A Comparative Study of Twitter Sentiment Analysis Methods for Live Applications

Angel Cambero
ac5444@rit.edu

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation
A Comparative Study of Twitter Sentiment Analysis Methods for Live Applications

by

Angel Cambero

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

Thesis Advisor
Dr. Joe Geigel

Department of Computer Science

B. Thomas Golisano College of Computing and Information Sciences

Rochester Institute of Technology
Rochester, New York

August 2016
The thesis “A Comparative Study of Twitter Sentiment Analysis Methods for Live Applications” by Angel Cambero has been examined and approved by the following Examination Committee:

________________________________
Dr. Joe Geigel
Thesis Committee Chair

________________________________
Dr. Reynold Bailey
Reader

________________________________
Dr. Leon Reznik
Observer
Dedicated to Aimena and Rodatam
Acknowledgements

Firstly, I would like to express my deepest thanks to Dr. Joe Geigel, my thesis advisor, for his support and guidance this past year and during my Master’s studies at RIT. His patience, inspiration, and passion helped me immensely throughout this process and I could not have asked for a better advisor to help me complete my thesis.

I would also like to thank the rest of my thesis committee, Dr. Reynold Bailey and Dr. Leon Reznik, as well as Dr. Hans-Peter Bischof, for their encouragement. A huge thanks to my advisors Rebecca O’Connor and Cindy Wolfer, for helping me acquire relevant information and services available to me, and helping me make responsible decisions consistent with my interests, goals, abilities, and degree requirements.

My sincere thanks also goes to the whole Social Product Development & Experiences Team at Intuit for offering me internships and jobs opportunities, and especially to Joy Caro for guiding me, allowing me to work on diverse, exciting projects, and helping me to reach my educational and career goals.

I would like to thank the Ministry of Higher Education, Science and Technology (MESCyT) of the Dominican Republic for the financial support for my graduate studies, and the great support I received from Professor Pedro Ureña, Professor Jose Solís, Sam Little and Gabriela Henriquez during the application process.

I would also like to thank all my friends from Instituto Evangelico, IPISA and PUCMM for their encouragement and my fellow classmates from RIT for the late nights we stayed up working together to make deadlines, and for all the fun we had. In particular, I would like to thank Marisa Palacio for believing in me and standing by me.

Last but not least, for my family’s continued confidence and belief in me, I thank them.
Abstract

A Comparative Study of Twitter Sentiment Analysis Methods for Live Applications

Angel Cambero

Supervising Professor: Dr. Joe Geigel

Sentiment Analysis is growing exponentially due to the importance of the automation in mining, extracting and processing information in order to determine the general opinion of a person. The problem that this thesis proposes to address is to determine what methods are more suitable to extract subjective impressions in real time from Twitter. For live applications, since the opinions collected from Twitter are limited to certain amount of characters and it will happen in a real-time environment, this provides an interesting scenario; we will test using both the Machine Learning Approach and the Lexicon-based Approach, and then combine them in an effort to increase the accuracy. In order to test the real-time factor, I will implement a web service with the purpose of collecting real-time feedback from Twitter in real-time, which will be later processed and analyzed for accuracy and realtime performance.
# Table of Contents

Dedication .............................................................................................................................. iii

Acknowledgements ................................................................................................................ iv

Abstract .................................................................................................................................. v

1 Background ........................................................................................................................ 1

1.1 Introduction ...................................................................................................................... 1

1.2 What is Sentiment? .......................................................................................................... 1

1.2.1 Sentiment Analysis ..................................................................................................... 2

1.2.2 Sentiment Analysis on Twitter .................................................................................... 3

1.3 Sentiment Analysis Classification ................................................................................... 3

1.3.1 Sentiment Analysis Classification: Levels .................................................................. 4

1.3.2 Sentiment Analysis Classification: Techniques .......................................................... 5

1.4 Hypothesis ....................................................................................................................... 9

1.5 Roadmap ......................................................................................................................... 10

2 Framework Design ........................................................................................................... 10

2.1 Stage 1 – Building Webservice with Real-time Twitter API Searching Capabilities Using an External Sentiment Analysis Library ........................................................................ 11

2.1.1 Subjects .................................................................................................................... 12

2.1.2 Materials .................................................................................................................. 12

2.1.3 Data Collection ........................................................................................................ 12

2.1.4 Experimental Procedure ........................................................................................ 13

2.2 Stage 2 – Integrating Sentiment Analysis Framework with an External System .......... 14

2.2.1 Subjects .................................................................................................................... 14

2.2.2 Materials .................................................................................................................. 15

2.2.3 Data Collection ........................................................................................................ 15

2.2.4 Experimental Procedure ........................................................................................ 15

3 System Implementation ..................................................................................................... 17

3.1 Approaches to Building Lexicon-Based Method ............................................................ 17

3.2 Approaches to Building Learning-Based Method .......................................................... 19

3.3 Technologies Used ........................................................................................................ 21

4 Analysis .............................................................................................................................. 22

4.1 Combining Lexicon-Based and Learning-Based Methods ............................................. 22

4.2 Data ................................................................................................................................ 26

5 Conclusions ....................................................................................................................... 28

5.1 Summary of Conclusions ............................................................................................ 28

5.2 Future Directions .......................................................................................................... 29

References ............................................................................................................................. 30

User Manual ........................................................................................................................... 34
List of Figures

