TOWSON UNIVERSITY
OFFICE OF GRADUATE STUDIES

A SYSTEM FOR COLLECTION AND ANALYSIS OF OPINIONS IN MICROBLOG DATA: A TEXT MINING APPROACH

by

Ranjan M. Vaidyanathakumar

A thesis

Presented to the faculty of

Towson University

in partial fulfillment

of the requirements for the degree

Master of Science

Department of Computer Science

Towson University
Towson, Maryland 21252

(January, 2013)

TOWSON UNIVERSITY
OFFICE OF GRADUATE STUDIES

THESIS APPROVAL PAGE

This is to certify that the thesis prepared by Ranjan M. Vaidyanathakumar

entitled A SYSTEM FOR COLLECTION AND ANALYSIS OF OPINIONS IN MICROBLOG DATA: A TEXT MINING APPROACH

has been approved by the thesis committee as satisfactorily completing the thesis requirements for the degree Master of Science

[signature]	12/20/12
Chair, Thesis Committee	Date
[signature] JOSH DEHLINGER	12-20-12
Committee Member	Date
[signature] Michael McGuire	12-20-12
Committee Member	Date
Committee Member	Date
Committee Member	Date
Janet V. DiIorio	12-20-12
Dean of Graduate Studies	Date

ACKNOWLEDGMENTS

Completing my Master's thesis is one of the most challenging activities in my career. It has been a great privilege to spend two years in the Department of Computer Science at Towson University, and its members will always remain dear to me.

My first debt of gratitude must go to my advisor, Dr.Siddharth Kaza. It was a tremendous learning curve for me in the field of data mining, web and mobile development under his guidance. I am thankful for his trust in me and opportunities he provided to learn as a researcher. I am grateful for his unflagging grooming and serving as a role model to me in the field of academia. He has been my mentor, constructive critic, strong and supportive adviser throughout my graduate school journey.

Special thanks to my committee, Dr. Josh Dehlinger and Dr. Michael P. McGuire for their guidance and helpful suggestions. I also thank other faculty members, Dr. Ramesh Karne, Dr. Sharma Pillutla, and Dr. Amy Becker for their support during my studies. I owe them my heartfelt appreciation.

My friends and well-wishers in US, India and other parts of the world were sources of laughter, joy and sharing. Special thanks to Lakshmi, Varun, Sagar, Praveen, Ankita, Jyothsna, Poornima, Rana, Srikanth, Bhagya, Prashanth, Mallik, Usha Karne.

Most of all, I owe everything to my family, my mom, Lakshmi, uncle, Nagaraju, and my dearest sister, nephew and cousins. Their love and trust provided my inspiration and was my driving force.

ABSTRACT

A SYSTEM FOR COLLECTION AND ANALYSIS OF OPINIONS IN MICROBLOG DATA: A TEXT MINING APPROACH

Ranjan M. Vaidyanathakumar

Microblogging has become a very popular communication platform among Internet users. Its applications are rich sources of data for text mining, opinion mining and sentiment analysis. Its services are also becoming a platform for marketing and public relations for organizations and political parties. Political parties are interested to know if people support their program or not. Social organizations are asking people's opinion on current debates. All this information can be obtained from microblogging services, as their users post everyday what they like/dislike, and their opinions on many aspects of their life. In our paper, we focus on using Twitter, the most popular microblogging platform, for the task of text mining and opinion mining commonly known as sentiment analysis. We propose a system to acquire, manage, manipulate, analyze microblog data and report results. We discuss and apply various text processing techniques for opinion mining and apply several machine learning algorithms to analyze if bloggers have an opinion on a particular issue. In this study, we collected over 9,256,819 tweets on the issue of same-sex marriage in Maryland and across USA. Using Naïve Bayes and support vector machine classifiers we find that we can identify opinionated tweets with an accuracy of 90% and 55% respectively.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

According to an August 2011 study by the Pew Research Center's Internet & American Life Project, 65% of adults use social media applications (up from 61% in 2010 and 29% in 2008) and 43% of adults visit these sites on a daily basis (up from 38% in 2010 and 13% in 2008) [1]. This is the single largest shift of users to an application (over half a billion users over all online social networks) since the move to the world-wide-web [2].

With the increase in social media applications public opinion has become increasingly useful tool for guiding the decisions made by political and business entities. Unfortunately, classical methods for measuring public opinion are time consuming, expensive and error prone. A number of these limitations can be overcome using freely available data sources from online social networking sites and microblogs such as Twitter. Through the analysis of millions of microblog messages and effective text processing techniques and machine learning algorithms we could classify the opinion of user generated text.

The aim of this thesis is to acquire Twitter data through Twitter APIs, manage them in easily and effective manner, manipulate and process them for text mining activities, analyze the processed data and report results.

The selection of a sample of data to process for a specific study is a crucial issue in order to obtain meaningful results. For example, the use of a very small sample of data may introduce biases in the output and lead to incorrect inferences or misleading

conclusions. The acquisition of large data manually from social media applications is a difficult task. In order to overcome these data acquisition problems, we developed an automated data collecting system. Acquired data is parsed, along with the addition of some derived linguistic features and the removal of others and subsequent insertion into a database. Structured data is then used for information retrieval, lexical analysis to study word frequency distributions, and pattern recognition. The overarching goal is to turn text into data for analysis, via application of natural language processing (NLP) and analytical methods.

We focused on social media content related to debate over the issue of same-sex marriage. These tweets give insight about subjectivity of issue like person's opinion, objective information like unbiased opinion and polarity of the issue like support, nonsupport or being neutral. Given the increased focus on the issue in Maryland and across USA, the debate offers a natural first case study for this thesis. Keeping this issue in mind and issues which could impact in broader sense we framed the following research questions.

- How can we build a system that extracts structured information from micro-blog text for mining?
- What are some robust methods for identifying the opinion described in a text?
- How opinionated (positive or negative) or neutral are people about the issue?

Social media has a growing potential to continue to revolutionize the communication space and the way we process and share information. Individuals are in a position to project their influence across space and time with unprecedented ease. Shedding light on

this will provide benefits to various fields of study ranging from sociology, digital government, computer science, and society in general.

2. LITERATURE REVIEW

This section discusses points of current knowledge including essential findings in the area of machine learning, natural language processing, classification methods, twitter studies and tool studied for this study.

2.1 Machine Learning

Machine Learning (ML) is a branch of computer science and more specifically a branch of artificial intelligence which aims to create smart systems with the development of algorithms that take as input empirical data. I present a brief overview of ML here.

The first definition of machine learning was provided by Arthur Samuel in1959 as a "Field of study that gives computers the ability to learn without being explicitly programmed" [3]. In 1997, Tom M. Mitchell provided a widely quoted, more formal definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" [4].

Machine Learning largely relies on statistical models to allow a learning agent to learn about its environment. Environmental input is always in the form of data; however the ability of the learning agent to affect change upon its environment varies largely across problem domains. For example, ML approaches could be used for designing a robotic control system or a system to make predictions about future weather patterns. Both systems environmental input takes the form of data (video, audio, and sonar for the robot, or temperature, wind, and humidity for the weather predictor), but while the robot may have a number of ways to influence its environment and thus influence its own future

input, the weather predictor can do nothing to influence future climates directly. Some other applications are listed below:

1. **Database mining:** Machine learning has recently become vital because of the huge amount of data being generated in various application domains. There are large datasets from growth of automated data from web from structured data to unstructured data. Sources of data include Web data like click-stream or click through data, medical records, biological data like gene sequences, engineering information from data from sensors, log reports, and photos [5].
2. **Self-customizing programs** like Netflix, Amazon, iTunes genius by taking users info to learn based on its behavior. For example, Netflix faces a problem to improve the accuracy of future movie recommendation based on their movie preferences. Machine learning algorithms are used to predict user ratings for films, based on previous ratings without any other information about the users or films [6].
3. **Understand human learning and the brain:** By building systems that mimic (or try to mimic) how the brain works we can have a better understanding of the associated neurobiology [7].

