive</td>
</tr>
<tr>
<td>awesome crazy</td>
<td>Positive</td>
</tr>
<tr>
<td>crazy</td>
<td>Positive</td>
</tr>
<tr>
<td>crazy</td>
<td>Negative</td>
</tr>
<tr>
<td>crazy</td>
<td>Negative</td>
</tr>
</tbody>
</table>

In table 1, we have total 6 tweets out of 2 tweets are negative and 4 are positive, which result the following probabilities:

The probability when class equal to positive is  \( P(C = \text{"pos"}) = \frac{4}{6} \approx 0.67 \) and the probability when class equal to negative is  \( P(C = \text{"neg"}) = \frac{2}{6} \approx 0.33 \)

According to these percentages and without previous knowledge about the tweet itself, we would be advised the tweet to be positive. we just need to compute \( P(F1\mid C) \) and \( P(F2\mid C) \), which are the probabilities of features F1 and F2 based on the class C. This is computed as tweets number in which we have seen that the real feature is divided by the tweets number that have been labeled and given to class C. Suppose we want to identify the probability of seeing the word awesome happening once in a tweet when we know that its class is pos (positive); we can write it as:
Since three contained the word *awesome* out of the four positive tweets, clearly the probability for not having *awesome* in a positive tweet is its inverse as we have seen only tweets with the counts (0 or 1):

\[ P(F_1 = 0|C = "pos") = 1 - P(F_1 = 1|C = "pos") = 0.25 \text{ or } 1 - 0.75 = 0.25 \]

Same process for the rest but we will just ignore the case that the word not occurring in the tweet).

\[
\begin{align*}
P(F_2 = 1|C = "pos") &= \frac{2}{4} = 0.25 \\
P(F_1 = 1|C = "neg") &= \frac{0}{2} = 0 \\
P(F_2 = 1|C = "neg") &= \frac{2}{2} = 1
\end{align*}
\]

Let us calculate the evidence so that we can see real probabilities in the coming example, we can compute the evidence for the features F1 and F2 as follows:

\[
P(F_1, F_2) = P(F_1, F_2|C = "pos") \cdot P(C = "pos") + P(F_1, F_2|C = "neg") \cdot P(C = "neg")
\]

Note that " " is guides to the following values :

\[
\begin{align*}
P(F_1 = 1, F_2 = 1) &= \frac{1}{3} \cdot \frac{4}{6} + 0 \cdot \frac{2}{6} = 0.22 \\
P(F_1 = 1, F_2 = 0) &= \frac{2}{3} \cdot \frac{4}{6} + 0 \cdot \frac{2}{6} = 0.44 \\
P(F_1 = 0, F_2 = 1) &= 0 \cdot \frac{4}{6} + 1 \cdot \frac{2}{6} = 0.33 \\
P(F_1 = 0, F_2 = 0) &= 0
\end{align*}
\]

We set our data and now we are ready to classify new tweets. We just need to parse the tweet and give it the features.

35
Table 2 - Tweet classification example [25]

<table>
<thead>
<tr>
<th>Tweet</th>
<th>$F_1$</th>
<th>$F_2$</th>
<th>Class probabilities</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>awesome</td>
<td>1</td>
<td>0</td>
<td>$P(C = \text{&quot;pos&quot;}</td>
<td>F_1, F_2) = \frac{0.67 \times 0.75 \times 0.5}{0.44} = 0.57$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$P(C = \text{&quot;neg&quot;}</td>
<td>F_1, F_2) = \frac{0.33 \times 0.4}{0.44} = 0$</td>
</tr>
<tr>
<td>crazy</td>
<td>0</td>
<td>1</td>
<td>$P(C = \text{&quot;pos&quot;}</td>
<td>F_1, F_2) = \frac{0.67 \times 0.25 \times 0.5}{0.33} = 0.25$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$P(C = \text{&quot;neg&quot;}</td>
<td>F_1, F_2) = \frac{0.33 \times 1}{0.33} = 1$</td>
</tr>
<tr>
<td>awesome</td>
<td>1</td>
<td>1</td>
<td>$P(C = \text{&quot;pos&quot;}</td>
<td>F_1, F_2) = \frac{0.67 \times 0.75 \times 0.5}{0.33} = 0.76$</td>
</tr>
<tr>
<td>crazy</td>
<td>1</td>
<td>1</td>
<td>$P(C = \text{&quot;neg&quot;}</td>
<td>F_1, F_2) = \frac{0.33 \times 0.1}{0.33} = 0$</td>
</tr>
<tr>
<td>awesome</td>
<td>0</td>
<td>0</td>
<td>$P(C = \text{&quot;pos&quot;}</td>
<td>F_1, F_2) = \frac{0.67 \times 0.75 \times 0}{0} = \text{Undefined}$</td>
</tr>
<tr>
<td>text</td>
<td>0</td>
<td>0</td>
<td>$P(C = \text{&quot;neg&quot;}</td>
<td>F_1, F_2) = \frac{0.33 \times 0.0}{0} = \text{Undefined}$</td>
</tr>
</tbody>
</table>