Figure 1. Sentiment Classification Techniques ................................................................. 4
Figure 2. Overview of Stage 1: Architecture for the Test in the Exhibition ......................... 11
Figure 3. Graphic Representing an Introverted Personality using “Projections .................. 13
Figure 4. Overview of Stage 2: Architecture for the Integration with MarketersPro .......... 14
Figure 5. Graphic Representing a Real-Time Search on MarketersPro with ....................... 16
Figure 6. Lexicon-Based Approach based on AFINN-111 .................................................. 17
Figure 7. Lexicon-Based Approach based on AFINN-111 .................................................. 18
Figure 8. Lexicon-Based Approach based on SentiWordnet ............................................. 19
Figure 9. Lexicon-Based Approach using VADER Sentiment Lexicon .............................. 19
Figure 10. Learning-Based Approach Architecture ............................................................ 20
Figure 11. One of the Initial Architectural Overview Variations: SA Methods .................. 23
Figure 12. One of the Initial Architectural Overview Variations: SA Methods order .......... 24
Figure 13. Final Architectural Overview: SA Methods order changes dynamically .......... 25
Figure 14. Graph Depicting the Percent of Tweets in Each Dataset .................................. 26
Figure 15. Format Used to Organize Tweets in the Created Datasets ................................. 27
Figure 16. Dataset Class Distribution Representation in Percent ...................................... 28
1 Background

1.1 Introduction
This thesis intends to test whether the combination of SA methods could enhance the accuracy of the analysis in a real-time environment. Processing and analyzing existent opinionated data using Sentiment Analysis is an attractive field of research, especially because of the amount of free and available existent information that could be extracted. We are going to test with two different approaches, the Lexicon-based and the Machine Learning approach. For the former will be used a Dictionary-based approach and a Naïve Bayes Classifier for the latter. The main three experiments are to test the integration of the Twitter API with an external Sentiment Analysis tool (see 3.2), to test the accuracy of my Sentiment Analysis implementation (see 3.1) and to test my Sentiment Analysis method in a real-time scenario (see 3.2).

Once completed, the contributions of the thesis include the comparison and combination of different methods in an attempt to increase the accuracy. This thesis also contributes with the code itself, which will be released under an open source license. The first experiment will be used in ImagineRIT and the latter one, if successful, will be integrated in an Intuit’s product (see 4.3).

These experiments are relevant to the computer science community because Sentiment Analysis is a field growing expeditiously, and we will contribute with a study related with the Hybrid approach, which represents only the 6.97% of the different methods articles by 2013 [6]. Although, in this study we focused in Twitter and in two specific methods, future experiments can be done to other social networks and combining other methods.

1.2 What is Sentiment?
To understand sentiment analysis, we first have to define what is sentiment itself. Feelings can be confusing, distracting, irritating, or frustrating unless you understand
what they are about [8]. Generally speaking, sentiments are the same as feelings, and they are focused mainly on opinions and attitudes rather than facts; this is why sentiments are really subjective. However, some sources refer to “feeling” as a natural response to attraction, pleasure, pain or repulsion [37] while sentiment implies an emotion motivated by a perception or opinion [37].

There are a multitude of emotions which psychologists have attempted to organize into six different categories: love, joy, surprise, anger, sadness and fear [38]. Emotions like joy and sadness are commonsense emotions that we experience every day in different degrees [39] and these two are what we are going to target throughout the next sections. Sentiment Analysis is mainly concerned with detecting positive or negative opinions [6]. This helps to avoid more complexity since it is also highly impractical to consider more emotions considering the size of the datasets [40].

The greatest significance of sentiments is that nobody has to try to have feelings; this is an ingrained part of us as human beings. This means that every person is a potential opinion generator for the sentiment analysis tools.

1.2.1 Sentiment Analysis

Sentiment analysis is basically the automation of the analysis of a given text in order to determine the feelings conveyed in it. Sentiment analysis and opinion mining have become known as interchangeable terms.

Sentiment analysis is defined by Subhabrata Mukherjee [9] as a “Natural Language Processing and Information Extraction task that aims to obtain writer’s feelings expressed in positive or negative comments, questions and requests, by analyzing a large numbers of documents”. In other words, sentiment analysis intends to define the feelings of the writer regarding a particular topic based on the writer’s opinion.

Sentiment analysis is important as it can help to provide insight into different fields. Even when sentiment analysis is not perfect, because the sentiment itself is really
subjective, there is no doubt that processing and analyzing existent opinionated data has only just begun. Even when social media monitoring tools such as Trackur and Mention claim that their Sentiment Analysis accuracy is over 70 percent [19], most of the information found, such as [10], claim that anyone who says that they are getting more than 70 percent accuracy is lying. This is in agreement with [15], which states that human raters typically only agree 79 percent of the time, making this really difficult to automate systems to achieve high accuracy.

1.2.2 Sentiment Analysis on Twitter

Sentiment Analysis is a field that is growing fairly rapidly. 81 percent of Internet users (or 60 percent of Americans) have done online research on a product at least once [5], meaning every year there are more articles targeting different text domains over years, where the reviews represent around the 49.12% of the articles [6]. One would not always want to apply sentiment to product reviews; there are too many other fields. One good example of this that has been experimented [7] is the comparison of Twitter sentiment versus Gallup polls of consumer confidence. The results yielded were positive and the correlation was 0.804, inferring that we can use Twitter to measure public opinion. This is precisely what we are going to use Twitter for during this experiment: to extract opinions from it and determinate the tweets’ polarity in real-time.

1.3 Sentiment Analysis Classification

Depending on the perspective, Sentiment Analysis can be classified in different ways; one of them is based on the sentiment classification technique used. This type of classification is divided in two different approaches (see Figure 1): the machine learning approach and lexicon-based approach. We can also include a third classification [13], the hybrid approach.

Sentiment Analysis can also be classified according to the way the opinions to be analyzed are identified. The three main classification levels are the sentence,
The main three levels are the document level, aspect level and the sentence level [23]. The classification depends on the different levels of analysis.

