Real-Time Sentiment Analysis in a Chat Application

Deep Ramesh Katariya
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14623
drk8921@cs.rit.edu

Abstract—The world in recent years has seen a steady rise in the number of users who are getting connected to the Internet. With mobile technology becoming cheaper, affordable, and essential, the number of people who now own such devices is increasing which again is a reason more and more people are gaining access to the web. Internet nowadays is like a doorstep solution to almost every trivial problem that one might face. We, as the users of Internet resort to the web for many things such as looking through a product’s review before buying it, posting feedback about a brand/product so that others can refer to it, posting tweets on social media about a subject and so on. This led to emergence of Sentiment Analysis which is concerned with mining the sentiments in the opinions expressed on social media by people. Sentiment Analysis has been found useful and employed by lots of companies and is now a major area of research. Lots of emphasis has been placed on the applications of Sentiment Analysis and how it can be leveraged into decision making. However not much work has been done in the past pertaining to Real-time sentiment analysis and its applications. In this project we have attempted to detect user’s mood by employing Sentiment Analysis. In this report, we will go through the entire process of how we performed real-time sentiment analysis in a chat application by detecting user’s mood based on his/her messages.

Index Terms—Real-Time Sentiment Analysis; Natural Language Processing; Mood Detection; Machine Learning

I. INTRODUCTION

With technology advancing day by day, many of the devices such as mobiles, laptops, computers have become affordable and easy to access. People have started making the most of these available technologies. Talking about a mobile phone, texting and calling are the most commonly used features. Researchers have also shown that, most people prefer to get the job done by texting instead of calling. We have several messenger applications available in the market these days. Lots of new features are being incorporated in these applications. While pondering over these features, I realized that none of the available messenger applications have a feature that lets us detect the mood of the other person while talking to them. Even though it is possible to determine the other person’s mood just by talking to them or after exchanging a few messages, such a feature could be handy and a stepping stone towards the development of predictive texting. Development of such a feature could also be used to demonstrate how Sentiment Analysis can be performed in real time. Also, not much work has been done in the field of Real-Time Sentiment Analysis in the past and this could be good opportunity to tap in an explored field.

Based on the above mentioned thought process, the idea for this project was decided in an attempt to demonstrate how we can perform Real-Time Sentiment Analysis to detect a user’s mood in a chat application by analyzing and processing incoming messages from that user.

The flow of this report is as follows: Section II briefs about the overview of Sentiment Analysis, its applications, and the literature review of the works done in past. Section III talks about the implementation details and explains the overall system prototype architecture and the tools used for the development of the application. Section IV focuses on the results, issues faced, and findings from the project. Section V concludes the report and talks about possible future work for the project.

II. BACKGROUND

A. Sentiment Analysis Overview

Sentiment Analysis is process of identifying and categorizing opinions in a piece of text to determine the polarity of the opinion and thus determine how the author feels about a specific subject. Also know as Opinion Mining, it is basically used to determine what the author’s feelings towards the subject are using their comments/feedbacks. In it, texts are classified using their polarity as negative, positive, and neutral. Sentiment Analysis has found applications in a lot of industries with increasing number of industries finding it useful. Some of the areas where it is being used extensively are opinion mining on social media, used by companies to evaluate their customer’s experience with the company and analyzing customer reviews and feedbacks gathered from all the relevant sources of social media.

Moving ahead to the applications of Sentiment Analysis, it has been proved by researchers [3] that during elections, the tweets and posts concerned with the elections are very much related to the actual election results. This is a strong example to understand how rich and important source of information, content on the social media can be. Using the tweets as the source data, these days, many companies are investing more time in analyzing how people view their brand or company. It makes sense because the amount of people using social media to voice their opinions is increasing day by day and voicing opinions online is a very convenient and easy task.
Anyone can go online and post reviews of a product that they purchased and share their opinion about it. Such people would then go and post their experience with the product. Also, there are people who spend time to review a product and then compile a review consisting of good and bad points about that product. In such cases, performing Opinion Mining on such reviews can help the parent company of the product to understand what the customers like and do not like about their product. This helps the company in decision making and at the same time help them to market similar products to the same customers who have something good to say about another product. This brings us to another important application of Sentiment Analysis which is Recommender Systems.