For machine learning tasks, data can be seen as instances of the possible relations between observed variables; the algorithm acts as a machine learner that studies a portion of the observed data called examples or training data. In a classification/regression problem, training data is the knowledge about the data source which we use to construct the classifier/regressor. It helps to capture characteristics of interest of the data's unknown

underlying probability distribution, and employs the knowledge it has learned to make intelligent decisions based on new input data [8].

There are three approaches of machine learning in a natural-language-processing context - supervised learning (with completely labeled training data), unsupervised learning (without any labeled training data) and semi-supervised learning which is a combination of both. These approaches are not mutually exclusive and frequently a combination of the two can/must be used to achieve desired results.

2.1.1 Supervised Learning

In supervised systems, the data as presented to a machine learning algorithm is fully labeled. A classifier is learned from the data, the process of assigning labels to yet unseen instances is called classification. That means: all examples are presented with a classification that the machine is meant to reproduce. An algorithm is presented input in some form and an output or behavior is generated. This output is then compared to a desired output for the given input and the model (classifier) is adjusted in an effort to minimize the error between the actual and desired output. This process is referred to as training, and an algorithm has learned to correctly map input to output it will be able to generate correct outputs for inputs on which it was not explicitly trained.

We cannot distinguish between characteristic properties of training data and peculiarities (e.g., noise) of specific sample. If we fit training data too closely, we model sample peculiarities which is called over-fitting. Supervised learning is frequently faster than unsupervised learning; however problems such as over-fitting and the need for input data labeled with the correct output make it unsuitable for a number of applications.

2.1.2 Unsupervised Learning

Unsupervised systems are not provided any training examples at all and used for clustering. It is then the algorithm's job to identify patterns, similarities, or dissimilarities within the input data and use this information to produce some desirable output or behavior. This is the division of data instances into several groups. The results of clustering algorithms are data driven, hence more 'natural' and better suited to the underlying structure of the data. As one would expect, unsupervised learning is generally more difficult and prone to error than supervised learning. Regardless, the large amount of freely available unlabeled data combined with costs of labeling data makes unsupervised learning attractive. This advantage is also its major drawback: without a possibility to tell the machine what to do (like in classification), it is difficult to judge the quality of clustering results in a conclusive way.

2.1.3 Semi-supervised Learning

This special property of classification makes use of both labeled and unlabeled data for training - typically a small amount of labeled data with a large amount of unlabeled data. Traditional classifiers use only labeled data (feature / label pairs) to train. Labeled instances however are often difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile, unlabeled data may be relatively easy to collect, but there are few ways to use them. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with the labeled data, to build better classifiers. Because semi-supervised learning requires less human effort and gives higher accuracy, it is of great interest both in theory and in practice.

Due to a larger data basis, semi-supervised systems often outperform their supervised counterparts using the same labeled examples. The reason for this improvement is that more unlabeled data enables the system to model the inherent structure of the data more accurately.

2.2 Natural Language Processing

Natural language processing (NLP) is the ability of a computer program to understand human speech as it is written. NLP is a component of artificial intelligence (AI). The goal is to enable machines to understand human language and extract meaning from text. It is a field of study which falls under the category of machine learning and more specifically computational linguistics [9].

At a rudimentary level, approaches to NLP can be separated into two distinct categories, Rule Based and Machine Learning. These two approaches can also be seen as representing opposite ends of a spectrum, which employs at one end deep analysis and requires only small amounts of data, and light analysis of large amounts of data at the other.

Rule based approaches use deep analysis and known information about the rules and structure of language to elicit contextual information from text for some specific purpose. Much work using this approach relies heavily on the ideas of Noam Chomsky regarding language and grammars. In general, such approaches consist only of a processing phase and only require data to act as input for evaluation purposes.

In contrast, machine learning typically relies on statistical measures obtained from analysis of large amounts of data to determine relationships without relying on hand written rules. The availability of certain types of data has driven much of the direction of NLP work towards machine learning based approaches. An example of this is the new rich source of massive amounts of human generated text available due to the web.

Another factor attributing to the rise in popularity of machine learning for NLP is the inability of rule based systems to handle unforeseen, uncommon, or poorly constructed input. Much of this inability has to do with the rigidity of such systems and the difficulty of NLP, something which is largely regarded as an AI-complete problem, implying that the difficulty of computational problems is equivalent to solving the central artificial intelligence problem - making computers as intelligent as people, or strong AI [10]. This thesis will employ approaches falling into the Machine Learning category, thus further discussion of NLP will largely focus on statistical attributes of bodies of text.

Word Frequency: Given a body of English text an interesting statistic that can be measured is the frequency of each token. Frequency is simply the number of times each token appears in this body of text. Given a particular token *xi* and the number of tokens in a body of text *n*, the frequency of a particular token *wi* can be viewed as the probability that *wi = xi*,

$$P(wi = xi) = freq(wi)\, n \tag{1}$$

These frequencies have been shown to follow a power law distribution. In the naïve case the frequency of the i^{th} most common token, *xi*, is given by Equation 2. This frequency

distribution is known as Zipf's Law, after the linguist George Kingsley Zipf. Important to note is this law is only a generalization and breaks down for large values of *i*, as the frequency of any word appearing in a body of text cannot be less than one.

$$freq(xi) = freq(x1)\ I \tag{2}$$

Such information alone cannot directly be applied to solve the real world problems under normal circumstances. Due to the need for having an ordered list of tokens by frequency to approximate the frequency of a token, the presence of statistical normalities provide a basis on which further analysis can be done. In this study, to analyze the opinion of the sentence we used word frequency as a first approach and later we did improvement in the process by standard text preprocessing and classification techniques.

2.2.1 Opinion Mining

One of the major applications of machine learning is opinion mining. In this field, computer programs attempt to predict the emotional content or opinions of a collection of articles, blogs, and comments. This becomes useful for organizing data, such as finding positive and negative reviews, or extracting person's opinion while diminishing the need for human effort to classify the information. Opinion mining also refers to the application of natural language processing, computational linguistics, and text analytics to identify and extract subjective information in source materials [11].

The body of work we review is that which deals with the computational treatment of opinion, and subjectivity in text. This body of work is also known as review mining and appraisal extraction. There are some connections to affective computing, where the goals

include enabling computers to recognize and express emotions [6, 7]. A basic task in opinion mining is classifying the polarity of a given text at the document, sentence, or feature/aspect level - whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral [14]. Advanced sentiment (opinion) classification also looks at emotional states such as "angry," "sad," and "happy."

There have been many studies written on opinion mining for the domain of blogs and product reviews, Pan & Lee [15] gives a survey. Researchers have also analyzed the brand impact of micro-blogging [16]. There has been work on detecting sentiment in text [17] presents a simple algorithm, called semantic orientation, for detecting sentiment Pang & Lee [18] present a hierarchical scheme in which text is first classified as containing sentiment, and then classified as positive or negative. Much literature in the field of sentiment analysis has focused on different classification models for text. Previous approaches include hand-coded rules the winnow algorithm, random k-label sets, Support Vector Machines (SVM), and Naive Bayes [19].

2.3 Classification methods

This section discusses on three classifying algorithms which are among the most influential text mining algorithms in the research community.

2.3.1 Naive Bayes Classifier

A classifier based on the Naive Bayes algorithm. In order to find the probability for a label, this algorithm first uses the Bayes rule to express [20] P(*label*|*features*) in terms of P(*label*) and P(*features*|*label*):

$$P(label \mid features) = P(label) * P(features \mid label) / P(features)$$

The algorithm then makes the 'naive' assumption that all features are independent, given the label:

$$P(label \mid features) = P(label) * P(f1 \mid label) * ... * P(fn \mid label) / P(features)$$

Rather than computing P(*featues*) explicitly, the algorithm just calculates the denominator for each label, and normalizes them so they sum to one:

$$P(label \mid features) = P(label) * P(f1 \mid label) * ... * P(fn \mid label) / SUM[l](P(l) * P(f1 \mid l) * ... * P(fn \mid l))$$

In spite of their apparently over-simplified assumptions, Naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters. [21]

2.3.2 Support Vector Machine Classifier

In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier.