The document level is known as document-level sentiment classification because the main task is to determine if the document as a whole opinion has a negative or a positive sentiment [24]. In other words, for a given a text it would be assumed that the whole text expresses an overall positive or negative opinion about a single entity. Since this method assumed there is only one entity, this method is not the most suitable one
for texts with entities comparison or evaluating more than one entity.

The other two classifications are the sentence level and the aspect level. The sentence level is very similar to the document level, but with the main difference that in this case each sentence is analyzed individually to see if it expresses a negative, neutral or positive opinion. This level adds more flexibility than the document level because it is able to distinguish the objective sentences from the subjective sentences, and this can be used as a first filter [24]. However, we have to mention that there are objective sentences expressing opinion and subjective sentences not transmitting any sentiment.

The most fine-grained analysis is the aspect level, previously known as the feature level. Unlike the sentence and document levels, the aspect level discovers what each opinion is about [23]. The main difference is that this analysis finds a target for each opinion, instead of focusing on language units, like sentences, documents or paragraphs. The goal of this level is to identify the opinion or sentiment on entities and their different aspects. The majority of real-time sentiment analysis systems are based on this level [23].

1.3.2 Sentiment Analysis Classification: Techniques

In the Sentiment Analysis field, the sentiment classification technique is the most researched topic [5]. The goal of this task is to classify, positively or negatively, what an opinion document expresses. Sentiment Classification is mainly divided into two different approaches: the machine learning approach and lexicon-based approach [13]. The Lexicon-based approach uses a collection of positive and negative sentiment terms and can be divided into corpus-based and dictionary-based approach. Alternatively, the Machine Learning approach uses machine-learning algorithms, and Sentiment Analysis is solved in the same way as any other regular text classification problem. The Machine learning classifiers are divided into supervised learning and unsupervised learning. In the next two sub-sections we will expand on these two approaches.
1.3.2.1 Machine Learning Approach

The Machine Learning approach relies on treating the Sentiment Analysis as a text classification problem. Text classification is normally used to automate a business decision that requires processing text [25]. It uses a set of training records to train a model that is used later on to predict new records without a label. Each record is labeled to a class. When a new unlabeled record is given, the model is used to predict its label class [14]. These classes are positive, negative and neutral, however, most of the time the research papers do not refer to the neutral class. In the Machine Learning Approach we can distinguish two different sub-approaches based on the learning method used, the supervised and the unsupervised learning.

The supervised learning uses a supervised classifier, which learns from labeled training documents. The labeled training documents have topic-related words known as key features. The opinion words express a negative or a positive opinion. We are going to briefly mention the sub-classifications of the Supervised Learning method: Decision Tree, Linear, Rule Based and Probabilistic Classifier.

Decision Trees are used for prediction; they can easily be used for classification. Given a record with an unknown class label, this record is tested against the decision tree, and the path is traced from the root to the node that then determines the class prediction for the record [26]. These are popular because its construction does not require settings or any domain expertise [26]. ID3 and C5 are widely used packages for decision tree implementations in text classification problems [32].

Linear classifiers are known because of their simplicity. The main idea is to count the amount of positive and negative words in a sentence and compare the number of positive and negative words to determine the sentence's polarity. The lineal classifier adds weight to all of the words; the “most negative” words have the lowest weight and the “most positive” words have the highest weight. The most popular linear classifier is the Support Vector Machines Classifiers (SVM), whose main principle is to find the linear separator with the best separation between the classes [6].
The Rule Based Classifier is similar to the decision tree classifier because both encode rules on the feature space. The only difference is that the decision tree classifier uses the hierarchical approach [6], while the rule-based classifier allows for overlap in the decision space [32]. Multiple studies [30] [14] show different ways to convert from decision tree classifier to a rule-based classifier. In the rule-based classifier, the training phase generates the rules based on different criteria; the two most popular are support and confidence [33].

The last but not least Supervised Learning Method sub-classification is the Probabilistic Classifier, also known as the generative classifiers because they generate a model of each class [34]. They use mixture models, which assumes that each class is a component of the model [6]. The most famous probabilistic classifier and one of the most used is the Naïve Bayes Classifier, which is one of the simplest classifiers to be coded in every programming language because of the simple mathematics involved [35]. The model works with a bag-of-words, which is an unordered set of words. The frequency of each word is kept, but not the position. Given a document D, the Naïve Bayes returns the class with the maximum probability out of all the classes [34]. This classifier uses the Bayes Theorem to determine the label that a given feature set belongs to [6].

On the contrary, the Unsupervised Learning Method is used mainly when creating the class-labeled training documents is too difficult. In that case, we would collect the unlabeled documents and implement the unsupervised learning. This approach is widely used for document clustering analysis because it does not rely on predefined class-labeled training documents.

One way to summarize is to note that the unsupervised learning learns by observation and in the supervised learning it learns by examples [30]. In other words, in the supervised learning during the training phase, examples are presented to the model, but in the unsupervised model the learner is never taught with solutions. Olivier [36] makes the analogy that this method is more natural because it is in the same way that kids and animals learn.
1.3.2.2 Lexicon-Based Approach

Lexicon-based method is another unsupervised approach, but in this case it could use a dictionary with antonyms and synonyms of opinionated words and phrases with their respective sentiment orientation. The two automated approaches more commonly used to collect the sentiment word list are the dictionary-based and the corpus-based.

The Dictionary-Based approach has a main strategy to manually collect a small set of opinion words and then grow this set by searching in large collection of texts such as WordNet [22][38]. The new words are then added to the first set of opinion words and the cycle is repeated until there are no more words remaining to be found. The biggest downside of this method is that it relies completely on corpora and we will not always have a large collection of opinion words with a domain available. It is important to mention that not all of the words in a lexicon express a positive or negative opinion regarding an entity [14]. The Corpus-Based approach is mainly used in two situations: to discover new sentiment words from a domain corpus using a given list of known opinion words and to create a sentiment lexicon from another one [14]. This approach by itself is not as effective as the dictionary-based approach because it would need a corpus with all the English words [6]. The corpus-based is divided in the statically and the semantic approach depending on the technique used.

