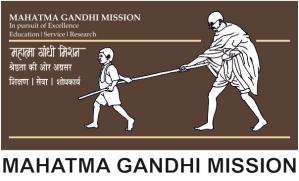
****

**MGM’S COLLEGE OF ENGINEERING & TECHNOLOGY  
 DEPARTMENT OF COMPUTER ENGINEERING**

**MINI PROJECT REPORT**

**ON**

**Automated Comment Filter**

**Submitted By:**

**Murtaza Khilawala(117CP3017A)**

**Gaurav Kamble(117CP3314A)**

**Sagar Kirtane(117CP3288A)**

**Under the Supervision of**

**Prof. Abhijit J Patil**

**For the Subject**

**NATURAL LANGUAGE PROCESSING**

**IN ACADEMIC YEAR 2019-20**

**MGM’S COLLEGE OF ENGINEERING & TECHNOLOGY  
 DEPARTMENT OF COMPUTER ENGINEERING**

**CERTIFICATE**

This is to certify that Mr. Murtaza Khilawala , Gaurav Kamble , Sagar Kirtane of B.E Class, 8th Sem, A Div bearing UICN No. 117CP3017A , 117CP3314A , 117CP3288A Roll No. 56 , 50 , 57 in the Department of Computer Engineering has completed the Mini Project work during the study of Natural Language Processing in Eight Semester and report embodies the result of work. It is therefore recommended and forwarded for submission.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Subject In-charge HOD

Date- \_\_\_\_\_\_\_\_\_\_\_

**Table of Content**

|  |  |  |
| --- | --- | --- |
| SR.NO. | TOPIC | PG. NO |
| 1. | Introduction | 6 |
| 2. | Procedure | 9 |
| 3. | Flow Chart | 10 |
| 4. | Source code | 11 |
| 5. | Output | 21 |
| 6. | Applications | 23 |
| 7 | Future Scope | 24 |
| 8. | References | 25 |

**Table of Figure**

|  |  |  |
| --- | --- | --- |
| SR. NO. | FIGURE NAME | PAGE NO |
| 1.1 | Filtering Mechanism and Black List | 8 |
| 3.1 | Flow chart | 10 |
| 5.1 | Validation Form | 21 |
| 5.2 | Comment Section | 21 |
| 5.3 | Comment Box | 22 |

**ABSTRACT**

*In this generation, using Online Social Network (OSN) is an unavoidable powerful weapon to exhibit peoples’ views and ideas. The users depending upon their interests can select the persons who must post/comment messages in their wall. The present excavation in OSN user wall is “No filtering of abusive messages”. That is the selected persons can post any sort of messages in their wall. So in this project, we programmed a filtered wall to permeate offensive messages using rule based and text classification techniques. We have evaluated the performance using lexical analysis and semantics analysis.*

**CHAPTER - 1**

**INTRODUCTION**

ocial networking is basically a social interaction which relies u

pon personal relationships. It is a remarkable

advancement on web which is highly needed for today’s expeditious evolution. Social networking websites for

instance Facebook, LinkedIn, and MySpace extends to us bountiful features. Let’s discuss them below:

x Flashing idea of sharing images, videos, and files.

x Posting messages on friend’s wall.

x Chat box that offers secrecy between two people.

x Conference chats with numerous people at a time.

x Following certain eminent persons or pages.

x Secure sharing of personal informat

ion with a definitive group of friends

ocial networking is basically a social interaction which relies u

pon personal relationships. It is a remarkable

advancement on web which is highly needed for today’s expeditious evolution. Social networking websites for

instance Facebook, LinkedIn, and MySpace extends to us bountiful features. Let’s discuss them below:

x Flashing idea of sharing images, videos, and files.

x Posting messages on friend’s wall.

x Chat box that offers secrecy between two people.

x Conference chats with numerous people at a time.

x Following certain eminent persons or pages.