Support Vector Machine classifier treat each feature as a dimension, and position features in n-dimensional feature space. An optimal hyperplane is then determined that

best divides feature space into classes, and future instances classified based on which side of the hyperplane they lie on, and their proximity to it. NLTK implementation of binary SVM is used in this study - that is, only two classes are supported.

2.3.3 K-means Clustering

In data mining, k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

The K-means cluster starts with k arbitrary chosen means then allocates each vector to the cluster with the closest mean. It then recalculates the means of each cluster as the centroid of the vectors in the cluster. This process repeats until the cluster memberships stabilize. This is a hill-climbing algorithm which may converge to a local maximum. Hence the clustering is often repeated with random initial means and the most commonly occurring output means are chosen. We are using k-means clustering algorithm as our unsupervised learning model.

2.4. Studies on Twitter and other microblogs

There have been several studies on Twitter and other microblogs, in this section we focus on opinion mining related studies.

2.4.1 Previous studies on Twitter

Detailed study [22] has been done in using emoticons as labels for positive and negative sentiment. This is very relevant to twitter because many users have expressed their emoticons through their opinions in their tweets.

Studies analyzing the political debate online have focused on traditional weblogs and social media websites, such as Facebook, MySpace, and YouTube. Previous research has shown that social media is widely used for political deliberation and that this deliberation reflects the political landscape of the offline world.

Although the reference to tweets in some political commentaries [23] shows that analysts are already turning to the Twitter sphere as an indicator of political opinion, to the best of our knowledge, there are no scientific studies systematically investigating the political opinion in micro-blogs. As a result, some research has posed the question whether we can even "use the word public opinion and blogging in the same sentence" [24].

Much past work in the field of NLP has largely focused on tasks such as similarity measures or classification for web documents. Such tasks are highly relevant as they can be directly applied for effective search. This work has been incredibly fruitful, producing well known algorithms such as *tf-idf* [25]. However, the type of document generally considered for these tasks have noticeably different characteristics when compared to tweets. Twitter messages have many unique attributes, which helped us to choose these messages for our study.

1. *Length*: Tweets are much shorter than the typical web document. This means that the frequency of any word in a tweet is likely to be either zero or one. The maximum length of a Twitter message is 140 characters. This is very different from the domains of other text mining research, which was mostly focused on documents which consists of multiple sentences.

2. *Topic*: Most tweets have a single subject and are highly informal.
3. *Available data*: Another difference is the sheer magnitude of data. In Pang & Lee's [26] study, the corpus size is 2053. With the Twitter API, it is much easier to collect millions of tweets for training purpose.
4. *Language model*: Tweets are likely to contain a high degree of spelling errors. The frequency of misspellings and slang in tweets is much higher than other domains. Twitter users post messages from many different mediums, including their cell phones.

2.5 APIs and Tools Studied

In this section we present few tools which we studied for data collection and text mining activities. We studied and tested many of the features in the below mentioned tools. This exercise helped us to choose the right tool for our study.

2.5.1 Twitter API Overview

Twitter provides several methods for retrieval of data and accessing user features in the form of three Application Programming Interfaces (APIs). Those methods related to accessing user and account information will not be discussed here as they are not useful for the purposes of this thesis. The two APIs remaining are the Search API and the Stream API. Both APIs are HTTP request based, which means users interact with the API by requesting a URL which contains parameters and data is returned in the form of data at that requested URL.

2.5.2 Twitter Limitations

Twitter is an excellent social media source for researchers and data scientists to collect huge amount of data. But it is important to be aware of what it can do for you and what it can't do. There is restriction given by Twitter: method call of Twitter API is limited by 350 calls per hour for one developer account. [27] Furthermore, it is impossible to collect enough data for applying data analysis techniques without automated data collecting system in place. In order to overcome these data collection problems, we invested to develop our own Twitter data acquisition process. In this research paper, we will introduce the design specifications and explains implementation details of data collector we developed in section 3.1

2.5.3 Twitter4J

Twitter4J is an unofficial Java library for the Twitter API. With Twitter4J, we can easily integrate Java application with the Twitter service. Twitter4J comes with the following features

1. 100% Pure Java - works on any Java Platform version 1.4.2 or later
2. Zero dependency - No additional jars required
3. Built-in OAuth support

These features helped us to develop our core logic quickly and securely with less possible time. The specific classes used for our research purpose is explained in research design section. In this thesis, we will introduce the process and explains implementation details of classes which we are using in section 3.1

2.5.4 Weka

Weka is an open source data mining library which is written in Java. It contains collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. We used Weka initially to check how our dataset behaves for text classification purpose. It helped us to understand the data little bit and features we could possibly extract.

We investigated good amount of time in analyzing Weka classifiers. We found it is not best suited for text classification and opinion mining.

2.5.5 RapidMiner

RapidMiner, formerly YALE (Yet Another Learning Environment), is an environment for machine learning, data mining, text mining, predictive analytics, and business analytics. It is used for research, education, training, rapid prototyping, application development, and industrial applications.

It enables experiments to be made up of a large number of arbitrarily nestable operators, which are detailed in XML files and are made with the graphical user interface of RapidMiner. RapidMiner provides more than 500 operators for all main machine learning procedures, and it also combines learning schemes and attribute evaluators of the Weka learning environment. It is available as a stand-alone tool for data analysis and as a data-mining engine that can be integrated into any applications or products.

We investigated good amount of time in analyzing RapidMiner classifiers and (like Weka) found it is not best suited for text classification and opinion mining.

2.5.6 Natural Language Toolkit (NLTK)

The Natural Language Toolkit is a suite of libraries and programs for symbolic and statistical natural language processing for the Python programming language. NLTK includes graphical demonstrations and sample data. It is accompanied by extensive documentation, including a book that explains the underlying concepts behind the language processing tasks supported by the toolkit [28].

NLTK is ideally suited to students who are learning NLP or conducting research in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning. NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems.

Some the main feature of NLTK Packages:

- Access to documents: Interfaces for text collections
- Strings Processing: Tokenization, sentences detection, stemmers
- Discovering of collocations: Tokens which appear often together than by chance
- Part-of-speech tagging: Distinguish nouns from verbs
- Classification: Classifiers in general, based in Python dictionaries as training
- Chunking: Split up a sentence in granular units
- Syntactic Analysis: Complex analysis (syntactic and others)

- Semantic Interpretation: λ (lambda) calculus, 1st order logic
- Evaluation Metrics: Precision, coverage
- Statistics: Frequencies distribution, estimators
- Applications: WordNet browser, chatbots

Language identification is a key task in the text mining process. Successful analysis of extracted text with natural language processing or machine learning training requires a good language identification algorithm. If it fails to recognize the language, this error will nullify subsequent processes. NLP algorithms must be adjusted for different corpuses and according to the grammar of different languages. Certain NLP software is best suited to certain languages. NLTK is the most popular natural language processing package for English under Python [29]. We found the efficiency of language processing depends on features like text identification, text prepossessing, NLP and machine learning capabilities. We identified and assessed all such feature support with NLTK and hence we decided to use this for our study.

3. RESEARCH DESIGN

This section presents the research design of this study. Figure 1 shows the research design and process used to acquire data, preprocessing the text and feature representation, machine learning and evaluation of results with formative and summative activities. The following subsections explain each component.

