CHAT REVIEWS

Cooray B.S.U.M

(IT14032066)

Degree of Bachelor of Science

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

October 2017

CHAT REVIEWS

Project Title: CHAT REVIEWS

Project ID: 17-066

Supervisor: Dr. Dharshana kasthurirathna

Dissertation submitted in partial fulfillment of the requirements for the degree of Science

Department of Information Technology

Sri Lanka Institute of Information Technology

October 2017

DECLARATION

I declare that this is my own work and this dissertation1 does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

…… …../…../…….

Cooray B.S.U.M

IT14032066

The above candidate has carried out research for the B.Sc. Special (Hons) degree in IT Dissertation under my supervision.

…… …../…../…….

Dr. Dharshana kasthurirathna

(Signature of the Supervisor)

ABSTRACT

The use of Internet chat applications has benefited many different segments of society. It also creates opportunities for criminal enterprise, terrorism, and espionage. We present a study of a real-world application of chat analysis which will analyze chat messages in four ways such as topic detection, Emotion Extraction, Evaluate healthy and Personal information sharing analysis. Also analyzing chat traffic has important applications for both the military and the civilian world. Here on this document, it compares the results of an unsupervised learning approach with those of a supervised classification approach with regards to chat review application. The paper also discusses some of the specific challenges presented by this chat review application.

Unsupervised learning techniques such as clustering are very popular for analyzing text for topic identification as well as emotion extraction. These techniques have several attractive features, the most significant being that they do not require labeled training examples. This however is also a disadvantage under some circumstances. Therefor meantime we do this research we will discover more and more technologies required for analyzing chat massages based on four different categories such as topic detection, Emotion Extraction, evaluate healthy and Personal information sharing analysis.

With use of this chat analysis application user will be able identify the chatting partner in analytical way. And system will keep an analytical review for each chat session user interacted. Also system will be capable of showing its analytical data in a user friendly manner (in a graphical way).

ACKNOWLEDGEMENT

The work described in this research paper was carried out as our 4th year research project for the subject Comprehensive Design Analysis Project. The completed final project is the result of combining all the hard work of the group members and the encouragement, support and guidance given by many others. Therefore, it is researchers' duty to express their gratitude to all who gave them the support to complete this major task.

The researchers are deeply indebted to supervisor Dr. Darshana kasthurirathna and Lecturers of Sri Lanka Institute of Information Technology whose suggestions, constant encouragement and support in the development of this research, particularly for the many stimulating and instructive discussions. We are also extremely grateful to Mr. Jayantha Amararachchi, Senior Lecturer/ Head-SLIIT Centre for Research who gave and confirmed the permission to carry out this research and for all the encouragement and guidance given.

The researchers also wish to thank all colleagues and friends for all their help, support, interest and valuable advices. Finally, the researchers would like to thank all others whose names are not listed particularly but have given their support in many ways and encouraged researchers to make this a success.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Definition
CMC	Computer-mediated Communication
NLP	Natural Language Processing
ML	Machine Learning
ASR	Automatic Speech Recognition
SNS	Social Networking Services
ASU	Automatic Speech Understanding
IM	Instant Messaging

1 INTRODUCTION

1.1 Background Context

Chat is an increasingly important form of CMC (Computer-mediated communication). It is employed by many sectors of society to improve communication, create value, and commit crimes.

We explored chat monitoring system first and how it is used to monitor chat messages. Then, we explored technologies related Chat monitoring systems and some of them are Natural Language Processing (NLP), Machine Learning (ML), followed by its applicability to chat. And text mining and NLP are commonly used together for different purposes, and one of most common applications is social media monitoring [3], where an analysis is performed on a pool of user-generated content to understand mood, emotions and awareness related to a topic [4].

Nowadays most of chat monitoring system use Machine learning algorithms for text classification. One of the main ML problems is text classification, which is used, for example, to detect spam, define the topic of a news article, or choose the correct mining of a multi-valued word. The Statsbot [5] team has already written how to train your own model for detecting spam emails, spam messages, and spam user comments [6]. And it's impossible to define the best text classifier. In fields such as computer vision, there's a strong consensus about a general way of designing models − deep networks with lots of residual connections. Unlike that, text classification is still far from convergence on some narrow area.

1.2 Research Gap

When comparing Chat Reviews web application with existing applications our one based on mainly analyzing trustworthiness of the chatting partner.

So when we compare existing chat monitoring applications with our one it only allows user for simple chat analysis features. So in that case other than the basic functions like keyword based search, topic identification emotion extraction, we gave a new functions (new message analytical areas) such as Detect Personal Information and Evaluate Healthy. This is not available on current any of the chat monitoring application.

Chat Monitoring Application	Keyword based searching	Topic Identification	Emotion Extraction	Message Encryption	Detect Personal Information	Evaluate Healthy
Intelligent Diagnosis System		**Yes**	**-**	**-**	**-**	
Honey Chatting	**-**	**-**	**-**	**Yes**	**-**	
GroupWize	**-**	**Yes**	**Yes**	**-**	**-**	
Chat Reviews	**-**	**Yes**	**Yes**	**-**	**Yes**	**Yes**

Table 1- Comparison of Exiting Application with the proposed system

Above diagram shows a brief comparison of the existing applications with the proposed system

1.3 Research problem

Nowadays we meet strangers all the time in our day to day life. So we attempt to have partnership with them without any hesitation at all. Also some time they are more and more smart than we think, then it's very difficult to identify the characteristics by only looking at their messages. Nowadays it is highly required to have intelligent way to detect those frauds (may be what that message really mean). We found that the solution is to use chat monitoring application for online chatting. Then we can review our messages in analytical way or it will lead us to think about our chatting partner in analytical way.