x Secure sharing of personal informat

ion with a definitive group of friends

ocial networking is basically a social interaction which relies u

pon personal relationships. It is a remarkable

advancement on web which is highly needed for today’s expeditious evolution. Social networking websites for

instance Facebook, LinkedIn, and MySpace extends to us bountiful features. Let’s discuss them below:

x Flashing idea of sharing images, videos, and files.

x Posting messages on friend’s wall.

x Chat box that offers secrecy between two people.

x Conference chats with numerous people at a time.

x Following certain eminent persons or pages.

x Secure sharing of personal informat

ion with a definitive group of friends

Social networking is basically a social interaction which relies u

pon personal relationships. It is a remarkable

advancement on web which is highly needed for today’s expeditious evolution. Social networking websites for

instance Facebook, LinkedIn, and MySpace extends to us bountiful features. Let’s discuss them below:

x Flashing idea of sharing images, videos, and files.

x Posting messages on friend’s wall.

x Chat box that offers secrecy between two people.

x Conference chats with numerous people at a time.

x Following certain eminent persons or pages.

x Secure sharing of personal informat

ion with a definitive group of friends

Social networking is basically a social interaction which relies upon personal relationships. It is a remarkable advancement on web which is highly needed for today’s expeditious evolution. Social networking websites for instance Facebook, LinkedIn, and MySpace extends to us bountiful features. Let’s discuss them below:

* Flashing idea of sharing images, videos, and files.
* Posting messages on friend’s wall.
* Chat box that offers secrecy between two people.
* Conference chats with numerous people at a time.
* Following certain eminent persons or pages.
* Secure sharing of personal information with a definitive group of friends.
* Imparting our likes and dislikes.

Social network holds an architecture that must have a profile, friends, groups, Discussions, Widgets. However all these are optional. A worthy point to note is that Social networks may also have a few detriments. It becomes an open ground for hackers to commit faults and launch virus attacks. It may also result in scams and false use of data or information . Sharing of information may sometimes be vulgar and it is inevitable. In this paper, we propose an idea of “Filtered wall” on the basis of personalized filtering recommendation algorithm . This notion is fundamentally developed to preclude the appearance abusive words on the user wall . We adopt the concept of expert analysis wherein a third part is given the utmost importance. The third party serves as a critic and he checks with the available documents for the presence of those abusive words. If present, the word is blocked as per the third party’s decision.

MESSAGE FILTERING SYSTEM:

Information Filtering is one of the best solutions for the aforementioned issue. It can be implemented in OSN for a variety of reasons like in posting, commenting messages on walls. There exist numerous techniques to visualize information filtering for the removal of offensive/vulgar messages.

The main contribution of the proposed method is the design of the system which provides customizable contentbased message filtering for the Online Social Networks, based on the Machine Learning techniques. Our work is related to the policy-based personalization for the OSN.

Message filtering systems is usually designed to classify the data and dynamically generated information dispatched from the sender (i.e) information producer to the user and the received information should likely satisfy the requirements of him/her [Schapire and Singer 2000]. In the content-based filtering method, every user is assumed as independent. So, a content based filtering system selects the data based on the correlation between the user preferences and the content. This is completely opposite to the collaborative filtering system, which selects the items based on the correlation between the people with the similar user preferences.

For the most of earlier research, electronic mail has been preferred as the domain for the information filtering. Documents which were processed by the content-based filtering is mostly in the form of text . This helps content-based filtering to come closer to the text classification. The filtering process can be designed as splitting the incoming document or information into relevant and non relevant. The complex filtering systems categorize the messages automatically as thematic categories.