1.3.2.3 Hybrid Approach

There are studies that use both approaches, like one presented by Hermida et al. [42], which is a method based on Emotinet. They proved that the method presented was effective to identify emotion from a text with or without less affect-related words in it. To achieve this goal they also used the Support Vector Machine Algorithm, which as we mentioned in the section 2.2.2.1 is the most popular linear, whose main principle is to find the linear separator with the best separation between the classes [6]. Our approach does not rely on an emotion corpus to store commonsense knowledge.

HP Laboratories [41] presented a similar idea, which uses both Machine Learning and
Lexicon-based approach, first using lexicon-based and then a chi-square test to identify new tweets. Our work uses a lexicon-based approach mixed with a rule-based approach, and it uses Bayes Naïve Classifier as the learning algorithm. There are other similar approaches, but using manually labeled sets, in other words, supervised learning. The approach presented by Barbosa et al [43] classifies in subjective and objective then discarded the objective tweets and classified the first group as positive or negative.

1.4 Hypothesis

The aim for this study is to propose a technique to achieve high sentiment analysis accuracy for tweets in a real-time environment. In order to do this, we are proposing to combine two different approaches: the Lexicon-Based approach and the Machine Learning approach. We are going to compare the methods from each approach and then we will integrate them in an effort to achieve our goal. To measure the overall accuracy, we will divide the number of correct classifications by the total number of classifications. For the Machine Learning Approach, we will focus on the training datasets since those are vital to building an effective predictive model. The datasets were split into two sets: the training set and the test set. The training set is used to train the NB Classifier, and the test set is used to measure the performance of the predictive model. We will create our own datasets and will use existent ones as well. For the Lexicon-Based approach, we will focus on the performance of different possible implementations, taking into consideration that Lexicon is a system designed specifically for Twitter.

The accuracy of each of these methods will be measured as well as the accuracy for the combination of these methods. For the Lexicon-Based Method, a lexical database for the English language such as WordNet will be used [22][38]. In Figure 1, the methods selected from each approach are highlighted in green: the probabilistic classifier Naïve Bayes Classifier (NBC) [35] and the Dictionary-based approach [6]. We selected the NBC method mainly because it has great performance and it is one of the simplest classifiers to be coded in every programming language because of the simple
mathematics involved [35]. In addition, the Dictionary-based approached will be used because, according to [44], it has more consistency and reliability with completely unseen data across different domains.

1.5 Roadmap

In this thesis, we experiment with two sentiment analysis methods: the Lexicon-based approach using the VADER Lexicon and rule-based sentiment analysis technique, and the Machine Learning approach using the Naïve Bayes classifier. In our experiments, we manually created datasets to train models and to test our implementations, and then performed extensive tests in an effort to achieve high accuracy. Chapter 2 starts with a high level analysis of the intended system, showing how the framework was separated into two different stages for more modular development and testing. Chapter 3 presents details of the system implementation using different experiments. Chapter 4 provides a detailed analysis about the combination of the methods and different obstacles that we faced. Finally, Chapter 5 concludes with the current status of the study and proposes future work to be done.

2 Framework Design

Since our framework involves not only sentiment analysis, but also a platform for integration with other applications, we decided to separate implementation in two stages. In Section 2.1 we present the first phase which consists of building the initial steps needed for the creation of our platform: the web-service deployment, a Twitter API controller, and the creation of endpoints for easy integration of the system with external any application in a real-time environment. In Section 2.2 we present the second stage, which is strongly focused on analyzing and testing sentiment analysis techniques and integrating our final solution with the first step.
2.1 Stage 1 – Building Webservice with Real-time Twitter API Searching Capabilities Using an External Sentiment Analysis Library

This first test was conducted using “Projections and Interactions” [4], an independent system that was exhibited at ImagineRIT 2016. With the creation of the RESTful server, it was very simple for them to consume our API. To access and read Twitter data, we need to use Twitter API. Because this study is focused on making a more modular real-time implementation, we needed to create a webserver that receives a tweet and outputs its polarity: either positive or negative.

For the tweets’ sentiment analysis classification, an external tool called Japerk Text Processing was used, as we can see in Figure 2, which was later replaced in the second stage. During this test, our service was continuously identifying and categorizing
tweets. These tweets were collected in real-time and an overall percentage of their polarity was calculated. This information was used then by “Projections and Interactions” to generate visuals representing the personality of the audience at the performance.

2.1.1 Subjects

The subjects for this study were volunteers who attended to the ImagineRIT festival in May 2016. It was fairly easy to gather a significant amount of volunteers thanks to the platforms provided by Rochester Institute of Technology for this festival.

2.1.2 Materials

We used Twitter API to pull the data we needed to analyze from the volunteers in real-time. The principle behind Twitter API's design is to provide a single method for accessing the Twitter API [11]. Our current server is hosted in an Amazon EC2 instance. The cloud hosting provided by AWS provides us with dynamically resizable servers [12], which is ideal for these tests where the number of users of our service is unlimited. This experiment was implemented using NodeJs based on Restify.