The research problem to be addressed by this research was identified as reviewing chat session in analytical way. A separate background analysis was carried on the usage of chat monitoring system and the possibility of developing such kinds of real world system.

Before developing the Chat Reviews Application based on machine learning concept, the project team went through a large amount of research papers to identify the main problems that need to be addressed when implementing the system. With the researches already done we got to know the existing technologies as well as upcoming technologies and algorithms and how to develop this system by modules.

It became clear that a unique system could be implemented by machine learning algorithms. And it was necessary to search for the most suitable machine learning technologies for implementing our system.

There were different problems that we considered during this research project. Some of them are;

- What are best machine learning algorithms for text classification and text mining?
- How to optimize existing machine learning algorithms?
- How to process large data set (Vocabulary)?
- How to collect training data set for each modules in the system?
- What is best programming language to implement machine learning algorithms?
- How to give real time out put to user?
- How to represent statistics generated by algorithms?
- How to keep the users interest to use our application?

This system tries to address the above problems with regard to user satisfaction and improved service and leave our new chat monitoring system in a user interactive manner.

1.4 Research objectives

- Introduce machine learning algorithms for each modules in the system.
- To review chat session in analytical way.
- To analyze characteristics (trustworthiness) of the chatting partner.
- Give user a graphical review of statistics

2 METHODOLOGY

2.1 Methodology

Several machine learning methodologies were used in the developing process by the development team. In my research module, Machine learning algorithms have been used. At the beginning of the development of the system, a deep study about machine learning concept was carried out and different articles were followed based on those. Weaknesses in the existing systems and new requirements were identified after communicating with different people with different age levels. Nowadays at least one social media application is being used by people belong to each and every age group. So the best way to do a study was to get some ideas and feedbacks from those users. At present social media applications are being used for different purposes. They are being used to interact with friends, for fun and to spend time. Therefor different users are having different perspectives. After the study, security of the system and speed of the algorithms was identified as the most important part missing in the existing applications.

After all the studies that had been carried out, I came up with a new area for developing this topic detection module. Developing an algorithm for topic detection module was was considered as one major segment of this module. The reason why this is important is it is more attracting feature of this chat monitoring system and it will be more useful for analytical purpose in the end. As I mentioned above, if user wants to review a particular chat session by topic, user can view statistics in graphical way.

Technologies and application platforms that we were going to use were decided at the important initial stages. Technical difficulties of the development methodologies that we were going to use and the efficiency of the algorithms we were going to implement were examined after doing a deep literary survey and surfing through internet. In the circumstances like ours, doing discussions and surveys is considered as the best or the only way to solve such technical and practical issues.

This research project was titled as Reviewing Chat session (Chat Reviews) based on different chat reviews module such as Topic detection, Emotion Extraction, Detect personal information and Evaluate healthy. The system must be able to extract the given message and analyze it in real-time (maximum of 10 seconds) also it will update the existing statistics immediately. And the graphical review will be provided as user friendly as possible.

Chat Review application is based on 4 main parts.

1. Topic Detection
2. Emotion Extraction
3. Detect Personal Information
4. Evaluate Healthy

After going through above modules final outcome will be shown to the user in the way of analyzing trustworthiness of particular chat session.

2.1.1 Algorithm for Emotion Extraction

Emotion Extraction module detects the emotion associated with messages. There are defined emotion classes along with the training data associated with each emotion class. The features are extracted from the ISEAR dataset. The proposed system select features and perform classification using Multinomial Naïve Bayes classifier.

The ISEAR dataset is in the form of sentences which are tagged with the emotion experienced by the user, who are writing the sentence. There are seven emotions in the dataset: anger, disgust, fear, guilt, joy, sadness, and shame. The sentences in the dataset need to be pre-processed before performing any type of operations in it.

Chi-square test features are found to be useful for feature extraction tasks. Generate a Feature Stats Object with metrics about the occurrences of the keywords in categories, the number of category counts and the total number of observations. These stats are used by the feature selection algorithm.

A chi-square test, also written as $\mathbf{X^2}$ test, is any statistical hypothesis test where the sampling distribution of the test statistic is chi-squared distribution when the null hypothesis is true. Without other qualification, 'chi-squared test' is often used as short for Pearson's chi-squared test. The chi-squared test is used to determine whether there is a significant difference between the expected frequencies and the observed frequencies in one or more categories.

In the standard applications of the test, the observations are classified into mutually exclusive classes, and there is some theory, or say null hypothesis, which gives the probability that any observation falls into the corresponding class. The purpose of the test is to evaluate how likely it is between observations and null hypothesis.