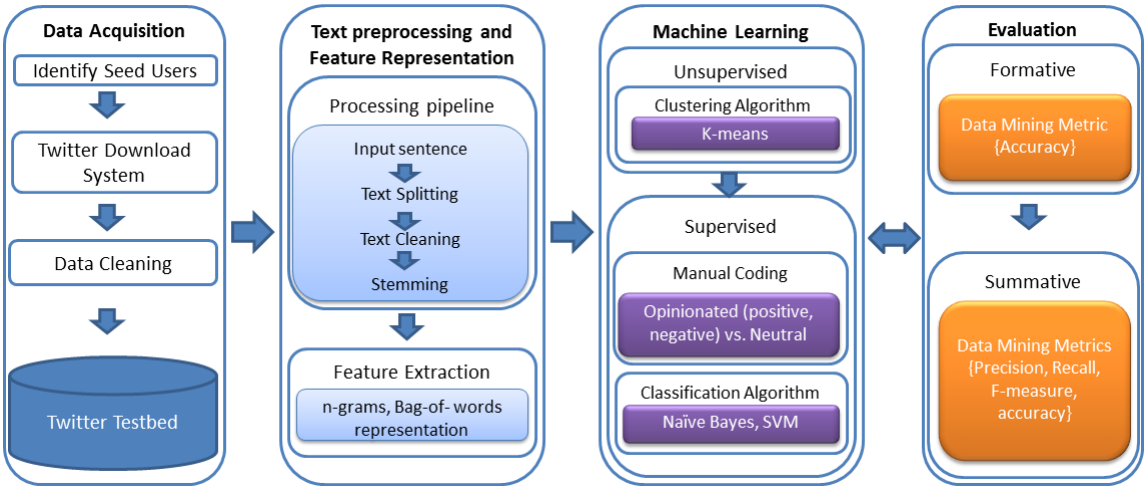

Figure 1. Research design

3.1 Data Acquisition

The data acquisition component aimed at collecting relevant posts for the study. We focused on twitter as the social media site for this study. With over 500 million active users as of 2012, generating over 340 million tweets daily, we found this is most relevant source of data collection for this study. We tried to automate this process with the minimal human intervention. The details are as follows.

3.1.1 Identify Seed Users

We identified 34 seed users for our case study with the help of social science domain expert and by searching for most influencing users of the issue. The reason why we chose these 34 seed users were because they were talking about the issue extensively and most of them were standard organizations having large number of followers. Followers are the people who have agreed to receive user's tweets through Twitter. We found popularity on Twitter is often measured by the number of followers a user has. We collected 63,955 followers in total for 34 seed users. We restricted to collect only seed user's followers as it is likely to have an enormous amount of noisy data if we go one more level down. The number of followers we have retrieved was good enough to download significant number of tweets; we can see the statistics in section 3.1.4. Table 1 shows the list of seed users used in this study.

Table 1. List of seed users

@gaymarriageusa	@theadvocatemag	@queerunity
@gaymarriagewatc	@nclrights	@marylandmoment
@md4equality	@pflag	@civilmarriage
@marylandpop	@aclulgbt	@gaycivilrights
@hrc	@freedomtomarry	@WA4Marriage
@glccb	@thetaskforce	@equalrightswa
@MEUSA	@briansbrown	@defenddoma
@allmdfamilies	@protectmarriage	@eqmatters
@aclu_md	@nomupdate	@itgetsbetter
@docequality	@SSMarriageTrap	@MarriageDefence
@freetomarry	@snoopervizion	@EqualityMD
@MDPolicy		

We used twitter search API to obtain seed users who are talking about the issue. We searched for the keywords like “gay marriage”, “marriage equality”, “same-sex” to identify right seed user.

3.1.2 Twitter Download System

Figure 2 depicts the components of twitter download system to collect and process the data to reduce noise and store it in structured form.

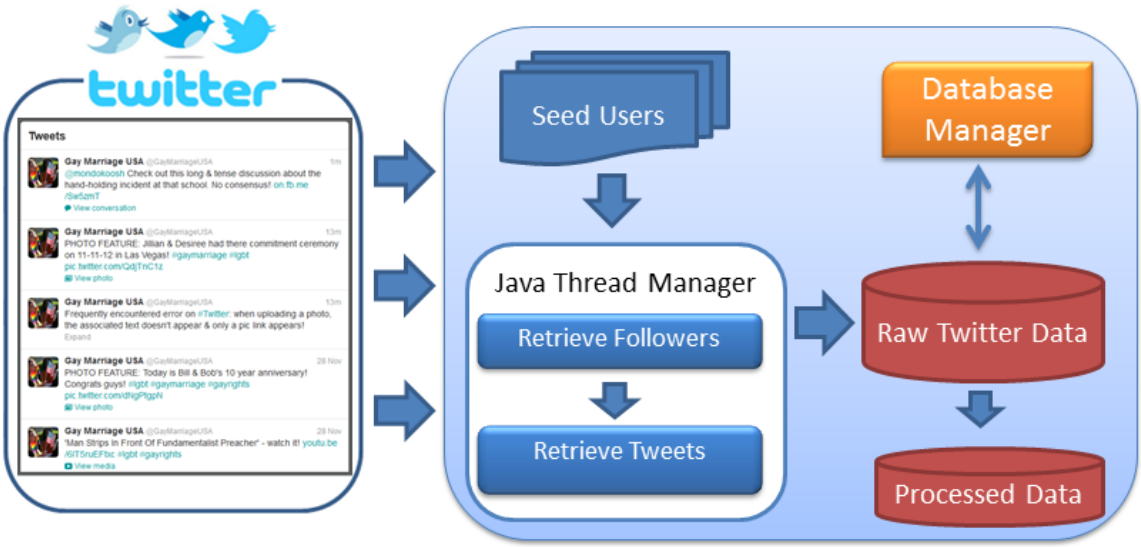

Figure 2. System architecture

The Twitter download system consists of the following components:

Java Thread Manager: The Java Thread Manager consists of two threads, retrieve followers and retrieve tweets. Each thread can perform tasks in parallel and continuously download tweets from seed users, followers and collect user information metadata. Twitter limits number of methods calls for one developer’s account to 350 calls per hour. We adapted the system design around this limit to gather tweets. We designed a process

to wait for the duration of one minute after every call. In this way we did not exceed 350 calls per hour and able to download data continuously. Previous studies have used other methods to handle multiple developers accounts to build large and parallel database by switching back and forth between developer account [30]. Let us look into each thread function.

Retrieve followers: This java thread gathers seed users and their follower's account information, metadata information and their relationship. We gathered all possible information like user's name, location, number of followers, number of followings, creation date.

Retrieve tweets: This java thread gathers tweets of seed users and their followers using user timeline method. A user timeline is a Twitter term used to describe a collected stream of Tweets for a user which is listed in real-time order. User timeline method retrieves the 20 most recent statuses posted in the last 24 hours from the user and return up to 3,200 of a user's most recent comments. Since our goal was to understand the sentiment of the issue based on user comment and we already chose the user; user timeline helped us to retrieve most relevant tweets. We gathered all possible information like tweet id, tweet, tweet creation date, location. We saved all gathered tweets into database shown in the following section.

Database Manager: The Database Manager is required to save all data gathered into database, and to retrieve stored data back. The database manager also handles processing of data to remove noise in data like meaningless data. Tweets come with lot of noisy data and in unstructured form. We saw noisy data caused by hardware failures,

new line characters, tabs, programming errors and gibberish input from user. These kinds of data cannot be understood and interpreted correctly by machines and algorithms. Noisy data can also contribute to increase the amount of storage space required and adversely affect the results of the data mining analysis [29]. We dedicated good amount of time to remove noise from the data and store in structured relational database format.

Database: We designed and implemented a database to store and handle large quantities of data in a schema that was efficient for research. Figure 3 shows schema diagram of the database. We took a snapshot of the database of two million records to conduct this study which is captured in snap_012420120930 column. This design helped us to go back and look for the raw data for any kind of improvement in the process.