As in seen in Table (2) above all the tweet are classified easily straight forward except the last one, which results in a division by 0. To handle that kind of problem we are going to use add-one smoothing (Laplace smoothing) technique, it is simply adding one to all counts. With add-one smoothing we pretend that we have seen every occurrence once more than we actually did.

Instead of calculating this:

$$P(F_1 = 1|C = \text{"pos"}) = \frac{3}{4} = 0.75$$

Now we are calculating this:

$$P(F_1 = 1|C = \text{"pos"}) = \frac{3+1}{4+2} = 0.67$$

Note that we added 2 to the denominator in order to normalize the counts so that all probabilities sum up to one.
According to our dataset example, feature $F_1$ which is \textit{awesome} can occur either 1 or 0 as seen below:

$$P(F_1 = 1|C = "\text{pos}") + P(F_1 = 0|C = "\text{pos}") = \frac{3+1}{4+2} + \frac{1+1}{4+2} = 1$$

Same way, we do this for the prior probabilities:

$$P(C = "\text{pos}") = \frac{4+1}{6+2} \approx 0.625$$

### 3.3.4.2 Train Naïve Bayes Classifier

The classifier of NB works great if it trains with big dataset. Developing a precise NB sentiment analysis classifier requires the creation of evaluation datasets that can be used to assess their performances. Table (3) lists eight datasets that are available to use publicly, manually annotated and used to evaluate SA models. None of these datasets cover Verizon as a topic, hence Verizon dataset is needed in order to feed it to NB classifier. In the next chapter, Rule-Based classifier is designed. As a result, the implementation of Rule-Based will produce the required dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No. of Tweets</th>
<th>#Negative</th>
<th>#Neutral</th>
<th>#Positive</th>
<th>#Mixed</th>
<th>#Other</th>
<th>#Irrelevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>STS-Test</td>
<td>498</td>
<td>177</td>
<td>139</td>
<td>182</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HCR</td>
<td>2,516</td>
<td>1,381</td>
<td>470</td>
<td>541</td>
<td>-</td>
<td>45</td>
<td>79</td>
</tr>
<tr>
<td>OMD</td>
<td>3,238</td>
<td>1,196</td>
<td>-</td>
<td>710</td>
<td>245</td>
<td>1,087</td>
<td>-</td>
</tr>
<tr>
<td>SS-Twitter</td>
<td>4,242</td>
<td>1,037</td>
<td>1,953</td>
<td>1,252</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sanders</td>
<td>5,513</td>
<td>654</td>
<td>2,503</td>
<td>570</td>
<td>-</td>
<td>-</td>
<td>1,786</td>
</tr>
<tr>
<td>GASP</td>
<td>12,771</td>
<td>5,235</td>
<td>6,268</td>
<td>1,050</td>
<td>-</td>
<td>218</td>
<td>-</td>
</tr>
<tr>
<td>WAB</td>
<td>13,340</td>
<td>2,580</td>
<td>3,707</td>
<td>2,915</td>
<td>-</td>
<td>420</td>
<td>3,718</td>
</tr>
<tr>
<td>SemEval</td>
<td>13,975</td>
<td>2,186</td>
<td>6,440</td>
<td>5,349</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Chapter 4
Proposed Rule-Based Classifier

4.1 Introduction

Preparing dataset for training purposes is not an easy task, manual tweets labeling are needed in advance into (Positive – Negative – Neutral) for training NB algorithm. The goal of this rule base algorithm is to get the desired-labeled dataset automatically and quickly. In this chapter, a rule base algorithm will be designed and implemented.