Chi-squared tests are often constructed from a sum of squared errors, or through a sample variance. Test statistics that follow a chi-squared distribution arise from the assumption of independently distributed data, which is valid in many cases due to the central limit theorem. A chi-squared test can be used to try to reject the null hypothesis that the data are independent.

```
N1dot = 0;
for(Integer count : categoryList.values()) {
    N1dot+=count;
}

//also the N0. (number of documents that DONT have the feature)
N0dot = stats.n - N1dot;

for(Map.Entry<String, Integer> entry2 : categoryList.entrySet()) {
    category = entry2.getKey();
    N11 = entry2.getValue(); //N11 is the number of documents that have the feature and belong on the
    N01 = stats.categoryCounts.get(category)-N11; //N01 is the total number of documents that do not

    N00 = N0dot - N01; //N00 counts the number of documents that don't have the feature and don't bel
    N10 = N1dot - N11; //N10 counts the number of documents that have the feature and don't belong to

    //calculate the chisquare score based on the above statistics
    chisquareScore = stats.n*Math.pow(N11*N00-N10*N01, 2)/((N11+N01)*(N11+N10)*(N10+N00)*(N01+N00));
```

N1dot = number of documents that have the feature

NOdot = number of documents that don't have the feature

N11 = number of documents that have the feature and belong on the specific category

NO1 = total number of documents that do not have the particular feature but they belong to the specific category

NO0 = number of documents that don't have the feature and don't belong to the specific category

N10 = number of documents that have the feature and don't belong to the specific category

For emotion identification from text, the multinomial implementation of Naïve Bayes classifier is used here. A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes theorem with strong (naive) independence assumptions. NB classifiers are used because it is fast, easy to implement and relatively effective . Multinomial Naive Bayes (MNB) classifier is a specific instance of a Naive Bayes classifier which uses a multinomial distribution for each of the features. Feature vector tables are constructed with the selected

features for the seven emotion datasets: anger, disgust, fear, guilt, joy, sadness, and shame. These feature vector tables are used to build the classification model.

With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial **$(p_1, \ldots, p_n)$** p_i is the probability that event i occurs (or K such multinomials in the multiclass case). A feature **vector X = $(x_1, \ldots, x_n)$** is then a histogram, with X_i counting the number of times event i was observed in a particular instance. This is the event model typically used for document classification, with events representing the occurrence of a word in a single document (see bag of words assumption). The likelihood of observing a histogram x is given by

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_{ki}^{x_i}$$

The multinomial naive Bayes classifier becomes a linear classifier when expressed in log-space.

$$\begin{aligned} \log p(C_k \mid \mathbf{x}) &\propto \log \left(p(C_k) \prod_{i=1}^{n} p_{ki}^{x_i} \right) \\ &= \log p(C_k) + \sum_{i=1}^{n} x_i \cdot \log p_{ki} \\ &= b + \mathbf{w}_k^\top \mathbf{x} \end{aligned}$$

where $b = \log p(C_k)$ and $w_{ki} = \log p_{ki}$.

If a given class and feature value never occur together in the training data, then the frequency-based probability estimate will be zero. This is problematic because it will wipe out all information in the other probabilities when they are multiplied. Therefore, it is often desirable to incorporate a small-sample correction, called pseudocount, in all probability estimates such that no probability is ever set to be exactly zero. This way of regularizing naive Bayes is called Laplace smoothing when the pseudocount is one, and Lidstone smoothing in the general case.

2.1.2 System Design

The design part is done by dividing it into several parts.

- User interface designing: This will be the first part in designing because the user is interacting with the interface which is provided. So the interface must be user friendly and must fulfill all the needs and requirements of the user. And must be attractive to user's eye.
- Database design: For the database management we decide to use MySQL [12] as it is more recommend for backend operation and Oracle Database 11g Release for topic detection module as it is supported for Data mining and Data analytical purposes. So System will consist with two databases.
- Algorithm design: We have used **Bernoulli document model** for classifying messages into topic.

2.1.3 Implementation

According to the proposed system user have the facility to access our system with any device that has internet facility. And System can receive messages from Facebook pages connected through messenger bot. Messages are going through all four modules integrated in the system within few seconds.

Topic detection module is developed using Java and Oracle Database 11g Release as database. It is integrated using PHP-Java integration methodologies with other three modules in the system.