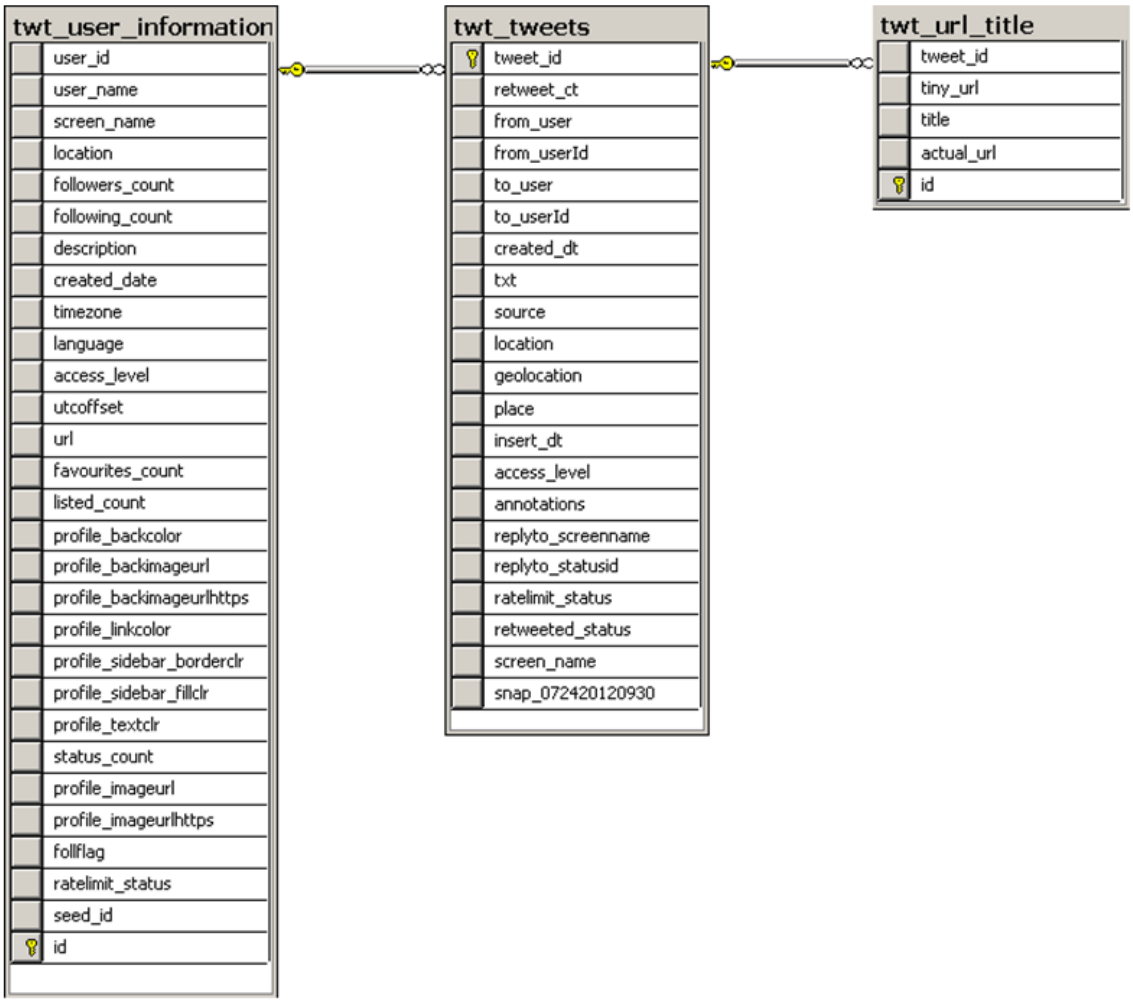

Figure 3. Database design

3.1.3 Data Cleaning

The Data Cleaning component runs on java thread which is outside the twitter download system. Tweets come with content tagged urls like bitly, goo.gl, t.co, TinyURLs which uses URL shortening service. URL shortening service is a technique on the World Wide Web in which a URL may be made substantially shorter in length and still direct to the required page. Twitter users' uses this feature to link directly to newsgroup postings that has long and cumbersome addresses and also because of limited

140 characters in tweet messages. These urls has limited span of life and decease after certain duration. Data cleaning thread parses all the urls in the tweets and replaces it with relevant title and actual url from the website. It uses JSOUP java library to perform this task. We replaced all the urls with its title to get appropriate and meaningful data and stored it separate table. We also stored actual urls in separate column for future use.

3.1.4 Twitter Testbed

Table 2 presents the basic statistics of the data collected from twitter data source.

Table 2. Twitter testbed statistics

Total tweets	9,256,819
Average number of tweets downloaded per day	~ 55,000 - 60,000
Total number of followers	63,955
Date Range	03-13-2008 to 12-12-2012

Table 3 shows list of few raw tweets which are used in this study.

Table 3. Examples of raw tweets

Gay Republicans Encounter Same-Sex Marriage Dilemma http://t.co/Od5jmGcC via @HuffPostGay
Homosexuality is not the problem. Prejudice is the problem. (Ricky Martin) #homophobia #prejudice #idaho #gaymarriage
RT @BrookeBCNN: .@GayMarriageUSA Murray Lipp coming forward for first time -- talked about petitioning DNC convention out of Charlotte h ...
@BetteMidler: Thanks for supporting equality!
Support national marriage equality! #gaymarriage #marriageequality http://t.co/dsVjA4Kd
1 year on, and same-sex couples in California still can't marry! http://fb.me/IHMqcsfJ

3.2 Text preprocessing & Feature Representation

The basic phase in opinion mining is text categorization. Text categorization is a process that group text sentences into one or more predefined categories based on their contents. To perform text categorization we followed text preprocessing and feature extraction tasks which are explained this section.

3.2.1 Processing Pipeline

The goal behind preprocessing task is to represent each sentence as a feature vector, that is, to separate the text into individual words. The following preprocessing activities were carried out to step by step to achieve this task.

Step 1. *Input the sentence*: In this step every sentence is inputted sequentially by reading it from the file.

Step 2. *Text splitting (Lemmatize, Tokenize, lowercase)*: We used NLTK's word tokenizer to tokenize words in the given sentence, also used wordnet lemmatizer to lemmatize to find out the lemma of the given word. The wordnet lemmatizer uses the wordnet database to lookup lemmas. Finally we converted every word into lowercase.

Step 3. *Text cleaning (remove stop words, punctuations, duplicates)*: We removed useless words like the, as, and which are called as stop words. We also removed punctuation that is one or two characters long. Finally we eliminated all the duplicates in the final list.

Step 4. *Stemming*: We used stemming technique to find out the root/stem of a word. Words with the same stem or word root mostly describe same or relatively close concepts in text and so words can be conflated by using stems. For example, the words, support,

supporting, supported all can be stemmed to the word 'SUPPORT'. In this study, the Porter Stemmer algorithm, which is the most commonly used algorithm in English, is used.

3.2.2 Feature Extraction

The goal behind feature extraction task is to transform the input data into the set of feature vector. Features extracted were carefully chosen and expected that the features set will extract the relevant information from the input data.

n-gram: n-gram is widely used feature extraction model in statistical natural language processing. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram"; size 3 is a "trigram" and so on. We chose each consecutive word sequence containing two words and represented as binary n-gram feature. In our study n-grams appearing in less than 0.5% of the training set sentences do not constitute a feature. We substituted the key <<url>> for all the urls and included in n-gram feature extraction. This helped us in identifying the importance of url in a tweet.

Bag-of-words Representation: Bag-of-words is a model that takes individual words in a sentence as features, assuming their conditional independence. The text is represented as an unordered collection of words. Each feature of the vector represents the existence of one word. This is effectively a unigram model, where each word is conditionally independent from the others. All the words (features) in the feature vector constitute the dictionary. The challenge with this approach is the choice of words that are appropriate to become features. Using this model the sentence "*Conservative members of Congress are*

taking action against same-sex marriage in DC" may be represented by the following feature vector:

F0 = {'conservati': true, 'member': true, 'congr': true, 'tak': true, 'action': true, 'against': true, 'same-sex': true, 'marri': true}.

Here we represent the feature vector as a python dictionary; NLTK, for example, uses this representation of a feature vector. This would be a satisfactory representation if that single sentence was only one in the whole tweets. If we want to be able to represent other tweets, for example "*Churches are against gay marriage*", the previous feature vector would not be a good representative. It is thus required to extend the set of words, and incorporate them as the features in the feature vector. The set of features in this case would be

F0 = {'conservati', 'member', 'congr', 'tak', 'action', 'against', 'same-sex', 'marri', 'church', 'gay' }.