2.1.3 Data Collection

Although the main purpose of this stage of the project was mainly to build the webserver and experiment with Twitter API to pull specific tweets in real-time for further classification using an external sentiment analysis classification tool, we were also able to create a dataset with tweets from April 27, 2016 to May 12, 2016. The dataset consists of 394 tweets with the hashtag “ImagineRIT”, which were later used for sentiment analysis classification. This dataset was pretty unique, the best was that pretty much every single tweet was organic, there were some retweets and so on, but in general it was a really clean collection of tweets, however, the worst was that most of the tweets were positive and that was not good to train our model, the ideal is to have a balance between the different classes.
2.1.4 Experimental Procedure

“Projections and Interactions” [4] is a game that interacts with a group by posing questions related to how they would react in social situations. Based on their answers, a graphic is generated which is meant to represent the group’s personality. Our implementation allows “Projections and Interactions” to achieve a higher level of customization as it reads the mood of those posting on Twitter, commenting on the project, or perhaps about ImagineRIT in general and creates the environment in which the generated graphic representing the group personality is created (Figure 3). The background will shift depending on if the overall mood of the crowd is happy, sad, or even closer to neutral. This helps make every experience unique and links it further to the group that is participating. It was really exciting for the families who attended to this exhibit to see a representation of all the family members personality in together in one graphic.

Figure 3. Graphic Representing an Introverted Personality using “Projections and Interactions” at ImagineRIT
2.2 Stage 2 – Integrating Sentiment Analysis Framework with an External System

In the first stage, we focused on testing only a certain part of our solution, but in this second stage we tested the rest of our implementation. In other words, we tested how our sentiment analysis technique performs classifying tweets in a real-time environment. For this we integrated our framework with MarketersPro, an internal tool currently used at Intuit by content moderators and marketers to get real-time organic content from social networks, including Twitter. A graphical representation of the second stage experiment is presented in Figure 4.

2.2.1 Subjects

The subjects were customers who tweeted using the hashtag “#TurboTax”. We collected 2,702 tweets from all over the world. Filtering tweets from TurboTax Official’s account and removing all the retweeted tweets helped to minimize the existent tweet biases and noise.
2.2.2 Materials

MarketersPro is based on searching for social media content in real-time, thus our system has to intercept the search responses to tag each tweet with its polarity. In order to do this, we needed to modify their system in both the backend and frontend. Also, It’s worth it to mention that In this stage, we were forced to change our implementation from NodeJs to Python; in Chapter 4 this will be explained in more detail.

2.2.3 Data Collection

The first training data set we used is hosted by University of Michigan [47]. This data set contains 7,086 sentences labeled with 0 or 1 depending on its polarity, and since the sentences were shorter than 160 characters and they were extracted from social media, we used them as tweets. In addition, we also included the Sentiment140 Dataset [46], which consist in 160,000 processed tweets.

Even when we had enough data to train our predictive model, we were able to increase the accuracy by creating our own datasets in our specific domain. The result was that by using Twitter Search API we gathered and manually classified two more datasets; one for ImagineRIT, as we mentioned in Section 2.1.3, and another dataset with the word domain “TurboTax”. The latter consists of a collection of 2,702 tweets. From these tweets, 697 were removed because they were tweets directly from the TurboTax account, and another 234 tweets were removed because they were retweeted. The remaining 1,771 tweets were used to train and test our model.

2.2.4 Experimental Procedure

MarketersPro is an internal tool developed by Intuit. The idea behind this project is to try and address the problem of people’s nervousness about doing taxes on their own and using TurboTax. Through research and testing, the social media team discovered that word of mouth is the most effective way to introduce a product to people. TurboTax customers were more likely to create an account or log in if they were presented with
relevant, positive social media posts about TurboTax. MarketersPro allows the marketers to search from social media channels, approve tweets, and create widgets with this organic content that later will be integrated into different TurboTax landing pages. My implementation will apply sentiment analysis to Twitter searches; this will help the marketers save time in identifying positive reviews. For this experiment we have to integrate our solution with their back-end and do some light changes in their front-end to show the class and the valence on-hover as we can see in the Figure 5.

![Figure 5. Graphic Representing a Real-Time Search on MarketersPro with Sentiment Analysis Classification Integrated](image-url)
3 System Implementation

In the previous chapter we showed how we separated our implementation into two different stages. This helped us to achieve a more modular development and testing. Also, we showed how we successfully integrated our framework in each phase with external applications. In this chapter, we will focus mainly on the second phase because this is where the core functionality of the system was developed.

3.1 Approaches to Building Lexicon-Based Method

Figure 6 shows our first approach, using AFINN-111, which is an effective lexicon that contains 2,477 words and phrases rated with an integer from minus five to plus five, depending on its polarity. Using AFINN-111 Sentiment Lexicon, we can assign a score to each of the words in the tweet and then sum all the word scores to determine the tweet classification.

![Figure 6. Lexicon-Based Approach based on AFINN-111](image)

After multiple tests, we focused on trying to increase the accuracy of this method, so we implemented our own algorithm in an effort to dynamically grow AFINN-111 as the Figure 7 shows.

The proposed algorithm added new words to the lexicon based on the word frequency. The polarity of the new word was determined based on the polarity of the tweet where the word was found. The idea is to calculate the tweet polarity using AFINN-111, using the same approach shown before, but then each word of every given tweet is search on the AFINN-111 effective lexicon. If the word is not found, then we add it to a temporary file with its frequency of appearance, and the summation of its polarity, from minus one to plus one. If the absolute value of the summation of a word polarity is more than ten,
this means that the word was present more than ten times in tweets with that positive or negative classification. This allows us to assume that this word has that same positive or negative classification and we can then save that word as such in AFINN-111, expanding it dynamically.

![Figure 7. Lexicon-Based Approach based on AFINN-111 with Custom AFINN-111 Updater Algorithm](image)

Even though we achieved slight improvement, we were still not satisfied with its performance. The biggest shortcoming was that it would require a large amount of tweets to be classified before the lexicon would start assigning correct classification to new words. The speed performance was also significantly affected, which for our purpose was unacceptable as we are creating a solution for use in a real-time environment.