2.1.4 Emotion Detection Overview

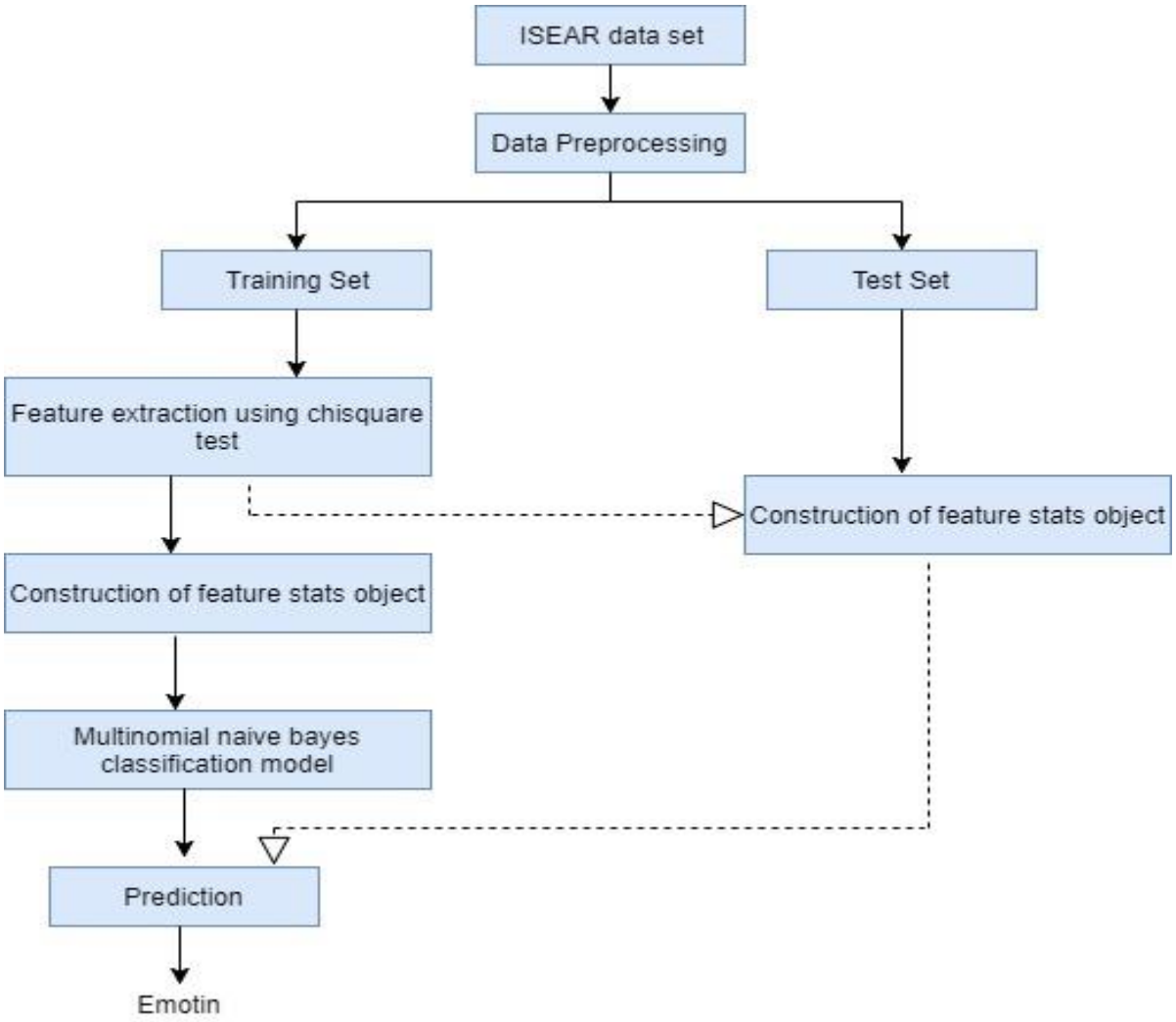

Figure 1 – Emotion Detection Overview

2.2 Testing and Implementation

2.2.1 Implementation Techniques

- HTTPS web host
- Servers required
    - Web Server - Appache Server

- Messenger bot in messenger platform
- To implement web application

- JavaScript
- PHP
- HTML
- CSS

- Microsoft Office package for Documentation
- Hardware Requirements
  - Internet Connection
  - And device that has the facility to access internet
- Software Requirements
  - Web browser enables with internet access

2.2.2 Testing Techniques

➢ **Testing Methodology**

Any software product should be sent through a testing process in order to identify the weaknesses or faults of the system. This section provides an overview of how the system behaves in the phases recognized in the system. This is accomplished through the testing process where each and every module or phase of the system is tested.

- **Unit Testing**

Unit testing will be carried out during the implementation process. Each unit will be tested by the developer himself.

- **Integration Testing**

Once more than one module is completed, integration testing will begin. Two modules will be integrated and tested. As a third module is joined, all three modules will be tested again.

- **System Testing**

Once all the modules have been successfully integrated, system testing will begin. Three kinds of System tests will be carried out. And for further testing application was given to some of the users and followed them.

2.2.3 User Interfaces

Following is the user interfaces for the graphical review of emotion extraction of particular chat session.

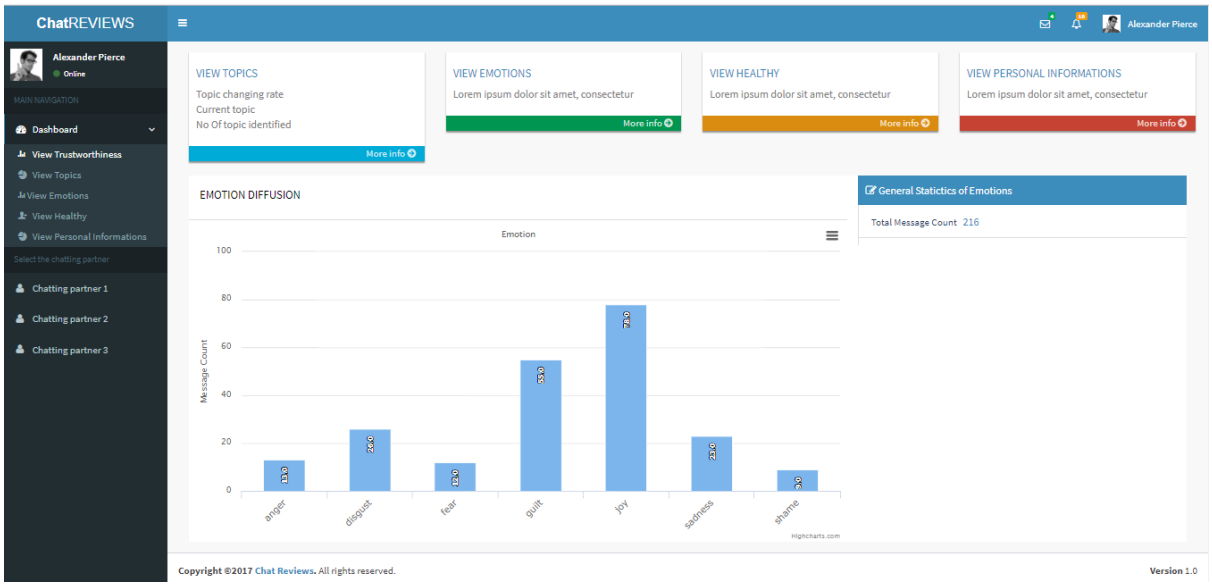

Figure 2 – UI of Emotion Extraction module

· User can find emotion diffusion in the particular chat session in the aim of analyzing trustworthiness of its partner.