Content-based filtering is based on the Machine Learning techniques. By using ML technique, a classifier is automatically induced by learning from the pre defined examples. Recently, there is large number of related work has been available. These available methods usually differ in the feature extraction methods. Sometimes, it also differs in model learning and collection of samples too. Several experiments prove that Bag-of-Words approaches yield good performances. By considering the learning model, the major approaches are content-based filtering and text classification. These methods usually show similar advantages and disadvantages. Most of the text filtering method by the ML has been applied to the long-form text. The performance of the text classification methods depends on the nature of the text document. Applying the content-based filtering to the messages posted on the OSN user wall has challenges due to the short length of the messages. This short text classification has got high attention in the research community. Recent work has spread lights on robustness, misspellings, nonstandard terms and noise.

To overcome the disadvantage of Block List we provide “Black List” where users are blocked temporarily for few days. This feature gets invoked when the person continuously posts (more than 3 times) offensive messages on user’s wall. Person in Black List cannot post messages on user wall for those temporary days. After that he/she will be able to post on the wall. Fig.1.1 depicts these features. At any point of time, their relationship with user will not be disturbed. Before enclosing the person into Black List, profile checking will be done by taking into account their relationship strength (%), trust (%), kind of relationship (friend, family, etc) with the user.

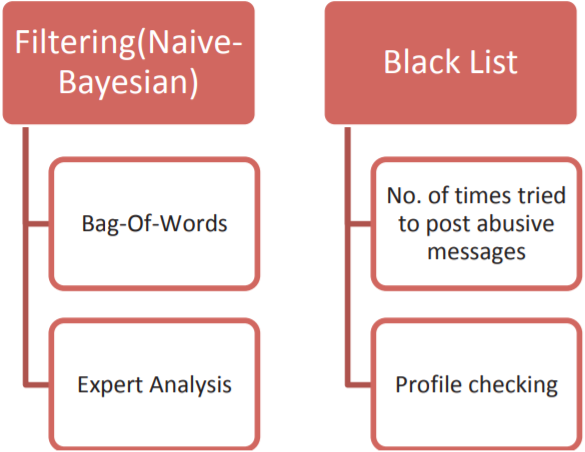


Fig.1.1: Filtering Mechanism and Black List

**CHAPTER -2**

**PROCEDURE**

1. Login using Validation form where we give the option of signing in or register using new username and password.
2. A comment section and comment box will be available for the client to add a comment and view all the comments on a post respectively.
3. After clicking the submit button the contents in the comment section will be given to the Moderator.java class and it will call various classes like CensorText.java class and ScamRemover.java class.
4. CensorText.java class will censor all the blacklisted word appearing in the comment text and replace it with asterik(\*).
5. ScamRemover.java class can remove all the scam website of a blacklisted domain name and give the comment text without the scam website.
6. After all the filtering of the comment area, the Moderator.java class will call Filer.java class file and post all the comments in a comment.txt file.
7. After posting the comment, The comment box is refreshed.

**CHAPTER - 3**

**FLOW CHART**

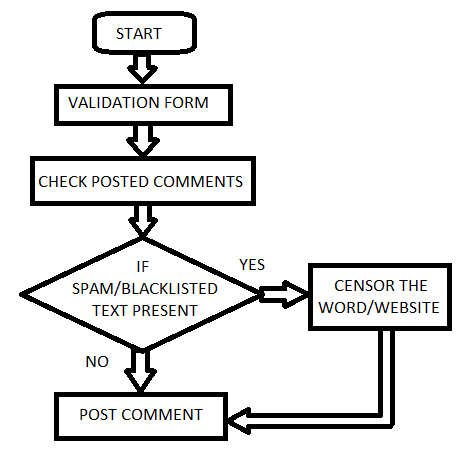
****

Fig.3.1: Flow chart

**CHAPTER - 4**

**SOURCE CODE**

**4.1: Moderator.java:**

package com.moderator.model;

import java.io.IOException;

public class Moderator {

private String username;

private String password;

private final String commentFile = "comment.txt";

public Moderator(String username, String password) {

this.username = username;

this.password = password;

}

public void publishComment(String comment) throws IOException {

if (comment != null || comment != "" || comment != "\\s") {

String text = getComment();