Figure 8 shows the next approach that we tested with. This one includes POS tagging, which tags each word as a noun, verb, adjective, etc. and also Word Sense Disambiguation (WSD), whose main goal is to find which sense of a word was used in a given tweet. This information helps to SentiWordNet to determine the positive or negative score of a word more accurately. As we mentioned in 1.4, Wordnet is a lexical database for the English language [22] and SentiWordNet, which replaces AFINN-111, is an extension of WordNet that assigns a sentiment score (positive, negative or neutral) to each synset of WordNet. This is significant improvement as before we were only counting the positive and negative words found in a tweet, and now we are considering the context.
The biggest weakness of this method is that we were not considering other factors like punctuation, capitalization, intensifiers, contrastive conjunctions and tri-gram examination [46]. For example, an exclamation point increases the intensity of a word, an uppercase word could emphasize a sentiment, adverbs before an adjective also could increase or decrease the sentiment intensity, or even a simple “but” or negation could completely invert a sentiment’s polarity.

In an effort to improve the previous solution, we incorporated the main concept from the rule-based approach, defining five rules as Figure 9 shows. A rule-based sentiment classification system basically follows a list of detailed rules; these rules are created from observation of the convention of natural languages. With the introduction of these heuristic rules we were able obtain a higher classification accuracy.

3.2 Approaches to Building Learning-Based Method

As we mentioned in Section 1.3.2.1, the Naïve Bayes Classifier is a Probabilistic Classifier, a sub-classification of the Supervised Learning Method. It uses mixture
models, which assume that each class is a component of the model [6]. The NBC is the most famous probabilistic classifier and which was clear when researching about its implementation since it’s almost standard. This is one of the most used classifiers mainly because of its simplicity to be coded due to the simple mathematics involved [35].

The model works with a bag-of-words, which is an un-ordered set of words. The frequency of each word is kept, but not the position. Vectorization is used to transform the set of tweets into numerical feature vectors. The vectorization involves three main steps: tokenizing, counting and normalizing the tweet. The Bayes’ theorem describes the probability of the occurrence of an event based on an associate event [48]. Following the same principle, the Naïve Bayes Classifier takes the decisions of determining the class of a feature. In our implementation the features of each tweet are obtained using the module sklearn.feature_extraction module from the class sklearn of Scikit-learn [49]. Figure 10 presents an overview of the implementation of our Naïve Bayes Classifier.

**Figure 10. Learning-Based Approach Architecture**

In the Machine Learning methods, the datasets are extremely important to create an efficient classifier. The datasets are split into two, for both training and testing the
classifier. In other words, we use the training set to tune the classifier and the testing set to assess its performance.

We began by initially testing our approach with a dataset hosted by University of Michigan [47], and then with the Sentiment140 Dataset [46]. Although we had a decent performance with general tweets, we knew we could increase the accuracy if we trained our model with data strictly related to our topic, “turbotax”. We then trained our predictive model with the turbotax dataset that we manually created, with a total of 1,771 tweets after being filtered.

3.3 Technologies Used

Since the MarketersPro project is based on AngularJs and NodeJs, we were initially trying to develop the SA methods in Javascript, but after trying many different libraries we concluded that one of the best platforms for this was Natural Language Toolkit (NLTK). We opted to implement the SA methods in Python and created a web service to integrate it with MarketersPro. Our python server runs on an EC2 instance in Amazon Web Services (AWS), using Ubuntu.

As we know, the purpose of the creation of the web service is to make it possible to exchange data with the applications that use our solution. The communication is over HTTP using JSON as the format. For “Projections and Interactions”, the Twitter API version to be consumed is version 1.1, and in this case, the search will be based on hashtags. For MarketersPro we did not have to use Twitter’s API since we received the tweets to be classified. Twitter provides an Application Programming Interface (API) to provide access to Twitter’s data. Twitter uses a single method that is very useful because it is translated into less code needing maintenance [11]. Twitter’s API is also really flexible; it has a large amount of filters available to search by and many other important features such as pagination. One big advantage of using Twitter and its API is that we instantly have access to billions of tweets. For example, Yang and Leskovec [37] collected 476 million tweets from 20 million users in only seven months.
To integrate the sentiment classification to Marketers Pro we had to modify both the backend and frontend of their system, the former is built using mainly NodeJs (Gulp, RequireJs, etc) and the latter one uses AngularJs (Bower, jQuery, Less, CSS3, HTML, etc). In the backend, we intercept the reply from Twitter and in real-time classify the tweet and tag it with its polarity. In the front-end we show an emoticon representing the positive (more than 0.6), negative (less than 0.4) or neutral classification (between 0.5 and 0.6), and we also show the polarity index for each respective tweet’s search (Figure 5).

4 Analysis

In this research, we are studying the combination of two different approaches, the Lexicon-Based and the Machine Learning, specifically the Naïve Bayes. In this Chapter, we are going to present some of the different ideas that we went through in the process of finding our final architecture.

4.1 Combining Lexicon-Based and Learning-Based Methods

One of the first approaches was to classify each tweet through both classifiers and then calculate the average polarity, but after measuring the accuracy and speed, the results were not favorable. We were not performing any better and the speed decreased significantly. We needed to create a system fast enough to be “real-time” and also increased in accuracy.

After more testing, we found a pattern: a larger amount of the tweets that were incorrectly classified by both methods, had a polarity between 0.4 and 0.6 where 0 is negative and 1 is positive. In other words, only a small portion of the incorrectly classified tweets were classified with a value greater than 0.6 and less than 0.4, tweets that we called “strongly classified”. Based on this, we saw an opportunity to limit the scope of tweets to be classified twice to those tweets with a polarity classification between 0.4 and 0.6. This was really useful because we could reduce the overhead of
the process by a considerable amount. For example, in the Turbotax dataset, 281 tweets out of 1,771 were found with a polarity between 0.4 and 0.6. Because the majority of the time the VADER Lexicon and Rule-Based Sentiment Analysis technique was performing better than the Naïve-Bayes, the architecture in Figure 11 was created, where the tweet would be first analyzed by the Lexicon-Based technique and if it was not “strongly classified” then it would be analyzed by the Naïve-Bayes classifier, and an average of both polarities would determine the tweet classification.