2.3 Research Findings

In this research study we came across many new findings. One of them is implementing **Bernoulli document model** (One major document model used for document classification in the world most sophisticated web applications like Gmail, Yahoo for their email classifications) for Instant messages analysis in the aim of detecting topic. And Developing the **Bernoulli document model** with 466,544 number of English Vocabulary was huge challenge since the length of vocabulary affects to execution time of the algorithm.

Also for the Topic detection module we had to find training data (Instant Messages) for each Topic Class separately. Then Likelihoods were generated from collected training data and stored in database (Oracle Database 11g) for purpose of reading it by algorithm developed using Java and reduced the execution time of the whole module greatly.

Furthermore we can find out messages not related to any of the topic in the system from the algorithm improved with confidence is also a new improvement of the **Bernoulli document model.**

3 RESULT AND DISCUSSION

3.1 Result of Emotion Extraction Module

The following subsection provides evidence to the implementation results and the solutions provided for the identified research problems. The main user interface of the "Emotion Extraction" is shown with respective results. So that all the interfaces were designed with a simple, attractive manner to keep the continuous user interaction.

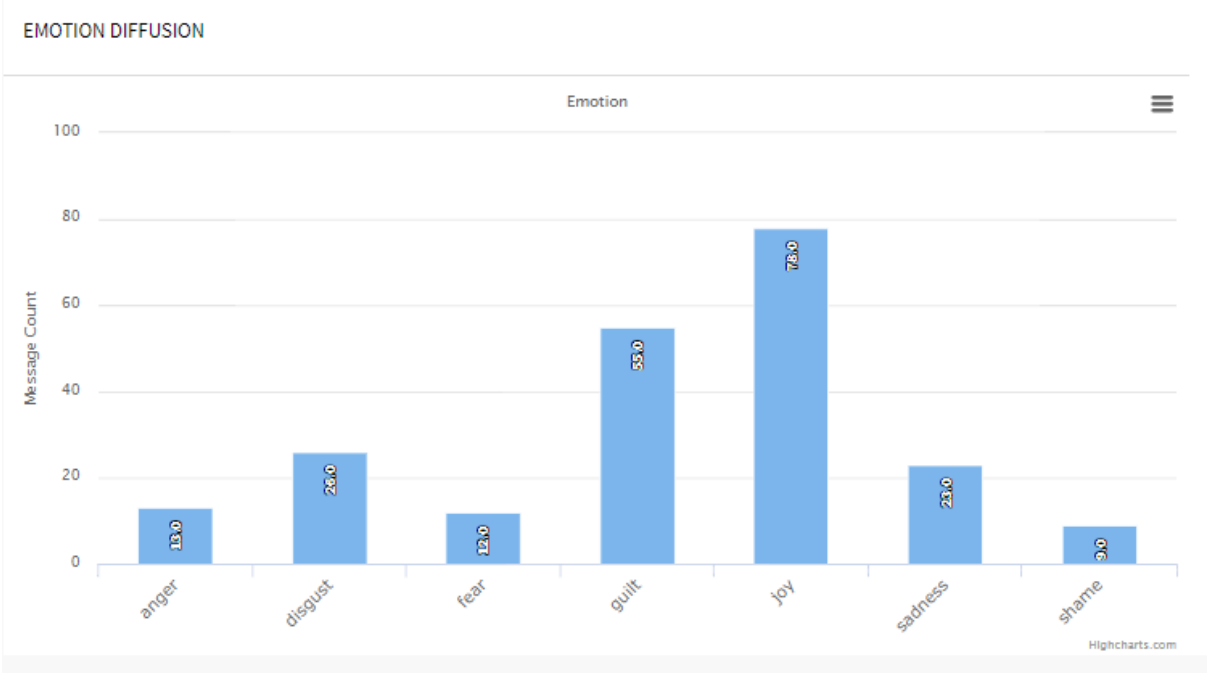

Figure 3 – Review of result of Emotion Extraction module

Results computed by Algorithm:

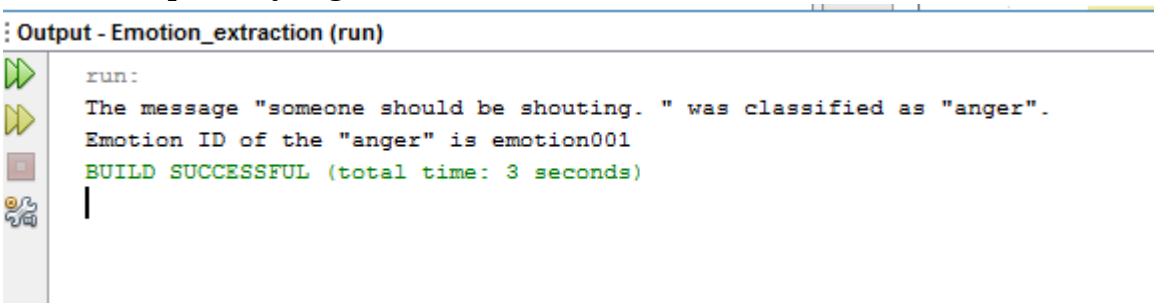

Figure 5– Results computed by algorithm

Above output shows the heights statistic scored emotion. For that computed the statistics for all emotions and then top one will be selected as the emotion which is having the highest number as shown above. Selected emotion is displayed in bottom line as shown above it is "emotion001".it is anger.