Feature vectors that fully represent both sentences would be (for the first sentence; similar feature vector is created for the second sentence):

F1 = {'conservati': 1, 'member': 1, 'congr': 1, 'tak': 1, 'action': 1, 'same-sex': 0, 'marri': 0, 'against': 1}.

Only some of the words appear in both sentences, and they are used for expressing the similarity between the sentences. Obviously, for any real use, the feature vector would have to contain a much larger number of words. We will explore some of the choices for selection of words that are suitable for sentiment analysis.

It is possible to register either the presence of the word appearance in some text, or the frequency - the number of times the word appeared. The frequency in the feature vector for sentence "*I am really really not in favor of same-sex marriage*" for word "really" would have value 2 (number of word appearances.) This may indicate the extent of the sentence polarity (positive or negative or neutral) on a finer grained scale. However, since we compare single sentences, it not very common to have one word appearing multiple times. Furthermore, previous researches have shown that for sentiment analysis the mere presence or absence of the word has the same performance as the more detailed frequency information. For that reason, we have chosen the appearance of the word as feature vector values in our experiment. Ideal bag-of-words feature vector would contain all the words that exist in the language [29].

3.3 Machine Learning

The process of machine learning studies patterns in data. From the literature review we already know two common approaches of machine learning: unsupervised and supervised. We considered one unsupervised (k-means clustering) and two supervised (Naive Bayes, SVM) classification algorithms. All three algorithms are available in NLTK framework. Implementation details are explained in the following subsection.

We extracted the relevant tweets from processed dataset and carried out the experiment with text preprocessing activities for both the approach. The keywords used to extract relevant tweets were '%marriage%', '%same sex%', '%same-sex%', '%equal right%'. The keywords are based on the words which were considered the most

appropriate by our domain expert. Table 4 shows the statistics of relevant tweets for the study.

Table 4. Relevant tweets for the study

Total number tweets used for this study	2,000,000
Relevant tweets	37,247
Date range in relevant tweets	03-13-2008 to 07-23-2012
Number of relevant tweets with URLs	26,803

3.3.1 Unsupervised Clustering

The purpose of performing unsupervised clustering was to understand our data. The aim of this approach was to find cluster of tweets that are somehow similar in characteristics. Clustering was useful for exploring our data as there were many cases and no obvious groupings in our study. It also served as a useful data-preprocessing step to identify homogeneous groups. In this approach, we go over common ways of preprocessing the text which has been described in section 3.2.1.

The clustering algorithm we used was to find the natural clusters in the data, by calculating the distance from the centers of the clusters. The position of centers is iteratively changed until the distances between all the points are minimal. The algorithm accepts the number of clusters to group data into, and the dataset to cluster as input values. It then creates the first *K* initial clusters (K= number of clusters needed) from the dataset by choosing *K* rows of random tweets from the dataset. In this study, we selected all relevant tweets (37,247) as a dataset to form 2 clusters. We picked 2 clusters because

the aim was to explore how the clustering algorithm groups the tweets. This would give us a greater insight into the characteristics of the tweets.

Next, the algorithm calculates the arithmetic mean of each cluster formed in the dataset. The arithmetic mean of a cluster is the mean of all the individual tweets in the cluster. Each tweet is assigned to the nearest cluster (the cluster which it is most similar to) using a measure of distance or similarity like the Euclidean distance measure. It re-assigns each tweet in the dataset to the most similar cluster and re-calculates the arithmetic mean of all the clusters in the dataset. This new arithmetic mean becomes the center of this new cluster. Following the same procedure, new cluster centers are formed for all the existing clusters.

The algorithm re-assigns each tweet in the dataset to only one of the new clusters formed. A tweet is assigned to the nearest cluster (the cluster which it is most similar to) using a measure of distance or similarity. The preceding steps are repeated until stable clusters are formed and the clustering procedure is completed. Stable clusters are formed when new iterations or repetitions of the clustering algorithm does not create new clusters as the cluster center or arithmetic mean of each cluster formed is the same as the old cluster center.

3.3.2 Supervised Classification

In the terminology of machine learning, classification is considered an instance of supervised learning. In supervised classification approach, the classifier is trained on training dataset and then used to classify a testing dataset. The labels we are interested in this study is whether the tweet expresses opinion (positive or negative) or not (i.e., the

tweet is neutral). In this approach, we go over common ways of preprocessing the text which has been described in section 3.2.1.

Manual Coding: To obtain a training dataset, two coders manually labeled several hundred tweets to obtain 100 opinionated and 100 neutral tweets. Table 5 shows few examples used for this study along with their labels. An iterative process of coding, inspection, discussion, and revision was carried out to inductively learn how the indicators (labels) of the relevant opinion evidenced themselves in the data, until the coders reached a solid labeling strategy. The human coded data were used as the "gold standard" to train and to assess the performance of the supervised algorithms.

Table 5. Examples of tweets used for classification

Label	Tweet
opinionated (positive or negative)	*1. I love how my mommy supports gay marriage :)(positive)* *2. Catholic church has a majority of homosexuals in its hierarchy and ranks. They are among the ultra hypocrites denouncing gay marriage. (positive)* *3. Gain back gay marriage in California I think you should lose your marriage right now because you don't deserve it. You couples disgust me. (negative)*
neutral	*1. Analyst: Marriage Equality May Soon Trump Anti-Gay Amendments Analyst: Marriage Equality May Soon Trump Anti-Gay Amendments :: EDGE New York City* *2. Bachmann campaign staffer says gay marriage will lead to a woman marrying the Eiffel Tower* *3. Is President Obama back pedaling on his stance against gay marriage?*

3.4 Evaluation

In order to evaluate the effectiveness of the feature extraction and classification algorithm we conducted several experiments.

We implemented a Python program that performs the experiments and handles the results. It uses algorithms and metrics implemented in NLTK2.0. Specifically, we used packages nltk.cluster, nltk.classify, nltk.metrics. We also used numerical python library *numpy* for parsing of command line parameters. Additionally, we used *scipy* and *matplotlib* package for scientific calculations. All our experiments were carried out on Ubuntu Linux 12.0 installed on a virtual machine.

We used K-means algorithm to perform clustering. The centers are initially randomly assigned. We decided to have 2 random clusters for this study based on our hypothesis. This was based on the assumption that we will have 2 labeled classifications in supervised classification. We repeated the algorithm on the same data multiple times to have better understating of the clusters. We have repeated the procedure 10 times in our experiments. We have used Euclidean distance as dissimilarity metric between feature vectors.

We used Naïve Bayes and SVM classifiers to perform the classification tasks. We used bigram and bag-of-words model for the feature extraction as mentioned in section 3.2.3 where every bigram feature name with a value of ‘true’. Each classifier was provided a list of tokens in the form of [(feats, label)] where feats is a feature dictionary and label is the classification label. In our case, feats were of the form {bigram feature: True} and label were 1 for opinionated tweets and 0 for neutral tweets. For accuracy

evaluation, we used package nltk.classify.util.accuracy with training and testing data. We used 90% training and 10% testing split to conduct the experiments.

3.4.1 Formative

Formative evaluation was conducted at all stages to ensure high inter-rate reliability in manual coding and high accuracy of machine learning techniques. We selected accuracy as our metric in this activity. We iteratively tweaked the algorithms and features for the better accuracy during this activity.

3.4.2 Summative

Summative evaluation involved thorough manual understating and analysis of the results. We used other useful metrics like precision and recall in addition to accuracy in this stage. These metrics can provide much greater insight into the performance characteristics of a binary classifier.

We used precision to measure the exactness of the classifier. A higher precision means less false positives, while a lower precision means more false positives. This is often at odds with recall, as an easy way to improve precision is to decrease recall.

$$\text{Precision} = \frac{tp}{tp + fp}$$

Where *tp* = true positive, *fp* = false positive.

We used recall to measure the completeness, or sensitivity, of the classifier. Higher recall means less false negatives, while lower recall means more false negatives.

Improving recall can often decrease precision because it gets increasingly harder to be precise as the sample space increases.

$$\text{Recall} = \frac{tp}{tp + fn}$$

Where *tp* = true positive, *fn* = false negative.