![Figure 11. One of the Initial Architectural Overview Variations: SA Methods](image)

Order is Predetermined

Continuous tests with similar and previously unseen tweets helped us to determine that even when the Lexicon-Based classification was more reliable and consistent with completely unseen data and across different domains, the Naïve Bayes Classifier would perform better for tweets similar to the ones used to train its predictive model. Since our implementation was targeting a product, MarketersPro, which is associated with the hashtag “Turbotax”, in an effort to increase accuracy we created and manually classified
a dataset where all the tweets contained the word “TurboTax”, a word that we called the “main feature” of the dataset. The first experiment was to train the Naïve Bayes predictive model using 90% of the dataset created during ImagineRIT, and then using the other 10% to test NBC accuracy. We also tested it using the the VADER Sentiment Lexicon and the results showed that the NBC performed better. At this point we gathered, filtered and manually classified 1,771 tweets containing the hashtag Turbotax.

Figure 12. One of the Initial Architectural Overview Variations: SA Methods order changes dynamically depending on the tweet’s content.

Figure 12 shows a more elaborated architecture; the main difference between this and that in Figure 11 is that in Figure 12 the order of the sentiment analysis methods varies depending on the tweet’s content. For example, given a tweet that contains the word “Turbotax”, which is a NBC’s “main feature” word, it would be classified by predictive model and only if it was not “strongly classified”, then it would be classified by the Lexicon-based approach. If the tweet does not contain a word which is a known “main
feature”, then it would go through the Lexicon-based approach first. Independent of which method(s) was/were used, the polarity of the tweet is finally calculated and the tweet classification is obtained. Since the architecture gives preference to the sentiment analysis technique that will perform better depending on the tweet, we can achieve an increased accuracy. However, we found a way to slightly increase the accuracy more as we can see in Figure 13.

![Figure 13. Final Architectural Overview: SA Methods order changes dynamically depending on the tweet’s content, and Lexicon-Based trains NBC.](image)

In our tests, using the “strong classified” tweets from the Lexicon-based to train the Naïve Bayes Classifier made the system perform slightly better. This is because even though there are tweets training the NBC incorrectly, the number of tweets training it
correctly is still enough to slightly increase the overall accuracy. Depending on how performs the lexicon-based classifier, using the tweets that are classified as “strong classified” to train the Naïve Bayes Classifier could slightly outperform the previous implementation. However, a potential weakness is that if the VADER Lexicon Sentiment Analysis decreases its accuracy in the “strong classified” tweets, it will automatically make the NBC decrease its accuracy as well because this latter one

4.2 Data

It is evident that the supervised learning classifiers’ success depends not only on the approach or algorithm used, but also on the efficient utilization of the datasets. Our datasets are fundamentally collections of data saved in a JSON format and grouped depending on the criteria used to find them, in this case the data. We used three existing datasets and created two datasets using Twitter Search API. The data sets were created to encompass tweets related with an event and a brand, ImagineRIT and TurboTax respectively.

![Figure 14. Graph Depicting the Percent of Tweets in Each Dataset](image)

To train the probabilistic classifier, we utilized two existing datasets: one hosted by
University of Michigan [47], containing 7,086 sentences (used as tweets) labeled with 0 or 1 depending on its sentiment polarity. The second dataset, Sentiment140, contains 160,000 tweets, each classified based on the emoticons found within each tweet. Of the two datasets that were created, the first was made of 394 tweets containing the word “ImagineRIT”, and the second contained 1,771 tweets with the word “Turbotax”. Figure 14 depicts the distribution of tweets from each dataset used to train the model. Even though the created datasets are considerably smaller, the performance increases dramatically when a tweet containing one of the mentioned words is classified. These tweets that were manually classified follow the JSON format, shown in Figure 15.

![Figure 15. Format Used to Organize Tweets in the Created Datasets](image)

It’s important to have a balance between the amount of tweets from each classification, as this helps us in the training of the model so we can achieve more efficient classification of tweets. Figure 16 compares the distribution of tweet classifications in each dataset. For example, in the ImagineRIT dataset we have 80 neutral tweets (20%), 311 positive tweets (78%) and only three negative tweets (0.76%). This lack of balance between positive and negative classifications is not the ideal scenario because we need more equal training data from each class, and this could in turn affect the accuracy of the classifier. However, this was expected as most of the tweets were from students and student’s family who were having fun in the event. Even when the ImagineRIT dataset makes up less than 1% of the total tweets of all four datasets, it is very
significant if we are applying sentiment analysis to tweets specifically related to the ImagineRIT event. The University of Michigan Dataset and the Sentiment140 Dataset do not contain any neutral tweets, so we are focusing on the two classes that are consistent in all four datasets, positive and negative.

![Figure 16. Dataset Class Distribution Representation in Percent](image)

### 5 Conclusions

#### 5.1 Summary of Conclusions

Our initial goal was to develop a technique to combine two different sentiment analysis methods that yielded an accurate sentiment analysis classification in a real-time environment using Twitter as the source of content. Such a method would be very useful for systems that need to classify tweets in real-time mainly from a specific domain, but not limited to that. Our system, depending on the tweet's content, would be able to delegate, to one of the methods, the responsibility of classifying and would use the second method for double validation and in some cases for training.
Our solution implements two different sentiment analysis approaches, Lexicon-Based and Machine Learning, and it is able to dynamically organize the architecture depending on which method should be used first based on the tweet received. It was fascinating to see how the lexicons and the training datasets are probably the most important piece in a sentiment analysis system.