For an example;

Message 1: Some noisy guests arrived at the hotel.

Emotion: anger (emotion001)

Message 2: I saw someone that I thought I knew repeatedly drunk.

Emotion: disgust (emotion002)

Message 3: The first day that I was close to a dead body

Emotion: fear (emotion003)

3.1.1 Test Cases

It shows the test cases for emotion extraction module.

Test case #	Description	Pre-conditions	Test Steps	Input	Expected output
01	Analyze emotion diffusion	● Load the Chat-Reviews application	1. Go to Home Page 2. Select View Emotions	No inputs	● Graphical review of emotion diffusion for particular chat.
Test case #	Description	Pre-conditions	Test Steps	Input	Expected output
02	Messages matching with given emotions in the emotion extraction module	● Messages should be received from Facebook pages	1. Enter Message	Message: at the airport and my plane was just about so leave.	● Message should be classified as “sadness”, will increment the current value of bar chart.

Table 2 - Test cases

3.2 Discussion of the System

As discussed earlier the main target of this application to review a particular chat session and analyze trustworthiness of chatting partner from statistics computed by modules implemented in the system. So by using our system we would provide quality and interesting features for users.

The main system attributes is described as follows,

- Reliability

Reliability is the probability that an application will accurately perform its specified task under stated environment conditions. Simply, that is how much a user can depend on the system. The Purpose of this system is to provide a reliable and efficient service to any person around anywhere who wants to analyze his chatting partner. Other than these, application can be used as a communication media with other users.

- Availability

The specific system outage is not required for the application and system availability is totally dependent on the reliability of the system tools and interface devices. Chat Reviews system will be available at any time with internet connection. If the connection is lost, Application won't be available. User can be access through any device with internet access.

- Security

User will be able to login to the system with username and password which is given at the registration. Anyone who is having login details can access the system.

- Maintainability

Maintainability is designed as the probability of performing a successful repair action within a given time. In other words, maintainability measures the ease and Speed with which a system can be restored to operational status after a failure occurs. Also the basic thing is in that application falls identification part. Here in every case databases should be maintained properly.

- Privacy

The application can be used by several users after login with username and password. Therefore each user's performance and progress can only be viewed after login to the application using correct username and password.

- Modifiability

The system is able to add new modules with new features. Therefore whole application is built with object oriented and module concepts.

3.2.1 Flow of the Project

- From a research, our team identified problems related to social networks and that's all because of online fake users.

- The project team realized that there is a requirement of a chat monitoring application nowadays that can help people to understand their chatting partner in analytical way.

- When a user deal with the existing systems they got lot of problems. Finally all decided to build a chat monitoring application.

- After the initial discussion we were able to identify the basic problems in current systems and we got a clear idea about some features that were not covered even in places where they were using existing systems.

- All of our members identified the target of the system. We gave much priority to develop a useful and effective chat monitoring system which is better than existing systems.

4 CONCLUSION

And when considering chat monitoring system nowadays most of them just gives basic functionalities. But the proposed system has separate modules called **Topic Detection**, **Emotion extractions**, **Detect Personal Information** and **Evaluate Healthy**. And they provide strong analytical statistics which is used to analyze chat session in the aim of analyzing trustworthiness of chatting partner. Also statistics computed by modules will be displayed in a graphical way for the user in a user friendly manner. Then the user will be able to review result generated by each modules separately easily.

So finally the outcome of our proposed system carries web application with Socializing concepts to analyze our chatting partner. And based on analytical review user will be able to review his chatting partner's characteristics (Trustworthiness). System also support to review multiple chat session by one user then user also be able to compare his chatting partners in analytical way. Analyzing multiple chatting partners is an outstanding feature given by Chat Reviews.

Finally by using this chat monitoring system user can be aware his chatting partner and can be safe from unnecessary things like "giving money for a person never met or seen before" upon a request made by chatting partner. For the safe chatting in social networks or any other chatting application we recommend to use **Chat Reviews**.

REFERENCES

[1] Messenger.msn.com, 'MSN Messenger', 2014. [Online]. Available: http://messenger.msn.com/. [Accessed 13- July- 2017].

[2] Messenger.yahoo.com, 'Yahoo Messenger. Available', 2014. [Online]. Available: http://messenger.yahoo.com/. [Accessed: 13- July- 2017].

[3] Expertsystem.com, 'social-media-monitoring', 2016. [Online]. Available: http://www.expertsystem.com/solutions/corporate-intelligence/social-media-monitoring/. [Accessed 13- July- 2017].

[4] Expertsystem.com, 'Natural Language Processing And Text mining', 2016. [Online]. Available: http://www.expertsystem.com/natural-language-processing-and-text-mining/. [Accessed 13- July- 2017].