CensorText censor = new CensorText();

comment = censor.validate(comment);

ScamRemover scam = new ScamRemover();

comment = scam.remove(comment);

String newEntry = username + ":" + comment;

Filer filer = new Filer(commentFile);

filer.write(text + newEntry);

}

}

public String getComment() throws IOException {

Filer filer = new Filer(commentFile);

return filer.read();

}

}

**4.2: CensorText.java:**

package com.moderator.model;

import java.io.IOException;

public class CensorText {

String censorFile = "censoredwords.txt";

String validate(String text) throws IOException {

String[] textArray = text.split("\\s");

for (int i = 0; i < textArray.length; i++) {

if (isCensored(textArray[i])) {

textArray[i] = censored(textArray[i]);

}

}

text = "";

for (int i = 0; i < textArray.length; i++) {

text += textArray[i];

if (i != textArray.length - 1)

text += " ";

}

return text;

}

private String censored(String text) {

char[] vowel = { 'a', 'e', 'i', 'o', 'u' };

for (int i = 0; i < vowel.length; i++) {

text = text.replace(vowel[i], '\*');

}

return text;

}

private boolean isCensored(String text) throws IOException {

String[] words = getCensoredWords();

for (String word : words) {

if (text.contains(word))

return true;

}

return false;

}

private String[] getCensoredWords() throws IOException {

Filer filer = new Filer(censorFile);

String text = filer.read();

return text.split(",");

}

}

**4.3: ScamRemover.java:**

package com.moderator.model;

import java.io.IOException;

public class ScamRemover {

String scamFile = "scamwebsite.txt";

public String remove(String text) throws IOException {

String[] textArray = text.split("\\s");

for (int i = 0; i < textArray.length; i++) {

if (isScam(textArray[i])) {

textArray[i] = "";

}

}

text = "";

for (int i = 0; i < textArray.length; i++) {

text += textArray[i];

if (i != textArray.length - 1)

text += " ";

}

return text;

}

private boolean isScam(String text) throws IOException {

String[] words = getScammedWebsite();

for (String word : words) {

if (text.contains(word))

return true;

}

return false;

}

private String[] getScammedWebsite() throws IOException {

Filer filer = new Filer(scamFile);

String text = filer.read();

return text.split(",");

}

}

**4.4: Account.java:**

package com.moderator.model;

import java.io.IOException;

public class Account {

private String fileName = "account.txt";

Filer filer = new Filer(fileName);

public boolean isNameTaken(String name) throws IOException {

String accountInfo = filer.read();

String[] list = accountInfo.split("\n");

for (String line : list) {

String[] data = line.split(",");

if (data[0].equals(name)) {

return true;

}

}

return false;

}

public boolean isAccountAvailable(String name, String password) throws IOException {

String accountInfo = filer.read();

String[] list = accountInfo.split("\n");

for (String line : list) {

String[] data = line.split(",");

String name1 = data[0];

String password1 = data[1].replace("\n", "");

if (name1.equals(name)) {

if (password1.equalsIgnoreCase(password))

return true;

}

}

return false;

}

public void addData(String username, String password) throws IOException {

filer.write(filer.read() + username + "," + password + ",");

}

}

**4.5: Filer.java:**

package com.moderator.model;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class Filer {

private String fileName;

public Filer(String fileName) {

this.fileName=fileName;

}

public void write(String string) throws IOException {

try (FileWriter writer = new FileWriter(fileName)) {

writer.write(string+"\n");

writer.flush();

}

}

public String read() throws IOException {

String text = "";

int ascii;

FileReader reader = null;

try {

reader = new FileReader(fileName);

} catch (FileNotFoundException fe) {

System.out.println("File Not Found");

}

while ((ascii = reader.read()) != -1) {

text = text + ((char) ascii);

}

return text;