Precision and recall can be combined to produce a single metric known as F-measure, which is the weighted harmonic mean of precision and recall to facilitate comparison. We found F-measure to be about as useful as accuracy.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

F-measure is a measure of a test's accuracy. It considers both the precision and the recall of the test to compute the score. On the other hand, accuracy is the proportion of true results (both true positives and true negatives) in the dataset. An accuracy of 100% means that the measured values are exactly the same as the given values.

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

Where *tp* = true positive, *tn*= true negative, *fp*= false positive, *fn* = false negative.

4. EXPERIMENTAL RESULTS

In this section, we present the results for both formative and summative evaluation activities.

4.1 Formative Evaluation Results

In the formative evaluation stage, we evaluated the unsupervised clustering algorithm. As per our input, 2 clusters were formed based on the similarity criterion which is distance. We saw two or more tweets belong to the same cluster if they are "close" according to a given distance. The two clusters were not ideal grouping of tweets, they comprised of both opinionated and neutral tweets in both the clusters. Table 6 shows results of clusters formed. Our goal was to use this as a formative step to understand our data for supervised classification.

Table 6. Results of clustering

Cluster	Tweet
1	*a) #CNN.s Kyra Phillips Grills Anti-Marriage Equality David Tyree: <<URL>> (Neutral)* *b) I Love my #Gays! #Gaymarriage <<URL>> (Positive)* *c) Excellent perspective: Social Justice: Is Marriage Equality a Civil Right? <<title>> (Neutral)*
2	*a) wow! Thank you Mr. President. BREAKING: Obama Embraces Marriage Equality <<URL>> via @thinkprogress (Positive)* *b) Gay marriage is something we shouldn.t even have to vote on. It should be a basic right. #VoteNoOn1 (Positive)* *c) #gaymarriage news Rethinking Gay Marriage - New York Times <<URL>> (Neutral)*

We evaluated the supervised classification algorithms and found some important features which were contributed to the accuracy of the classifier. Figure 4 shows 10 most informative features, most of the features are highly descriptive adjectives. The one word that seems bit odd is "take". This kind of information helped us to tweak the algorithms and features iteratively for the better accuracy.

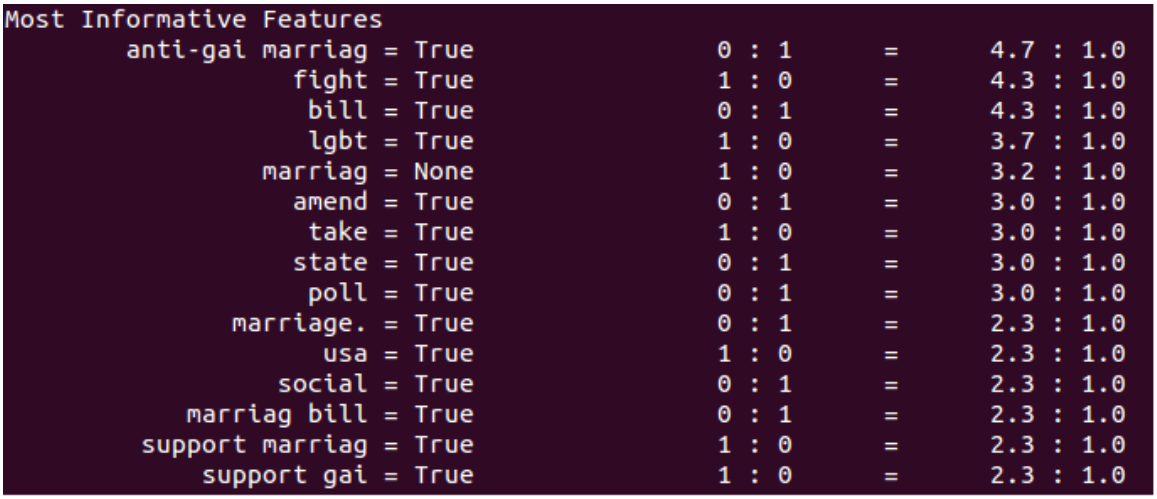

Figure 4. Most informative features for Naïve Bayes classifier

4.2 Summative Evaluation Results

In summative evaluation stage, we evaluated only the Naïve Bayes and SVM classifiers and did not evaluate the clustering algorithm. Table 7 tabulates summative evaluation results for the classifiers.

Table 7. Summative evaluation results for the classifiers

Classifier	Accuracy	Precision		Recall		F-measure	
		Opinionated	Neutral	Opinionated	Neutral	Opinionated	Neutral
Naïve Bayes	90%	83%	100%	100%	80%	91%	89%
SVM	55%	100%	53%	10%	100%	18%	69%

Looking at the number for the Naïve Bayes classifier:

1. 90% accuracy indicates proximity of measurement results to the true value.
2. If a tweet given an opinionated classification is only 83% likely to be correct. Not so good precision leads to 17% false positives for the opinion label.
3. Any tweet that is identified as neutral is 100% likely to be correct (high precision). This means no false positives for the neutral class.
4. Nearly every tweet that is opinionated is correctly identified as such, with 100% recall. This means no false negatives in the opinion class.
5. Few tweets that are neutral are incorrectly classified. Low recall causes 20% false negatives for the neutral label.

There were very few incorrectly classified tweets with Naïve Bayes classifier, for example, *"@GayMarriageWORD yes, and know gay marriage is a constitutional right.* ***Take*** *that religious conservatives!"*. The reason we found was the word 'Take' was contributing to most informative features as we discussed in the Section 4.1 but it was not a descriptive adjective pertaining to our study.

We ran an SVM on the combined features but found that this did not yield a better performance than Naïve Bayes. Kim's [31] study have shown multitude of length-based, lexical, POS-count, issue-aspect-mention count, and metadata features are most effective when utilizing SVM. But we used only lexical, bigrams and bag-of-words features in this study. Since many of the other features used in NLP studies are not available to us due to the short size of twitter messages.

The features which we have used in this research are pertaining to content-focused approach. Other features we could have used in both content-focused approaches and user-focused approaches are content-based, graph-based, and individual-based features [32]. If we had enough coded training set, we could have used k-fold cross validation techniques for better performance.

5. CONCLUSION AND FUTURE WORK

Overall, the success and failures of all these different approaches gave us a good overall picture of the challenges of opinion mining. In this study, we have collected, stored and analyzed the opinion of social network comments. We evaluated the fitness of different feature extraction and learning algorithms (supervised and unsupervised) on the classification of text according to their opinion about the issue. We have tested the impact of bi-gram and bag-of-words model on the classifier's performance. In addition, we looked into the influence of feature vector on the classification accuracy and other metrics. Good performance was achieved using Naïve Bayes classifier over the SVM classifier; however, SVM requires more sophisticated feature selection techniques that might improve its performance.

However, the study also has a few limitations. The experiments were not carried out on the entire data set, so there are possibilities where we have selected biased user tweets for our study. It is also possible that the initial seed set may have been biased.

In addition to the areas mentioned in the previous sections, there are several potential areas for future work:

1. We could benefit from performing experiments using a larger labeled data set using our techniques.
2. We should use of inter-rater reliability process in coding.
3. Ensemble methods: An individual classifier may not be the best approach. It would be interesting to see what the results are for combining different classifiers.

4. Use of part of Speech (POS) tagger: A POS tagger can be used to look at adjectives to determine if a tweet contains opinions.
5. Using the above improvements, we plan to classify tweets into positive, negative, and neutral rather than just opinionated and neutral.

BIBLIOGRAPHY

[1] K. Z. Mary Madden, "65% of online adults use social networking sites," *Pew Internet*. [Online]. Available: http://pewinternet.org/Reports/2011/Social-Networking-Sites.aspx.

[2] B. Krishnamurthy, "A Measure of Online Social Networks," *1st international Conference on Communication Systems and Networks Bangalore, India*, 2009.

[3] a. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, Jul. 1959.

[4] T. Mitchell, "Machine Learning," *McGraw Hill*, 1997. [Online]. Available: http://en.wikipedia.org/wiki/Machine_learning#cite_note-3.