[5] Statsbot.co, ‘Where analitics happens’, 2017. [Online]. Available: https://statsbot.co/?utm_source=blog&utm_medium=post&utm_campaign=text_classifier. [Accessed 13- July- 2017].

[6] Statsbot.co, ‘Data Scientist Resume Projects’, 2017. [Online]. Available: https://blog.statsbot.co/data-scientist-resume-projects-806a74388ae6. [Accessed 13- July- 2017].

[7] K. Tzeras and S. Hartman, “Automatic indexing based on Bayesian inference networks”. In: 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93), pp. 22-34, 1993.

[8] Mattshomepage.com, ‘Bernoulli Naive Bayes Classifier’, 2014. [Online]. Available: http://mattshomepage.com/articles/2016/Jun/07/bernoulli_nb/. [Accessed: 20- July- 2017].

[9] Wikipedia.org, ‘Supervied learning overview’, 2014. [Online]. Available: https://en.wikipedia.org/wiki/Supervised_learning. [Accessed: 19- July- 2017].

[10] Wikipedia.org, ‘Bag-of-words model’, 2014. [Online]. Available: https://en.wikipedia.org/wiki/Bag-of-words_model. [Accessed: 19- July- 2017].

[11] Wikipedia.org, ‘Feature Vector for pattern recognition’, 2014. [Online]. Available : https://en.wikipedia.org/wiki/Feature_vector. [Accessed: 14-April- 2017].

[12] Mysql.com, ‘Mysql server’, 2017. [Online]. Available : https://www.mysql.com/. [Accessed: 14-April- 2017].

APPENDICES

Appendix A: Source Code

```
package classifier;


import dataObjects.Document;
import dataObjects.FeatureStats;
import dataObjects.NaiveBayesKnowledgeBase;
import features.FeatureExtraction;
import features.TextTokenizer;
```

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;

/**
 * Implements Multinomial Naive Bayes Text Classifier
 */
public class NaiveBayes {
   private double chisquareCriticalValue = 10.83; //equivalent to pvalue 0.001. It is used by feature selection algorithm

   private NaiveBayesKnowledgeBase knowledgeBase;

   /**
    * This constructor is used when we load an already train classifier
    *
    * @param knowledgeBase
    */
   public NaiveBayes(NaiveBayesKnowledgeBase knowledgeBase) {
      this.knowledgeBase = knowledgeBase;
   }

   /**
    * This constructor is used when we plan to train a new classifier.
    */
   public NaiveBayes() {
      this(null);
   }
```

```
/**
 * Gets the knowledgebase parameter
 *
 * @return
 */
public NaiveBayesKnowledgeBase getKnowledgeBase() {
   return knowledgeBase;
}

/**
 * Gets the chisquareCriticalValue paramter.
 *
 * @return
 */
public double getChisquareCriticalValue() {
   return chisquareCriticalValue;
}

/**
 * Sets the chisquareCriticalValue parameter.
 *
 * @param chisquareCriticalValue
 */
public void setChisquareCriticalValue(double chisquareCriticalValue) {
   this.chisquareCriticalValue = chisquareCriticalValue;
}

/**
 * Preprocesses the original dataset and converts it to a List of Documents.
 *
```

```
     * @param trainingDataset
     * @return
     */
    private List<Document> preprocessDataset(Map<String, String[]> trainingDataset) {
        List<Document> dataset = new ArrayList<>();

        String category;
        String[] examples;

        Document doc;

        Iterator<Map.Entry<String, String[]>> it = trainingDataset.entrySet().iterator();

        //loop through all the categories and training examples
        while(it.hasNext()) {
            Map.Entry<String, String[]> entry = it.next();
            category = entry.getKey();
            examples = entry.getValue();

            for(int i=0;i<examples.length;++i) {
                //for each example in the category tokenize its text and convert it into a Document object.
                doc = TextTokenizer.tokenize(examples[i]);
                doc.category = category;
                dataset.add(doc);


            }


        }
```

```
        return dataset;
    }

    /**
     * Gathers the required counts for the features and performs feature selection
     * on the above counts. It returns a FeatureStats object that is later used
     * for calculating the probabilities of the model.
     *
     * @param dataset
     * @return
     */
    private FeatureStats selectFeatures(List<Document> dataset) {
        FeatureExtraction featureExtractor = new FeatureExtraction();

        //the FeatureStats object contains statistics about all the features found in the documents
        FeatureStats stats = featureExtractor.extractFeatureStats(dataset); //extract the stats of the dataset

        //we pass this information to the feature selection algorithm and we get a list with the selected features
        Map<String, Double> selectedFeatures = featureExtractor.chisquare(stats, chisquareCriticalValue);

        //clip from the stats all the features that are not selected
        Iterator<Map.Entry<String, Map<String, Integer>>> it = stats.featureCategoryJointCount.entrySet().iterator();
        while(it.hasNext()) {
            String feature = it.next().getKey();

            if(selectedFeatures.containsKey(feature)==false) {
                //if the feature is not in the selectedFeatures list remove it
```

```
            it.remove();
        }
    }

    return stats;
}

/**
 * Trains a Naive Bayes classifier by using the Multinomial Model by passing
 * the trainingDataset and the prior probabilities.
 *
 * @param trainingDataset
 * @param categoryPriors
 * @throws IllegalArgumentException
 */
public void train(Map<String, String[]> trainingDataset, Map<String, Double>
categoryPriors) throws IllegalArgumentException {
    //preprocess the given dataset
    List<Document> dataset = preprocessDataset(trainingDataset);



