Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 4, April 2017, pg.201 – 211

A STUDY ON LITERATURE SURVEY ON AUTOMATED FILTERING OF UNWANTED MESSAGES ON OSN

Ms. Shravani Krishna K[1]
PG Scholar, Department of Computer Science and Engineering,
New Horizon College of Engineering, Bangalore, Karnataka, India
[1] kamatham.shravani@gmail.com

Mr. Muralidhara S[2]
Sr.Assistant Professor, Department of Computer Science and Engineering,
New Horizon College of Engineering, Bangalore, Karnataka, India
[2] muralidhars@newhorizonindia.edu

Abstract: One fundamental issue in today's Online Social Networks (OSN's) is to give users the ability to control the messages posted on their own private space to avoid that unwanted content is displayed. Up to now, OSN's provide little support to this requirement. To fill the gap, in this paper, we propose a system allowing OSN users to have a direct control on the messages posted on their walls. This is achieved through a flexible rule-based system, which allows users to customize the filtering criteria to be applied to their walls, and a Machine Learning-based soft classifier automatically labelling messages in support of content-based filtering.

Keywords: Online social networks, Information filtering, short text classification, Trust.

INTRODUCTION

Social network is interactive medium to share and communicate some amount of data related to human life. OSN is used to share some content type. It could be image, text, video and audio. It is platform to build relationship among people who are interested in sharing views, picture, real time connections and texts.

Social network provide various types of services such as profile, social links. It allows you to create a list of user with whom you want to communicate, to create a public profile and view the connections within system.

Given architecture is three-tier architecture

1) First layer is called as Social Network Manager (SNM)-It aims to provide the basic ffunctionalitieses (i.e.) profile and relationship management.

2) Second layer provides support for external Social Network Applications (SNAs).

3) Third layer is Graphical user interface (GUI)-With the help of GUI user can interact with the system.

4) Filtering-It is used to filter the unwanted messages using blacklists.

5) Content Base filtering-It is used to select information based on the correlation between the content of the item and user performances.

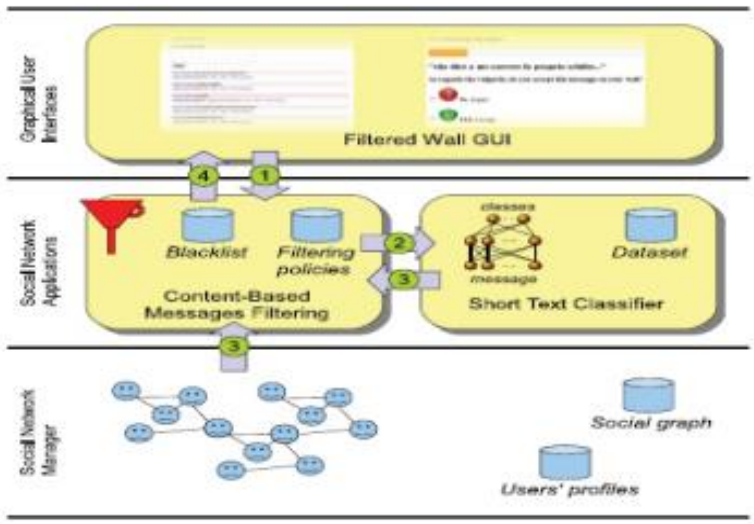

Fig.1: General Architecture of online Social Network

Some of the social network sites are Twitter, Facebook, and YouTube which are used worldwide. A social network can be defined as a set of factors and ties among them. Web mining is used to discover useful and related information from a large amount of data. Online social network information filtering can be used for a different purpose in OSN anyone can post on OSN wall as well as on others wall. With the help of information filtering user can able to control the messages written on their own wall by filtering out unwanted messages.

Currently OSN provides very less support to prevent not required messages on users wall for e.g. Facebook allows users to decide who is allowed to insert messages on their wall (i.e. Family members, friends, group of friends, friends of friends) However no-content based preferences are supported and hence it is not possible to prevent undesired messages such as vulgar or political one no matter who post them on the wall.

In our proposed system there are three methods message filtration by admin, message filtration by user and short text classifier. In message filtration by admin method the messages filtered by admin, Admin sets the word category. In message filtration by user messages are filtered by user, user sets the word category. In short text classifier short text word are set by admin in the data base.

Message filtering in this module, the unwanted messages are filtered. The other users that can send vulgar message to an OSN user is temporarily blocked by the OSN user. If the user sends vulgar messages that match

the filtering pattern specified by the OSN user more than a specified threshold value than that user is unfriend permanently.

Filtering Rules in defining the language for FRs specification, there are three main issues that, should affect a message filtering decision. First of all, in OSNs like in everyday life, the same message may have different meanings and relevance based on who write sit. As a consequence, FRs should allow users to state constraints on message creators. Creators on which a FR applies can be selected on the basis of several different criteria; one of the most relevant is by imposing conditions on their profile's attributes. Given the social network scenario, creators may also be identified by exploiting information on their social graph. This implies to state conditions on type, depth and trust values of the relationship creators should be involved in order to apply them the specified rules.