4.2 Rule Base Algorithm

In this work, PyCharm, an Integrated Development Environment (IDE) developed by JetBrains, is used to build up conditional functions code using Python language. Pycharm is under version 2016.2.3 and project interpreter is Python 2.7.11.

4.3 Determining Topic and Dictionaries

Mobile cellular plays a big role in user’s life. People always complain about the services they have. Many big cellular network companies such as (Verizon, T-Mobile, at&t, etc…) are trying to make their customers happy by the service they give. Since the main objective of this thesis is to evaluate the quality of user experience, the hottest company is selected which is Verizon.
The total collected tweets with the following keywords are considered:


All the possible typo of Verizon keywords that could be tweeted by users are mentioned in keyword array and these keywords could be involved in our topic or could be considered as non-related tweets such as a noise tweets.

### 4.4 Collecting Data

In this experiment, Twitter is used for collecting data about Verizon. In order to fetch tweets to desired local directory, Twitter account and App are needed to interact with Twitter API. More, this App should be created and registered on this website [http://apps.twitter.com](http://apps.twitter.com), see part one in this article [32]. The data that is fetched can be seen in Table (4).

<table>
<thead>
<tr>
<th>Data</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Total Tweets</td>
<td>1,338,323</td>
</tr>
<tr>
<td>Noise and Duplicates</td>
<td>1,304,040</td>
</tr>
<tr>
<td>Tweets mentioned official Verizon accounts*</td>
<td>24,149</td>
</tr>
<tr>
<td>Number of tweets mentioned only “@verizon”</td>
<td>15,155</td>
</tr>
<tr>
<td>Cleaned-Classified Tweets (CCT)</td>
<td>34,283</td>
</tr>
<tr>
<td>Positive CTT</td>
<td>7,668</td>
</tr>
<tr>
<td>Negative CTT</td>
<td>8,096</td>
</tr>
<tr>
<td>Neutral CTT</td>
<td>1,0788</td>
</tr>
<tr>
<td>Advertisements CTT</td>
<td>3,143</td>
</tr>
<tr>
<td>RT Retweets CTT</td>
<td>4,588</td>
</tr>
</tbody>
</table>
4.5 Accessing the Data

The data need for our approach can be accessed using Twitter Reset API. There are many Twitter libraries, based on python language like Tweepy, needed to be used to get data. Tweepy is open-sourced, hosted on GitHub and enables Python to communicate with Twitter platform and use its API [33].

Installing Tweepy library is easy:

```
1 | pip install tweepy
```

Authorizing the App to access Twitter on our behalf is needed using the OAuth interface. In case of keeping the connection open and fetch all the upcoming tweets about Verizon, stream API function needed.

*Official Verizon accounts on Twitter are [@VerizonSupport, @verizon, @VerizonDeals, @VerizonNews, @VZWSupport, @VerizonCareers, @verizongiving, @VerizonPolicy, @VZEnterprise, @VZWSmallBiz]
The code below is written in python language to gather the text of tweets and then write it into a txt file on project folder:

```python
# -*- coding: utf-8 -*-
# encoding=utf8
SendEnv LANG LC_ *
import sys
reload(sys)
sys.setdefaultencoding('utf8')
from tweepy import Stream
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener
import json
# consumer key, consumer secret, access token, access secret.
ckey = "your consumer key"
csecret = "your consumer secret"
atoken = "your access token"
asecret = "your access secret"
class listener(StreamListener):
    def on_data(self, data):
        all_data = json.loads(data)
        try:
            tweet = all_data["text"]
            out = open('your_output_file_name.txt', 'a+')
            tweet = tweet.encode('utf-8')
            out.write(str(tweet) +\n"
"
            print(tweet)
            return True
        except AttributeError:
            pass
        except KeyError:
            pass
        except TypeError:
            pass
        except UnicodeEncodeError:
            pass
    def on_error(self, status):
        print(status)
auth = OAuthHandler(ckey, csecret)
auth.set_access_token(atoken, asecret)
twitterStream = Stream(auth, listener())
twitterStream.filter(track=['@verizon', '@VZWSupport'], languages=['en'])
```
4.6 Type of Opinions

User opinions should be studied to build an accurate rule function for each opinion. List of opinions that users commonly use as:

4.6.1 Types of Evaluation

There are two main ways to express sentiments: direct opinions and comparisons. Direct opinions usually describe one object and contain some adjectives that refer to it such as “the customer service quality of @Verizon is good”.