}

}

**4.6: ModeratorTest.java:**

package com.moderator.model.test;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.IOException;

import com.moderator.model.\*;

public class ModeratorTest {

static String comment;

static String name;

static String text;

static Moderator mod = null;

public static void main(String[] args) throws IOException {

Toolkit toolkit = Toolkit.getDefaultToolkit();

Dimension dimension = toolkit.getScreenSize();

Frame loginWindow = new Frame("Login");

loginWindow.setLayout(null);

Frame mainWindow = new Frame("Main");

mainWindow.setSize(dimension);

mainWindow.setLayout(null);

mainWindow.setVisible(false);

Label username = new Label("Name:", Label.CENTER);

Label password = new Label("Password:", Label.CENTER);

Label info=new Label("",Label.CENTER);

TextField name = new TextField(20);

TextField pass = new TextField(20);

pass.setEchoChar('\*');

Button login = new Button("submit");

Button register = new Button("Register");

Button close = new Button("cancel");

loginWindow.add(username);

loginWindow.add(name);

loginWindow.add(password);

loginWindow.add(info);

loginWindow.add(pass);

loginWindow.add(login);

loginWindow.add(close);

loginWindow.add(register);

username.setBounds(70, 90, 90, 20);

password.setBounds(70, 130, 90, 20);

info.setBounds(100, 310, 200, 20);

name.setBounds(170, 90, 110, 20);

pass.setBounds(170, 130, 110, 20);

login.setBounds(40, 260, 100, 40);

register.setBounds(150, 260, 100, 40);

close.setBounds(260, 260, 100, 40);

loginWindow.setVisible(true);

loginWindow.setSize(400, 400);

TextField commentArea = new TextField();

commentArea.setBounds(20, dimension.height-215, dimension.width / 2 - 10, 100);

List field = new List();

field.setBounds(dimension.width / 2 + 20, 50, dimension.width / 2 - 40, dimension.height - 100);

Button close1 = new Button("Close");

close1.setBounds(20, dimension.height-110, 230, 50);

Button commentPost = new Button("Comment");

commentPost.setBounds(230, dimension.height-110, 250, 50);

mainWindow.add(commentPost);

mainWindow.add(close1);

mainWindow.add(commentArea);

mainWindow.add(field);

close.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

login.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String uname = name.getText();

String upass = pass.getText();

if (uname != "" || upass != "") {

Account account = new Account();

try {

if (account.isAccountAvailable(uname, upass)) {

mod = new Moderator(uname, upass);

mainWindow.setVisible(true);

refreshComment(field);

} else {

info.setText("Invalid Credentials");

}

} catch (IOException e1) {

}

}

}

});

register.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String uname = name.getText();

String upass = pass.getText();

if (uname != "\\s" || upass != "\\s") {

Account account = new Account();

try {

if (account.isNameTaken(uname)) {

info.setText("Username already taken");

} else {

account.addData(uname, upass);

}

} catch (IOException e1) {

}

}

}

});

close1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

commentPost.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

comment = commentArea.getText();

mod.publishComment(comment);

commentArea.setText("");

field.removeAll();

refreshComment(field);

} catch (IOException e1) {

e1.printStackTrace();

}

}

});

}

private static void refreshComment(List field) throws IOException {

String text = mod.getComment();

String[] textArray = text.split("\n");

for (String text1 : textArray) {

field.add(text1);