    //produce the feature stats and select the best features
    FeatureStats featureStats =  selectFeatures(dataset);



    //intiliaze the knowledgeBase of the classifier
    knowledgeBase = new NaiveBayesKnowledgeBase();
    knowledgeBase.n = featureStats.n; //number of observations
    knowledgeBase.d = featureStats.featureCategoryJointCount.size(); //number of features
```

```
        //check is prior probabilities are given
        if(categoryPriors==null) {
            //if not estimate the priors from the sample
            knowledgeBase.c = featureStats.categoryCounts.size(); //number of cateogries
            knowledgeBase.logPriors = new HashMap<>();

            String category;
            int count;
            for(Map.Entry<String, Integer> entry : featureStats.categoryCounts.entrySet()) {
                category = entry.getKey();
                count = entry.getValue();

                knowledgeBase.logPriors.put(category,
Math.log((double)count/knowledgeBase.n));
            }
        }
        else {
            //if they are provided then use the given priors
            knowledgeBase.c = categoryPriors.size();

            //make sure that the given priors are valid
            if(knowledgeBase.c!=featureStats.categoryCounts.size()) {
                throw new IllegalArgumentException("Invalid priors Array: Make sure you pass a
prior probability for every supported category.");
            }

            String category;
            Double priorProbability;
            for(Map.Entry<String, Double> entry : categoryPriors.entrySet()) {
                category = entry.getKey();
                priorProbability = entry.getValue();
```

```
            if(priorProbability==null) {
                throw new IllegalArgumentException("Invalid priors Array: Make sure you pass a prior probability for every supported category.");
            }
            else if(priorProbability<0 || priorProbability>1) {
                throw new IllegalArgumentException("Invalid priors Array: Prior probabilities should be between 0 and 1.");
            }

            knowledgeBase.logPriors.put(category, Math.log(priorProbability));
        }
    }

    //We are performing laplace smoothing (also known as add-1). This requires to estimate the total feature occurrences in each category
    Map<String, Double> featureOccurrencesInCategory = new HashMap<>();

    Integer occurrences;
    Double featureOccSum;
    for(String category : knowledgeBase.logPriors.keySet()) {
        featureOccSum = 0.0;
        for(Map<String, Integer> categoryListOccurrences : featureStats.featureCategoryJointCount.values()) {
            occurrences=categoryListOccurrences.get(category);
            if(occurrences!=null) {
                featureOccSum+=occurrences;
            }
        }
        featureOccurrencesInCategory.put(category, featureOccSum);
    }

  //  estimate log likelihoods
```

```
        String feature;
        Integer count;
        Map<String, Integer> featureCategoryCounts;
        double logLikelihood;
        for(String category : knowledgeBase.logPriors.keySet()) {
           for(Map.Entry<String, Map<String, Integer>> entry :
featureStats.featureCategoryJointCount.entrySet()) {
              feature = entry.getKey();
              featureCategoryCounts = entry.getValue();

              count = featureCategoryCounts.get(category);
              if(count==null) {
                 count = 0;
              }

              logLikelihood =
Math.log((count+1.0)/(featureOccurrencesInCategory.get(category)+knowledgeBase.d));
              if(knowledgeBase.logLikelihoods.containsKey(feature)==false) {
                 knowledgeBase.logLikelihoods.put(feature, new HashMap<String, Double>());
              }
              knowledgeBase.logLikelihoods.get(feature).put(category, logLikelihood);
           }
        }
        featureOccurrencesInCategory=null;
     }

     /**
      * Wrapper method of train() which enables the estimation of the prior
      * probabilities based on the sample.
      *
      * @param trainingDataset
```

```
     */
    public void train(Map<String, String[]> trainingDataset) {
        train(trainingDataset, null);
    }

    /**
     * Predicts the category of a text by using an already trained classifier
     * and returns its category.
     *
     * @param text
     * @return
     * @throws IllegalArgumentException
     */
    public String predict(String text) throws IllegalArgumentException {
        if(knowledgeBase == null) {
            throw new IllegalArgumentException("Knowledge Bases missing: Make sure you
train first a classifier before you use it.");
        }

        //Tokenizes the text and creates a new document
        Document doc = TextTokenizer.tokenize(text);



        String category;
        String feature;
        Integer occurrences;
        Double logprob;

        String maxScoreCategory = null;
        Double maxScore=Double.NEGATIVE_INFINITY;
```

```
    //Map<String, Double> predictionScores = new HashMap<>();
    for(Map.Entry<String, Double> entry1 : knowledgeBase.logPriors.entrySet()) {
       category = entry1.getKey();
       logprob = entry1.getValue(); //intialize the scores with the priors

       //foreach feature of the document
       for(Map.Entry<String, Integer> entry2 : doc.tokens.entrySet()) {
          feature = entry2.getKey();

          if(!knowledgeBase.logLikelihoods.containsKey(feature)) {
             continue; //if the feature does not exist in the knowledge base skip it
          }

          occurrences = entry2.getValue(); //get its occurrences in text

          logprob += occurrences*knowledgeBase.logLikelihoods.get(feature).get(category);
//multiply loglikelihood score with occurrences
       }


       if(logprob>maxScore) {
          maxScore=logprob;
          maxScoreCategory=category;
       }


    }

    return maxScoreCategory; //return the category with heighest score
  }
```

```
}
```