The same message on OSNs may have different meanings and relevance based on who write sit. It is necessary to apply constraints on messages. Constraints can be selected on several different criteria's. User can state what contents should be blocked or displayed on filtered wall by means of Filtering rules. Filtering rules are specified on the basis of user profile as well as user social relationship. FR is dependent on following factors,

1. Author
2. Creator Spec
3. Content Spec
4. Action

An authorize person who defines the rules. Creator Spec denotes the set of OSN user and Content Spec is a Boolean expression defined on content. Action denotes the action to be performed by the system on the messages matching content Spec and created by users identified by creator Spec.

Social Network Manager (SNM) The initial layer is Social Network Manager Layer provides the essential OSN functionalities (i.e., profile and relationship administration).It also maintains all the data regarding to the user profile. After maintaining and administrating all users data will provide for second layer for applying Filtering Patterns (FPs) and Black lists (BL).

Social Network Application (SNA) In second layer Content Based Message Filtering (CMBF) and Short Text Classifier is composed. Also we are detecting phishing links and filtering images posted on user walls in this layer. This is very important layer for the message, images and link categorization. Also Black list is maintained for the user who sends frequently bad words in message. Links are filtered and the user the alerted if phishing link detected. Images are scanned and if found hidden messages are displayed.

Graphical User Interface (GUI) Third layer provides Graphical User Interface to the user who wants to post his messages as a input and filtered wall is provided. In this layer Filtering Rules (FR) are used to filter the unwanted messages and provide Black list (BL) for the user who are temporally prevented to publish messages on user's wall.

Blacklists users are those users whose messages are banned from their contents. BL rules allow the wall owner to decide users to be blocked on the basis of their profiles and relationship with wall owner. This prevention can be done for a

specified period or forever according wall owner's desire. BL is dependent on author, creator specification and creator behaviour.

IMPLEMENTATION

In the software development there is always a need of the execution of the code to make the software work according to the need of the output based on the user. For any project to build up successfully there is a need of both support such as hardware and software so that the task can be done easily and accomplish the work to user satisfaction. For the phase of implementation there also need of the programming skills for the functions to work properly. The clustering of all this process is called implementation.

SOFTWARE REQUIREMENT

In the software requirement there is always a need of the coding skills for developing the software and understanding the problem due which we can frame a correct code to design the solution to the problem using the coding skills.

JAVA PROGRAMMING

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

Object oriented

Portable

Distributed

High performance

Interpreted

Multithreaded

Robust

Dynamic

Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer.

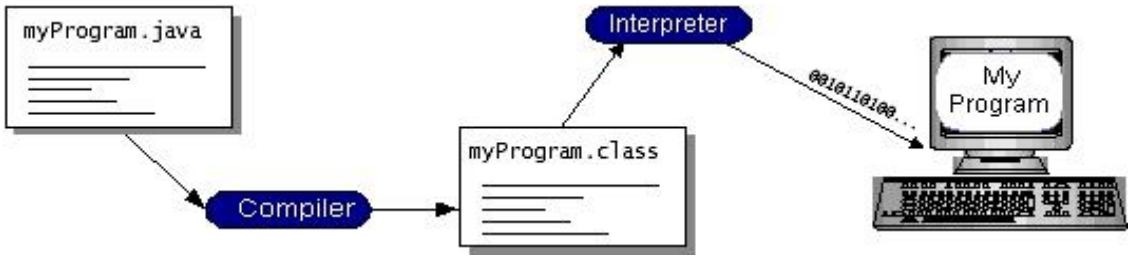

Fig 1. Compiler and interpreter flow

You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

THE JAVA PLATFORM

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

1. The Java platform has two components:
2. The *Java Virtual Machine* (Java VM)
3. The *Java Application Programming Interface* (Java API)
4. You've already been introduced to the Java VM. It's the base for the Java 5.Platform and is ported onto various hardware-based platforms.