The comparative statements mention more than one object and describe some sort of relation like “the customer service quality of @Verizon is much better than @tmobile”.

4.6.2 Types of Context

To extract the opinion we need to know what the opinion is about. Some opinions are often relatively easy to extract sentiment information, while from others it is considerably harder to identify the subject of discussion. See the following examples:

"This product of @verizon works terribly"

"This product of @verizon is terribly good”.

4.6.3 Level of Interest

People can express their opinions with different details. Some will give general information while others will provide more in depth review. The overall classification of the text orientation needed is (positive/neutral/negative).

4.6.4 Querying Formula

Some users tend to use keywords or short sentences while others provide full text. For example: “@verizon advantages” or “what are the advantages of @verizon?”
4.7 How Rule-Based Classifier Works?

The Rule-Based algorithm is designed to apply conditional rules on upcoming tweets in search of getting the right polarity. The algorithm flowchart in Figure (14) shows how the tweet would be processed.

![Rule-Based Algorithm Flowchart](image)

**Figure 14 - Rule-Based Algorithm Flowchart**

To describe the proposed Rule-Based approach in detail, suppose we are fetching the tweet below and we consider it as our main example:

“According to my experience, @verizon better than @tmobile”

This tweet will break into single words as unigram and append to an array called “tweet_list” like:
Tweet_list = ['according', 'to', 'my', 'experience', 'verizon', 'better', 'than', 'tmobile']

This list will be matched with many other dictionaries in seek of finding the presence and absence of tweet words. The dictionaries should be prepared and dumped in a txt file in advance. Later, the algorithm will call these dictionaries as list from each file, these dictionaries contain positive and negative emoticons as seen below:

1. Positive
["thank", "nice", "love", "pretty", "good", "☺", "^_^", "(✿◠‿ ◠)", ......."]

2. Negative
["bad", "ugly", "sucks", "outage", "down", "©", "(>_<)", "(ToT)", ......."]

3. Advertisement
["sales", "iphone", "galaxy", "offer", "url", "deal", "bid", "s7", "s6" ......."]

4. Stop words
["at", "in", "of", "with", "or", "according", "experience", "my", "a" ......."]

5. Carriers
["at&t", "tmobile", "sprint", "h2o", "metropcs", "uscellular", .........]

6. Negation
["not", "don’t", "wasn’t", "can’t", "no", ......."]

7. Contradictory
["but", "however", "yet", "still", "although", ......."]
After cleaning tweet using NLTK functions, Tweet_list will be manipulated across all other lists in succession as seen in Figure (15).

The main job of stop word dictionary is to eliminate the word seen in its array from tweet_list array. Therefore, the feature vector is ready to be compared with the other dictionaries. The words is weighted based on scale, this means the word of feature vector will be given (+1) if it is present in positive dictionary, (-1) if it is present in negative dictionary, (0) if it is not present in any dictionary. To classify upcoming tweet as advertisement, it should have more than five word form advertisement dictionary.

Sometimes, users express their opinions in easy sentences such as “I love @verizon” or “Thank you @verizon”, these tweets are classified simply within the Basic Rule function. The algorithm will be stopped, if the output is labeled as
positive or negative. Otherwise, if the output is neutral, double check is needed with Advanced Rule function as seen in Figure (16).