Sentiment Analysis extensively benefits Recommender Systems. Let us understand this with an example. Customer A buys a home theater system. He then posts a review of the product online where he talks about some specific features. These features act as very strong points for marketing other products with similar features to Customer A. Now there is another Customer B who has bought some different products and mentioned about its features. These features have a lot in common with Customer A's product and have a high sentiment score as well. In this scenario, Customer A can be recommended the product which Customer B had purchased and vice versa. However, one caveat in with the use of Sentiment Analysis in Recommender systems is that many times, the reviews are very short and do not cover all the aspects of the product. Many of the reviews are spam and are of very little or no value to the analysis. Working with such reviews is difficult and does not yield good results and requires more processing.

B. Literature survey

Lots of researches have been conducted in the field of Sentiment Analysis and Natural Language Processing using different approaches. In this section we will go through some of the past researches that are relevant to this project. In [1] the researchers have followed an approach which is somewhat different from the conventional techniques. Usually it is said that, analyzing a large dataset yields lots of useful information about it but in this paper, the authors have experimented with less data but with more precise information. They have explored the effectiveness of performing Real-Time Sentiment Analysis approaches on small datasets. The authors in [1] have defined Sentiment Analysis as the process of extracting sentiments of the users about a particular topic or an aggregate of sentiments of a group of people towards a particular topic using Natural Language Processing and data mining techniques. As a part of the research they have used Lexicon Based Sentiment Analysis experiments on small Twitter datasets. These datasets are real time dynamic datasets and comprise of the most popular tweets. They have compared the results of the process with using larger datasets. To evaluate the accuracy of the system, human assistance was required to manually extract sentiments from the piece of texts. The results of analyzing small datasets were compared to the results of using the Sentiment140 [11] dataset. Finally the authors concluded that the results were consistent and that it is beneficial to use small but relevant datasets instead of large datasets. The approach used in [1] is similar to some extent to the approach used in our project.

The researchers in [2] on the other hand have surveyed Sentiment Analysis with respect to its various machine learning methods. They have explained various Machine Learning methods of Sentiment Analysis in brief. Some of the very popular methods are as follows - Support Vector Machine, Naïve Bayes, Maximum Entropy and Artificial Neural Networks. As we move ahead in paper [2], the authors have stated three major methods of Sentiment Analysis as Machine Learning, Dictionary Based and Hybrid. The authors went ahead and gave an overview of these methods. Dictionary based methods include corpus based methods where the text is processed and analyzed word by word by accounting the individual weight of the words present in the dictionary. The words in the sentence are matched to see if they are present in the dictionary. Machine Learning methods on the other hand depend on the popular machine learning methods. The authors then state the reason for their popularity as their high precision. These methods have an advantage over the dictionary based method because there is huge variation in text semantics due to which the error in the dictionary based methods is very large. Towards the end of the paper [2] the authors have stated some challenges in the field of Sentiment Analysis. They have expressed concern over the fact that most methods these days can be used with English language alone. The support for most of the other languages is very less. They are of the opinion that there is lot of information which can be gained which is present in languages other than English as well. To conclude the authors state that Sentiment Analysis is becoming more and more challenging with more and more problems being discovered to solve.

The aim of the researchers in [3] is somewhat similar to our project in one aspect. In [3] researchers are attempting to use Real-time streaming Twitter data to perform Sentiment Analysis about a top trending topic. They have then aimed at representing the evaluation results in terms of 10 major emotions. Also mentioned in [2] the authors [3] state that the accuracy of the model that they have created highly depends on the presence of words from real time data in the lexicon. This seems like one drawback of using Lexicon based approaches. Finally the authors state that one way of coping up with this drawback is to use a combination of Machine Learning methods with Lexicon based approaches.

In [4] the authors have explored the Customer Service domain where Sentiment Analysis can be used to mine the minds of the customers. The authors have aimed at determining relation between sentiment and moods exhibited by online conversations and actual level of satisfaction in the customers. They have shown that other factors such as length of the conversation, time required to respond to the customer and so on are weak indicators of satisfaction. The research [4] was aimed at helping the customer service agents to know
beforehand the factors that contribute towards customer satisfaction so that the service agents can identify the drawbacks in the existing process and improve in those areas to provide a better service experience to the customers. This in a way helps them to understand that the conventional factors of satisfaction such as responding quickly, make the customers wait less in a queue, etc., are not that important as they used to be. The researchers stated that the chat logs are available immediately after an online chat session with a customer ends. These chat logs are available for analysis immediately and can help in faster decision making to improve the process. Many of the things discussed in [4] are relatable with us in real life. For instance, after talking to a customer service agent on a telephonic call or an online chat session, many people do not give any feedback about their experience. And the ones who give are on the extremes such as ‘extremely satisfied’ or ‘extremely dissatisfied’. The authors state that, at such times, it is very helpful to go through such chat logs to determine certain things such as what was the topic of concern in the discussion, what were the sentiments exhibited by the customer and the agent, and finally overall satisfaction level of the customer. Supported by their findings, the researchers [4] stated that the sentiments of the service agents also played an important role in determining the satisfaction of the customers. Finally the authors concluded by saying that the lexicon that they used in their research focused on generic online chat logs. The performance and effectiveness of their research could have been better had there been a lexicon specific to the customer service domain.