}

}

}

**CHAPTER - 5**

**OUTPUT**

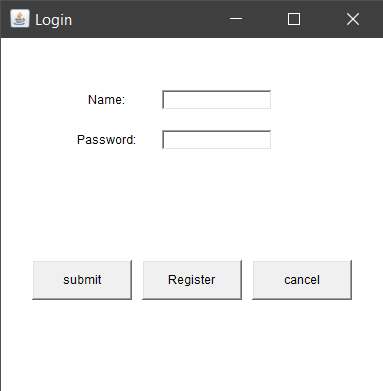


Fig.5.1: Validation Form



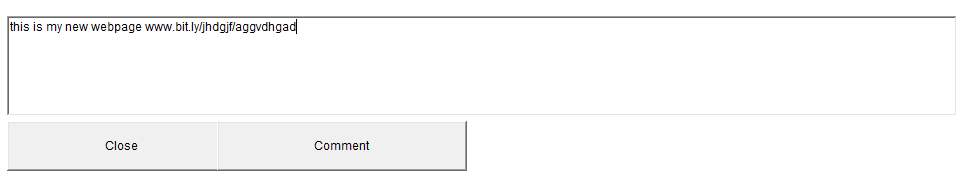
Fig.5.2: Comment Section



Fig5.3: Comment Box

**CHAPTER - 6**

**APPLICATION**

* **Dealing with hate and spam:**

Let’s rule out the obvious. Haters, trolls, and spammers are, like it or not, a part of the online community. But that does not necessarily mean you have to deal with each and every one of them individually. Say you have a social campaign going on. To ensure that no online negativity will influence your strategy, you can employ automation techniques. Once you identify hateful messages (e.g. by filtering out offensive words or phrases), you can address them with a thoughtful reply, hide, or delete them – the approach is up to you.

### Managing an e-commerce profile:

### If you are an online merchant, you probably have a lot of ads and campaigns going on. You showcase your amazing products and services on your Facebook account to build interest and generate revenue – plain and simple. So you naturally get a lot of questions on basic details – what’s the price? Do you ship to country XYZ? Are there different colors/sizes/models available? Again, automation allows you to efficiently cover those basic and frequent comments, giving you and your team more time to focus on the meaningful stuff.

### Using social media as a small business:

### This is a similar case to the e-commerce example above. Say you’re running a little bistro, maybe you have more than one location. And, although your hours on operation can be found online, people will continue to ask: when are you open? You, of course, want to deal with these comments, because you care about your customers. But you might just miss a **time-sensitive question** (e.g. regarding a reservation) amidst all the generic inquiries – and perhaps even lose a customer because of it. Make sure you have the capacity to answer unique questions by using auto-comment solutions to deal with the volume of repetitive requests for basic information.

### Running Facebook contests:

### [Creating contests on your Facebook page](https://neilpatel.com/blog/create-facebook-contest/) is a great way to engage your audience. For a simple incentive, you can generate some serious traffic and expand your community. Usually, in social contests, participating users are required to comment with a specific keyword or hashtag. Instead of manually reviewing every entry, you can set up an auto filter that will react to certain words or phrases, and will help you sift out contest-worthy replies. Huge time saver.

**CHAPTER - 7**

**FUTURE SCOPE**

By this Proposed System, we enforce powerful techniques to achieve filtering of messages in OSN user walls. Naive-Bayesian is one of the best machine learning text classifier. It earned such popularity since it is well suited for online filtering such as spam mail filtering. This method is very easy to implement which reaps an excellent performance with very less complexity. Even though it is a very simple approach it provides an effective efficiency which cannot be satisfied by the existing algorithms. The future work concentrates on image and video filtering and further development can be done to change the learning process by making the machine itself to get trained by online learning mechanism which will replace the usage of database with terabytes of size since it is highly dynamic which cannot be weighed with small set of data.

**CHAPTER - 8**

**REFERENCES**

* <https://napoleoncat.com/blog/facebook-auto-comments/>
* <https://www.researchgate.net/publication/333058790_Spammer_Detection_A_Study_of_Spam_Filter_Comments_on_YouTube_Videos>
* <https://www.researchgate.net/publication/276437679_Automated_Message_Filtering_System_in_Online_Social_Network>
* <https://fullscreen.com/2015/08/13/filter-unwanted-youtube-comments/>
* <https://www.codeproject.com/Articles/23472/A-Naive-Bayesian-Spam-Filter-for-C>