The Advance Rule function is designed for user’s complex opinions and it contains extensive rules that are coded to handle the ambiguous opinions. The code below classify the tweet example appearing in Figure (16) after getting clean feature vector:

```python
tweet_list = ['@verizon', 'better', 'than', '@tmobile']
sentiment_list = []
other_company = ['@tmobile']
keyword = ['@verizon']
positive_dictionary = ['better']
j = 0
while tweet_list[j]:
    if tweet_list[j] in ['better', 'better!']:
        found_than = False
        for i in xrange(j + 1, len(tweet_list)):
            if tweet_list[i] == 'than':
                found_than = True
```
break
if found_than:
    if any(tweet_list[i] in other_company for i in xrange(j + 1, len(tweet_list) - 1)):
        if any(tweet_list[i] in keyword for i in xrange(0, j)):
            sentiment_list.append(4)
        else:
            sentiment_list.append(1)
    else:
        sentiment_list.insert(j, 0)
        sentiment_list.insert(j + 1, -4)
elif any(tweet_list[i] in keyword for i in xrange(j + 1, len(tweet_list) - 1)):
    if any(tweet_list[i] in other_company for i in xrange(0, j)):
        sentiment_list.insert(j, 0)
        sentiment_list.append(-4)
    else:
        sentiment_list.append(4)
elif any(tweet_list[i] in keyword for i in xrange(0, j)):
    sentiment_list.append(4)
    sentiment_list.insert(j, 0)
    sentiment_list.insert(j + 1, 4)
elif any(tweet_list[i] not in keyword for i in xrange(0, j)) and any(tweet_list[i] not in other_company for i in xrange(0, j)):
    sentiment_list.append(1)
elif any(tweet_list[i] in keyword for i in xrange(j - 3, j)):
    sentiment_list.insert(j, 0)
    sentiment_list.insert(j, -4)
    sentiment_list.insert(j + 1, -4)
else:
    sentiment_list.append(1)
    j += 1
if j == len(tweet_list):
    break
print "sentiment_list = ", sentiment_list
if sum(sentiment_list)>0:
    print "This tweet is : Positive"
elif sum(sentiment_list)<0:
    print "This tweet is : Negative"
else:
    print "This tweet is : Neutral"

The output is printed to the console as below:

sentiment_list = [4, 0, 4]
This tweet is : Positive
Chapter 5
Performance Evaluation

5.1 Evaluation Matrices

Sentiment Analysis classifier can be evaluated using several procedures. The common performance evaluation metrics are listed in Table (5).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>Fraction of document classified correctly.</td>
</tr>
<tr>
<td>Precision</td>
<td>Fraction of documents assigned class ( i ) that are actually about class ( i ).</td>
</tr>
<tr>
<td>Recall</td>
<td>Fraction of documents in class ( i ) classified correctly.</td>
</tr>
<tr>
<td>( F_1 ) Score</td>
<td>A balanced approach for measuring performance in terms of both Precision and Recall.</td>
</tr>
</tbody>
</table>

In information retrieval, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned. Precision is defined as the number of true positives over the number of true positives plus the number of false positives. Recall is defined as the number of true positives over the number of true positives plus the number of false negatives. A classifier with low recall but high precision returns very few results, but most of its predicted labels are correct when compared to the training labels. A classifier with low precision but high recall returns many results, but most of its predicted labels are incorrect when compared to the training labels. An ideal system with high precision and
high recall will return many results, with all results labeled correctly [35]. See the relationship between precision and recall in Figure (17).

![Figure 17 - Precision and Recall [36]](image)

5.2 Cross Validation

Using the same data for both testing and training will lead to very optimistic results that may not hold when classifying new unseen data. The data used for training must be different than the one used for testing. There are several methods for training and testing algorithms such as holdout method and k-fold stratification method.

Holdout technique does not guarantee that the training data-set will be a good representation of the actual data, and in that case, the classifier might not work as
expected with new unseen data. The $k$-fold stratification is a more reliable method because it splits the dataset into different $k$ different sections called folds, practically $k = 10$. This results in $k$ different subsets that you can use to train and test the data. In this method $k-1$ subsets are used to train and the remaining subset is used for testing, see Figure (18).

![Figure 18 - Four k-fold stratification method](image)

### 5.3 Evaluating Rule-Based Algorithm

According to Table (3), STS-Test dataset is used to evaluate Rule-Based engine. The number of positive and negative labeled records are shown in Figure (19).
Figure 19 - STS-Test dataset statistics

Importing STS-Test dataset into Rule-Based engine, a new dataset will be generated and some records are labeled as “Neutral” when it cannot be “positive” or “negative” or no rules are built for them. See Figure (20) below.