The topic of research in [5] is completely different from rest of the researches that we have seen so far. It is true that most of the work in the domain of Sentiment Analysis has been related to textual data only. The author [5] states that even though we find lots of multimedia information on social media these days, very little research has been done in this area. These days people share images along with textual data on social media. The author [5] also states that images and textual data come hand in hand on social media these days. The author is of the opinion that Sentiment Analysis using visual data is a challenging field and encourages more research work to be done in this area.

In [6] the researchers devised a novel idea of developing an interactive Social-TV chat system where user’s can chat with each other while watching a show. The benefit of such system is that user’s can share their views about a particular show right when they are watching. At any given point of time there will be many people watching the same show and sharing their views. The authors were of the opinion that using such data it will be very easy to plot a graph to display the live show rating which will be helpful to people who are just browsing through the shows to decide which one to watch. Also, the authors were of the opinion that having a live show rating is more accurate than many of the conventional online show rating portals. This research aligns with the goal of our project in one aspect which is performing Real-Time Sentiment Analysis on online conversations. However, it is focussed on the subject being discussed which is not the case in our project. Our project is focussed on determining the user’s mood and feelings in general and not with respect to a subject.

In [7], the researchers felt the need to experiment with various combinations of Sentiment Analysis methods, Machine Learning techniques and features for analyzing real time students feedback about a particular topic. The authors mentioned that no work done was done in this field and that it was a good opportunity to explore this domain. In their findings the authors stated that the variation in results was large when they were experimenting with different methods. Finally, the authors concluded by saying that experimenting with different pre-processing techniques and level yielded varying results.

Similar to many of the previous researches, in [8] the authors have used Naive Bayes classifier to perform Sentiment Analysis on real-time Twitter data. Naive Bayes was implemented with Sentiment140 [11] training data which is sourced from Twitter tweets itself. The concluding opinion of the authors was that using SentiWordNet [10] along with Naive Bayes could result to a better performance of the classifier.

III. METHODOLOGY

A. Architecture

The architecture of the chat application consists of two major modules:

- Chat Application
- Sentiment Analysis Module

The goal of developing this chat application is to leverage Sentiment Analysis to detect the user’s mood by analyzing their messages. Consider two users, A and B chatting with each other. Once a proper network connection is established, both the users will get a notification on their chat box informing them about the connection. One of the user’s system behaves as a client and the other one as a server. Then User A sends the first message which will be transported over the network to the client application running on User B’s system. As soon as the message is received it is then processed by the Sentiment Analysis module on User B’s system to check for its polarity which after processing returns a score to the chat application. The score is then processed in the Chat Application module on User B’s system using a rule-based logic where actual mood detection process takes place. Using a pre-defined range of score values, it is decided what the current mood of the user on the other side can be. The original message along with the detected mood is then displayed on the chat window GUI for the User B. This same process is followed for every message received on both the user systems if both the users are online and connected. We will walk through both the modules along with their implementation details in the following subsection.

B. Implementation

We will walk through the implementation details in three phases

- Chat Application Module

Rochester Institute of Technology
1) Chat Application Module: First step towards development of the entire application was to develop the chat application. This entire application was developed solely using Python 3.4. I started working on developing a chat application that would run entirely on the Python console not requiring any GUI. After researching on working with Sockets in Python, I was then able to get the console-based chat application up and running. Python Sockets and Threads were the two main components used in the development of this module. Once all the error handling was done in this version of the application, I then started researching on various ways to create a GUI in Python. Python has a very rich library support for application development, be it frontend or backend. I came across a number of Python libraries such as Tkinter, Qt, GTK+, Kivy, FLTK, and OpenGL when researching for GUI development in Python. The next step was to get a brief understanding on all these libraries and shortlist the one that would be quick and easy to learn. After researching on the libraries, I zeroed in on using Tkinter as there were many web tutorials available online and some very good tools that would be helpful in development of GUI. Coming across PAGE - Python Automatic GUI Generator [12] was of great help. It is a drag and drop tool which let’s us create a GUI quickly and generates a boilerplate code to be used in our application. After I had the initial design for the chat window ready, using PAGE [12] I quickly had the chat window GUI ready and running. Next step was to alter the console based chat application to work with a GUI now. Since, Tkinter was easy to learn, working on this phase was quick and easy. Soon, I had a chat application ready and running with a design similar to many of the messenger applications that we use. One thing that separates the GUI design for this application with other messenger applications in the market is the presence of a display section which changes its color based on the chat rating which reflects the other user’s mood and another section which displays the rating for the most recent message received. At that time these two sections would only display dummy values so the next step was to develop the sentiment analysis module. After the completion of this phase, we had a working prototype of a chat application with the basic functionality of being able to send and receive messages. However, one major caveat here was that the application would only run on local network. It means that both the client and server need to be on the same local network to be able to connect and communicate. This hurdle was caused by the fact that port forwarding is locked on most of the networks and it is not possible to alter the network settings without proper authorization and permission. This is done to ensure that the internal network is protected by a firewall which will prevent any unauthorized outside client to gain access. Since, this application is just a prototype, I accepted it to work only on a local network and proceeded on to the next phase.