Delimiting our research to one specific social media, in this case Twitter, was extremely important since the implementations vary excessively depending on the type of data that a sentiment analysis system system has to classify. We provided a solution that fulfilled the hypothesis of combining different methods to achieve a higher accuracy, however, there are a lot of situations with room for improvement, such as increasing the accuracy for those questions that contain opinion words, but do not express any sentiment, or those sarcastic sentences, especially in the TurboTax dataset, where independently from the presence of sentiment words, it is difficult to deal with because the opinion words implies the inverse of its usual meaning.

5.2 Future Directions

We visualize a number of future directions for this study. We mainly concentrated efforts on classifying positive, negative or neutral correctly, but it would be interesting to distinguish between subjective and objective tweets, since this would be a great initial filter. Also, in our approach we handle the topic of classification for tweets based on the presence or absence of a keyword that we called “main feature”, but a more elaborated method could be developed where the context or even synonyms are taken into consideration.

Not only expanding the amount of manual classified datasets, but also expanding the domain of the inputs would be interesting. In this study we focused on Twitter, but more social networks could be added. Some changes we identified are to include completely different types of training datasets, oriented to other social networks, and to include more heuristic rules present in other domain contexts. Additionally, it would be interesting to create datasets with a more balanced distribution of tweets from different classes.
References


[27] Data Mining, Southeast Asia Edition: Edition 2. April 6, 2006. Jiawei Han, Jian Pei, Micheline Kamber & Morgan Kaufmann


[34] Yang, J. and Leskovec, J. Temporal Variation in Online Media. ACM International Conference on Web Search and Data Mining (WSDM ’11). Hong Kong. February 9-12, 2011.


Appendix A

User Manual

To run the project, make sure to install its dependencies:

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backports-abc</td>
<td>0.4</td>
</tr>
<tr>
<td>NLTK</td>
<td>3.2.1</td>
</tr>
<tr>
<td>Numpy</td>
<td>1.11.1</td>
</tr>
<tr>
<td>Scikit-learn</td>
<td>0.17.1</td>
</tr>
<tr>
<td>Scipy</td>
<td>0.17.1</td>
</tr>
<tr>
<td>Textblob</td>
<td>0.11.1</td>
</tr>
<tr>
<td>Tornado</td>
<td>4.4</td>
</tr>
</tbody>
</table>

Once you fork/clone the repository from GitHub, you will find the following folders structure:

<table>
<thead>
<tr>
<th>Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>classifiers</td>
<td>This folder contains all the classifiers (NaiveBayes classifiers) and vectorizers trained on different datasets. Each one of these files for the classifier and vectorizer will follow the formats: {DATASET}-classifier.p and {DATASET}-vectorizer.p to indicate the dataset where the data was trained on.</td>
</tr>
<tr>
<td>datasets_for_testing</td>
<td>Contains all the datasets to test the sentiment extractor models. This file are in JSON format and contains a list of tweets objects with the fields 'text', which is the text content of the tweet and, 'sentiment' that is the sentiment polarity.</td>
</tr>
<tr>
<td>datasets_for_training</td>
<td>Contains all the datasets to train the classifiers. This file are in JSON format and contains a list of tweets objects with the fields 'text', which is the text content of the tweet and, 'sentiment' that is the sentiment polarity</td>
</tr>
</tbody>
</table>
Also, the following python scripts which we describe in details:

<table>
<thead>
<tr>
<th>Script Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>general.py</td>
<td>This script file contains general purpose functions such as loading and storing datasets, classifiers, and vectorizers.</td>
</tr>
<tr>
<td>settings.py</td>
<td>This script file contains the settings of the project.</td>
</tr>
<tr>
<td>sentiment_extractos.py</td>
<td>Contains all the built sentiment extractors. A sentiment extractor is an object that takes as an input a text corresponding to a tweet and outputs a number indicating the sentiment of the given tweet.</td>
</tr>
<tr>
<td>process_sentiment_140_dataset.py</td>
<td>This script processes the raw data for the dataset sentiment 140</td>
</tr>
<tr>
<td>process_json_raw_datasets.py</td>
<td>This script processes the json raw datasets.</td>
</tr>
<tr>
<td>evaluate_sentiment_extractors.py</td>
<td>Evaluates the sentiment extractors (Lexicon based, machine learning based, and improved hybrid) on all the datasets.</td>
</tr>
<tr>
<td>train_naive_bayes_classifier.py</td>
<td>This script contains the functionality to train a naive bayes classifier on a given dataset.</td>
</tr>
<tr>
<td>test_naive_bayes_classifier.py</td>
<td>This script contains the functionality to test a naive bayes classifier on a given dataset.</td>
</tr>
<tr>
<td>tweets_server.py</td>
<td>This script contains the API server for sentiment analysis of tweets.</td>
</tr>
</tbody>
</table>

Finally, we are going to explain how to use the scripts:

- To start the tweets API server run the following command (for example): python3 tweets_server.py --port 12345
- To process the raw data for the sentiment 140 dataset run the command: python3 process_sentiment_140_dataset.py
- Use settings.py this script to define and change the settings values used by other scripts
- In order to evaluate all the built sentiment extractors on all the testing datasets, simply have to run the command: python3 evaluate_sentiment_extractors.py, As metric classification accuracy is going to be used. This is going to be printed out on the screen.
- To train a classifier simply run the following command: python3 train_naive_bayes_sentiment_classifier.py --dataset={NAMES} --prefix={PREFIX}
- To test a classifier simply run the following command: python3 test_naive_bayes_sentiment_classifier.py --dataset={NAMES } --prefix={PREFIX}