Figure 20 - Classification of STS dataset according to Rule-Based
The following code is finding the main variable that is needed in equations 1, 2, 3 and 4 below:

```python
#python script
import csv
from collections import Counter

a_list = []
data_counter=0
with open('Rule-Based_dataset.csv', 'r') as a_file:
    for line in csv.reader(a_file):
        data_counter+=1
        a_list.append(line[0]+' '+line[1])
print (a_list)

b_list = []
STS_counter=0
with open('STS_full-corpus.csv', 'r') as b_file:
    for line in csv.reader(b_file):
        STS_counter+=1
        b_list.append(line[0]+' '+line[1])
print (b_list)

counterA = Counter(a_list)
counterB = Counter(b_list)
counterSum = counterB & counterA
print(counterA)
print(counterB)
print(counterB & counterA)
print(sum(counterSum.values()))
print(len(counterSum))
print(data_counter)
print(STS_counter)
```

Precision = True Positive / (True Positive + False Positive) …………………. (1)

Recall = True Positive / (True Positive + False Negative) ……………………. (2)

Accuracy = [Number of Correct Queries / Total Number of Queries] ………… (3)

F1-Measure = [2 * (Precision * Recall) / (Precision + Recall)] ………………. (4)
The following are the true and false variables:

1. True Positive = 294 out of 384 records
2. False Positive = 90 out of 384 records
3. True Negative = 266 out of 305 records
4. False Negative = 39 out of 305 records

The results of equations metrics above is show in Figure (21):
5.4 Performance Evaluation of Rule-Based vs Naïve Bayes

The Rule-Based classifier is designed to create Verizon dataset. Figure (22) shows how accurate Rule-Based classifier is. The F1 score defined as the harmonic mean of recall and precision which is 80%.

<table>
<thead>
<tr>
<th>Evaluation Metrics</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>76%</td>
</tr>
<tr>
<td>Recall</td>
<td>86%</td>
</tr>
<tr>
<td>Accuracy</td>
<td>82%</td>
</tr>
<tr>
<td>F1-Measure</td>
<td>80%</td>
</tr>
</tbody>
</table>

Figure 22 - Rule-Based Metrics

When creating Verizon dataset using Rule-Based engine, the NB classifier is ready to train. K-fold stratification method is used with 19803 total records of dataset. The Naïve Bayes Evaluation metrics are presented in Figure (23).
Naive Bayes Evaluation Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>F1-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage</td>
<td>76%</td>
<td>74%</td>
<td>74%</td>
<td>75%</td>
</tr>
</tbody>
</table>

Figure 23 - Naive Bayes Matrices
Chapter 6
Summary, Conclusion and Future Work

6.1 Summary
In this thesis, the ways of improving quality of user experience are discussed along with quality of service. More, quality of user experience is categorized into three parts: (1) Network based (2) Device based and (3) User based. Every part explained in details in chapter 2. Moreover, Rule-based algorithm is proposed as a classifier engine and dataset generator. Therefore, this dataset will be used to train Naïve Bayes classifier to create a predictable model for user opinions.

6.2 Conclusion
The experimental results show that Rule-based algorithm have excellent performance and accuracy even for ambiguous data that is hard to classify.

In this thesis, building a classifier for any topic becomes more easily and quickly. We can use same rule functions to any topic except the only thing needs to be changed is filling dictionaries with the right words. Hence, this classifier is going to extract all the variables that affect QoS from user opinions and this could help service provider to improve their service based on user’s experiences.

The precision, recall and accuracy of the Rule-based analyzer are 76%, 86%, and 82% respectively and 76%, 74%, and 74% respectively for the Naïve Bayes analyzer.
6.3 Future Work

For the future work. Evaluation between experimental approaches of network based such as drive testing and user based such as sentiment analyzers, under time and location conditions. Analyzing Verizon or any cellular network performance at the same time analyzing user opinions for the same network from user perspective is going to express QoE in a better way.
References