[5] A. Ng, "Machine Learning." [Online]. Available: https://www.coursera.org/course/ml.

[6] Netflixprize, "Netflixprize," 2009. [Online]. Available: http://www.netflixprize.com//index.

[7] Demis Hassabis, "Combining systems neuroscience and machine learning: a new approach to AGI," 2010. [Online]. Available: http://nextbigfuture.com/2010/08/combining-systems-neuroscience-and.html.

[8] "Machine Learning." [Online]. Available: http://en.wikipedia.org/wiki/Machine_learning.

[9] Margaret Rouse, "Natural Language Processing," *TechTarget*, 2011. [Online]. Available: http://searchcontentmanagement.techtarget.com/definition/natural-language-processing-NLP.

[10] R. Bergmair, "Natural Language Steganography and an 'AI-complete' Security Primitive," *Proceedings of the 7th Information Security Conference*, 2004.

[11] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[12] R. Picard, "Affective Computing," *MIT Press*, 1997.

[13] Bo Pang1 and Lillian Lee, "Opinion Mining and Sentiment Analysis," *Foundations and Trends® in Information Retrieval*, 2006.

[14] Michelle de Haaff, “Sentiment Analysis, Hard But Worth It!,” *CustomerThink, retrieved*, 2010.

[15] B. Pang and L. Lee, “Opinion Mining and Sentiment Analysis,” *Foundations and Trends in Information Retrieval*, vol. 2, no. 1–2 (2008), pp. 1–135, 2008.

[16] A. C. B. Jansen, M. Zhang, K. Sobel, “The Commercial Impact of Social Mediating Technologies: Micro-blogging as Online Word-of-Mouth Branding,” *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems (pp. 3859-3864). ACM*, 2009.

[17] P. Turney., “Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised,” *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics*, 2002.

[18] B. Pang and L. Lee, “A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts,” *Proceedings of Association for Computational Linguistics*, 2004.

[19] R. Hsu and A. Wu, “Machine Learning for Sentiment Analysis on the Experience Project.”

[20] “Naive Bayes Algorithm.” [Online]. Available: http://nltk.googlecode.com/svn/trunk/doc/api/nltk.classify.naivebayes-module.html.

[21] H. Zhang, “The optimality of Naive Bayes,” *Proceedings International Florida Artificial Intelligence Research Society Conference*, vol. A A, no. 1(2), p. 3, 2004.

[22] Jonathon Read, “Using emoticons to reduce dependency in machine learning techniques for sentiment classification,” *In ACL. The Association for Computer Linguistics*, 2005.

[23] K. Skemp, “All A-Twitter about the Massachusetts Senate Primary,” *Retrieved*, 2009. [Online]. Available: http://bostonist.com/2009/12/01/ massachusetts senate primary debate twitter.php.

[24] D. D. Perlmutter, “Political Blogging and Campaign 2008,” *A Roundtable. The Press/Politics*, vol. 13(2), pp. 160–170, 2008.

[25] “Tfidf.” [Online]. Available: http://en.wikipedia.org/wiki/Tf-idf.

[26] S. V. B. Pang, L. Lee, “Thumbs up? Sentiment Classification using Machine Learning Techniques,” *Association for Computational Linguistics*, 2002.

[27] Twitter Dev Team, “Twitter Rate Limiting.” [Online]. Available: https://dev.twitter.com/docs/rate-limiting.

[28] S. E. K. E. L. Bird, *Natural Language Processing with Python*. O’Reilly Media Inc, 2009.

[29] K. Yessenov and S. Misailovic, “Sentiment Analysis of Movie Review Comments,” *Methodology*, pp. 1–17, 2009.

[30] C. Byun, H. Lee, and Y. Kim, “Automated Twitter Data Collecting Tool for Data Mining in Social Network,” *Research in Applied Computation Symposium (RACS)*, 2012.

[31] S. Kim and P. Pantel, “Automatically assessing review helpfulness,” *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, no. July, pp. 423–430, 2006.

[32] S. Zeng S. Kaza H. Chen, “Expertise Evaluation in Online Communities using Time-Series Models.”

CURRICULUM VITA

RANJAN M VAIDYANATHAKUMAR

SUMMARY

- Data Analyst having experience to acquire, manage, manipulate, analyze data and report results.
- Systems Analyst with proven ability to research problems, plan solutions, recommending software and systems, and coordinating development to meet business requirements. Deep understating of SDLC.
- Over 6 years of IT industry, educational institution experience in building wide range of applications, distributed databases and data warehouse systems including analysis, design, development, configuration, customization, deployment and maintenance.
- Proven ability to lead and motivate high performance teams. Extensive client facing and team management skills.

EDUCATION

Master of Science, Computer Science — January 2013
Towson University, Towson, MD, USA

Bachelor of Engineering, Industrial Engineering and Management — September 2005
Visveswaraiah Technological University, Belgaum, India

Software Diploma, Graduate in Software Architecture — December 2005
NIIT, Bangalore, India

COMPUTER SCIENCE RELATED EXPERIENCE

- Graduate Assistant, Towson University, MD, USA
- Internship with startup, Biosaic, LLC
- Research Assistant, Towson University, MD, USA
- Sr. IT Consultant - Technical Lead, Fujitsu Consulting India, Bangalore, India
- IT Consultant - Software Engineer, Oracle Corporation, Bangalore, India
- Programmer Analyst, YS Technocrats Limited, Bangalore, India

KEY TECHNICAL SKILLS

Databases : Oracle 9i/10g/11i, MySql, MS SQL Server 2005/2008, MongoDB
Data warehouse : Data Warehouse Architecture
Data Migration and Integration: ETLs in Informatica Power Center 8.6,
BI and Analytics : Oracle BI Apps, OBIEE plus 10.x, 11.x
Data Mining : Weka, Oracle Data Mining, Text Mining, Machine Learning, Sentiment Analysis, NLTK
Web/App Servers : Apache, Oracle Weblogic
Languages : Ruby, Java, PHP, PL/SQL, UNIX Shell Script, JSON, Python
Mobile Apps : Android 2.2, Augmented Reality
Frameworks : Rails, Oracle Apex 3.4, struts, Oracle Portal 9i,
IDE : Eclipse-Indigo/Helios, SQL Developer 11g, Toad

Admin Activities : Install and configure above mentioned technologies, SVN, ghosting, and cloud computing.

KEY PROJECTS SUMMARY

Project Name: HydroCloud: Integrative tool for hydrologic data
Role : Graduate Assistant
Responsibility: Provide expertise to acquire and manage the geospatial data.

- Objective of the project was to integrate current hydrologic research with advances in distributed database and data warehousing systems. The goal was to solve some of the technical challenges in using hydrologic data by applying emerging technology in spatio-temporal data warehouses and cloud computing. Involved in designing a prototype, distributed, cloud-based system architecture for the storage and retrieval of multi-scale spatial and temporal hydrologic data.

Project Name: biosaic.com, a social networking site
Role : Systems Analyst.
Responsibility: Consulted management to determine new IT system for a startup.

- Researched emerging technologies to decide which technology suits better for their business needs.
- Make design, implementation decisions to come up with prototype, layout a plan to approach angel investors or venture capitalists for funding.
- Develop and implement ruby and ruby on rails software application. Identify risks and dependencies within the schedule. Overseeing installation and configuration of the new system to customize it for the organization

Project Name: Augmented Reality Application on Android platform
Role : Independent Researcher
Responsibility: Developed an augmented reality application solution for the Android Smartphone. As part of the solution I used mixare, open source mix Augmented Reality Engine. The engine potentially uses the phone's various sensors (camera, accelerometers, GPS, and digital compass) to locate the user and display contextual data about his surroundings over the image received from the camera.

- We delivered the complete solution where in Android application would query the remote server for nearby points of interest, and display them overlaid on live images from the camera, based on whether or not the phone is pointing in the direction of the point of interest. The remote server is accessed by the phone using HTTP requests sent to the server and database. JSON is used for data interchange to communicate with the mobile application.