2) Sentiment Analysis Module: This was the second phase in the development process. After having a functioning chat application, I had to work on developing the Sentiment Analysis module which would then integrate with the chat application and make mood detection possible. The decision of leveraging the NLTK - Natural Language Toolkit [13] library in Python was made during the planning and research phase of the project.

Since the goal of the project is to use sentiment Analysis to detect user’s mood by extracting sentiments from their messages, there was a need for dataset that would be used in training the model to be used in the process. It was very difficult to obtain a dataset containing chat logs. I was able to source some datasets that contained chat logs but using it would attract ethical and privacy issues which is why I decided to use the sentence polarity corpus available from the NLTK library. As previously seen in literature survey, there are many different approaches in performing Sentiment Analysis. Considering the scope and nature of this project, lexicon-based approach was decided to be used.

The corpus that was decided to be used in training the model was sourced from the sentence polarity corpus in NLTK [13]. This corpus contains sentences having negative and positive polarity scores which were used to train the model. The model was then evaluated using the Naive Bayes classifier which is also included in the NLTK [13] library. Equal number of sentences with positive and negative polarity were used in training and testing the model. We shall discuss more about the results of the evaluation ahead in the paper in the results.
3) Integration and Testing: This was the final phase of developing the entire application. At this point of time, I had the Sentiment Analysis Module and the Chat Application Module ready. The next step was to integrate the Sentiment Analysis Module into the Chat Application Module. This was a fairly simple task where both modules had to be binded together. This was achieved by making sure that whenever a message is received at either the client end or server end, it shall be decoded and passed to the Sentiment Analysis module at that end. The Sentiment Analysis module would then return a polarity score that would be used by the application logic to detect the sentiment expressed in the message and thus determine the mood of message sender. This is when the received message will be displayed on the receiver’s chat window. Along with the received message, the color of the overall chat rating field and the current message rating field will change reflecting the mood of the user. The color indicators that have been used to reflect moods are yellow for neutral, green for good, dark green for very good, red for bad, and dark red for very bad. A case of color blindness has also been taken care of by displaying the mood of the user along with the color. For colorblind people, they can read the text reflecting the user mood otherwise the color indicator will suffice. Two main metrics that have been devised and used in the project are overall chat rating and individual message rating.

The next step after completion of the integration phase was to test the application. The application was tested with respect to functionality and performance. All kinds of scenarios were considered to check if the application was fault tolerant. For instance, if a client left the conversation or disconnected due to a network problem or the server did the same, the application would not crash due to network error. Instead the application would wait for the client to reconnect. However, the scores that were calculated till that time will not be retrieved and will be reset. Also, there was almost no latency even after integrating the Sentiment Analysis Module with the chat application which used VADER in real time. To test the performance of the Sentiment Analysis Module, chat messages that were generated while using the application were processed and

**VADER** - Valence Aware Dictionary and sEntiment Reasoner [9] was used to perform sentiment analysis in real time. It is a sentiment analysis tool which uses the lexicon and rule based approach which works on a sentence one word at a time. It maintains a lexicon of words each with their polarity scores where positive words have positive polarity scores and negative words have negative polarity scores. VADER is known to work very well when working with text sourced from social media but also works fairly good with text sourced from other domains as well. Let us see how VADER would process a sentence. It first checks for all the words in the sentence if they are present in its lexicon. For all the words that are present in the sentence as well its lexicon, it then retrieves the polarity score associated with those words from its lexicon. Now it generates a compound score using individual scores of all the words. The resulting score then acts as a indicator if the sentence is positive, negative, or neutral. This tool is also capable of handling negations for example, something like *not good* will result in a negative polarity score and *not bad* will result in a positive polarity score. Since VADER uses a lexicon to analyze the text, it might not be possible to update the lexicon every now and then. Due to this it does not work very well with the presence of slangs in texts. Since it works on a sentence word by word, it is also difficult to work with sarcasm and idioms which are tricky sentences. Also, its performance drops down if the context of the sentence under consideration is a complex one. The reason for choosing VADER as the Sentiment Analysis tool in this application is to be able to carry on the sentiment analysis process in real-time. Since the goal of this project is to perform Sentiment Analysis in real time as and when messages are received, we needed something that would process the texts in with almost no latency and overhead on the system. VADER [9] is quick and effective and provided good results both in terms of performance and accuracy.
Training classifier
Evaluating NaiveBayesClassifier results...
Accuracy: 0.759
F-measure [neg]: 0.7673745173745172
F-measure [pos]: 0.7499999999999999
Precision [neg]: 0.741604776119403
Precision [pos]: 0.7790948275862069
Recall [neg]: 0.795
Recall [pos]: 0.723

Process finished with exit code 0

stored to be later used as testing data. In the next section, we shall walk through the results of evaluation and findings of this project.

IV. Results and Issues Faced

In this section we will walk through the issues that I faced while working on the projects and the solutions to them along with results of evaluation of the application.

A. Issues and Solutions

The first major issue that I faced at the start of the project was to find a dataset containing chat logs. It was very difficult to find a suitable dataset for the scope of this project. This is mainly due to the fact that chat logs are not usually made public as they contain personal information and making them public would violate the privacy of the concerned people. A workaround was to use the sentence_polarity [13] corpus available in the NLTK library as the dataset to train the model and the messages generated from using the application as the testing data for final evaluation of the application.

While working on developing the logic for network connectivity and the GUI, there was a problem situation. When listening on a port for incoming connections, the application would go in a loop not triggering the chat window to open. After making changes in the code, the chat window would open but then the application would not be able to listen on any port for incoming connections. The solution I found to this problem was to separate and put the code in two different functions and use the concept of Multi-threading to trigger both the functions.

B. Results

The end product application developed was able to perform Real-Time Sentiment Analysis on the chat messages to determine the sentiments they contained which again reflected the mood of the message sender. The application was able to function correctly as per initial functionality requirements. To evaluate the performance of the Sentiment Analysis module, Naive Bayes Classifier model was used which was trained on the sentence_polarity corpus sourced from NLTK library and tested with the dataset which was obtained by compiling the messages generated while using the chat application. The overall accuracy that the application exhibited was decent a 76%. Looking at the results it can be seen that the application was able to equally classify sentences as positive and negative without being biased. This could be because the training dataset was not specific to the chatlogs domain and was a generic one. If it was possible to source a dataset containing actual chatlogs the results would have been different.

V. Future Work and Conclusion

In its current form, it is a prototype of a chat application having Real-Time Sentiment Analysis capabilities. A possible extension to this project can be developing it into a full-fledged chat application which will be able to connect users all over the globe with advanced networking capabilities. As of now, this application can only run on a local network which restricts the both the client and server to be on the same network. Most importantly it will be very beneficial to have a feature which stores all the conversations between two users along with the chat ratings which can later be used to perform time series analysis. Such kind of analysis can then be useful in evaluating and determining the chat rating trends between those users. This can again be extended to work in evaluating the success of customer support service where it can be used for lead identification and generation using chat archives.

Finally, to conclude, a prototype chat application which demonstrates how real time sentiment analysis can be used to detect user’s mood by analyzing the chat messages has been developed. Again, this was just an attempt to detect user’s mood using text which is not 100% effective and is only accurate up to a certain extent. Using NLTK libraries, performing Sentiment Analysis in real time is not time-intensive. By leveraging library support and inbuilt functions it is possible to determine polarity of simple sentences, but it is difficult when dealing with sentences having complex structure. This project was accomplished by using VADER [9] which is a lexicon based Sentiment Analysis tool along with Naive Bayes classifier. By using more advanced approaches for Sentiment Analysis that consider the context of the sentences as well, performance of the application can be further improved.

Acknowledgment

I would like to thank Dr. Carol Romanowski for giving me the opportunity work with her for this project and for all her timely and valuable feedbacks on my progress which kept me motivated throughout while working on the project. Also, I would like to thank my colloquium instructor Dr. Reynold Bailey for all the lectures from which I have benefited greatly and his feedback on every milestone presentation.

References