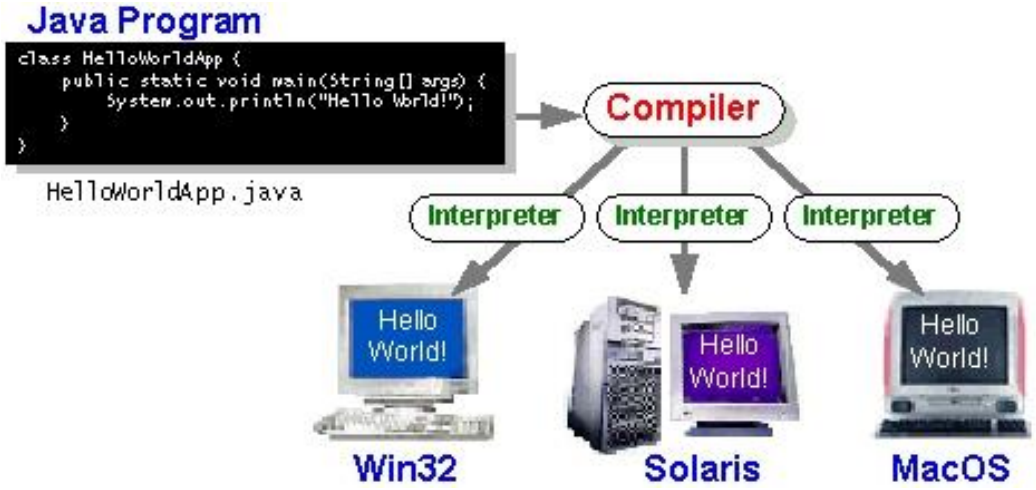

Fig 2 Byte code is platform independent

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

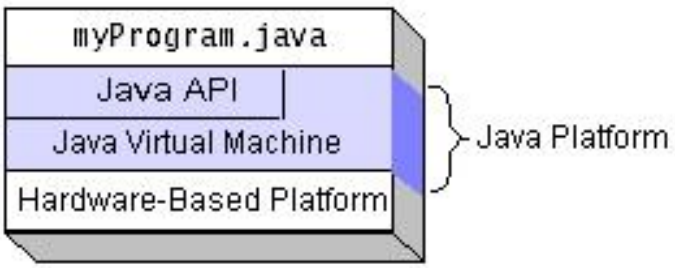

Fig 3 program that is running on java

ECLIPSE

Eclipse is an integrated development environment (IDE) used in computer programming, and it is widely used Java IDE It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written in Java and its use is for developing Java applications, but it can also be used to develop applications in other programming languages and its platform independent.

The basic steps described are as follows.

1. Create a new project

2. Mount a directory - specify a location to save project files

3. Add a new class to the project

4. Compile and run a Java program.

MYSQL

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to "generate" JDBC code and to hide many of JDBC's complexities from the end user.

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

The JDBC SQL API must "sit" on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and

vice versa. The usual SQL calls used by the programmer are simple *SELECT*'s, *INSERT*'s, *DELETE*'s and *UPDATE*'s, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

JAVA SERVLET PAGES

JSP document is a JSP page written in XML, a JSP document is also an XML document and therefore gives you all the benefits offered by the XML standard:

• You can author a JSP document using one of the many XML-aware tools on the market, enabling you to ensure that your JSP document is well-formed XML

• You can validate the JSP document against a document type definition (DTD).

• You can nest and scope namespaces within a JSP document.

• You can use a JSP document for data interchange between web applications and as part of a compile-time XML pipeline. In addition to these benefits, the XML syntax gives the JSP page author less complexity and more flexibility. For example, a page author can use any XML document as a JSP document. Also, elements in XML syntax can be used in JSP pages written in standard syntax, allowing a gradual transition from JSP pages to JSP documents.

Swings Most Swing developers know by now that Swing components have a separable model-and-view design. And many Swing users have run across articles saying that Swing is based on something called a "modified MVC (model-view-controller) architecture."

But accurate explanations of how Swing components are designed, and how their parts all fit together, have been hard to come by until now. To build a set of extensible GUI components to enable developers to more rapidly develop powerful Java front ends for commercial applications. The Swing established a set of design goals early in the project that drove the resulting architecture.

1 FILTERING PROCESS

In defining the language for FRs specification, we consider three main issues that, in our opinion, affect a message filtering decision. First, in OSNs like in everyday life, the same message may have different meanings and relevance based on who writes it. As a consequence, FRs should allow users to state constraints on message creators. Creators on which a FR applies can be selected on the basis of several different criteria; one of the most relevant is by imposing conditions on their profile's attributes. In such a way it is, for instance, possible to define rules applying only to young creators or to creators with a given religious/political view. Given the social network scenario, creators may also be identified by exploiting information on their social graph. This implies to state conditions on type, depth and trust values of the relationship(s) creators should be involved in order to apply them the specified rules. Fig.2. shows the filtering process.

The problem of setting thresholds to filter rules is also addressed, by conceiving and implementing within FW, an Online Setup Assistant (OSA) procedure. For each message, the user tells the system, the decision to accept or reject the message. The collection and processing of user decisions on an adequate set of messages distributed over all the classes allows computing customized thresholds representing the user attitude in accepting or rejecting certain contents. Such messages are selected according to the following process. A certain amount of non-neutral messages taken from a fraction of the dataset and not belonging to the training/test sets, are classified by the ML in order to have, for each message, the second level class membership values.

2 BLACKLISTING PROCESS

A further component of our system is a Blacklist (BL) mechanism to avoid messages from undesired creators, independent from their contents. BL is directly managed by the system, which should be able to determine who are the users to be inserted in the BL and decide when user's retention in the BL is finished. To enhance flexibility, such information is given to the system through a set of rules, hereafter called BL rules. Such rules are not defined by the Social Network Management, therefore they are not meant as general high level directives to be applied to the whole community. Rather, we decide to let the users themselves, i.e., the wall's owners to specify BL rules regulating who has to be banned from their walls and for how long. Therefore, a user might be banned from a wall, and at the same time, he will not be able to post in the wall.

3 ALGORITHM USED

Step 1 Start.

Step 2 A User tries post the message in a wall.

Step 3 Machine learning checks each word of the message.

Step 4 If (Words = = Good Words).

Step 5 Message is posted on the wall.

Step 6 Else if (Words = = Bad Words).

Step 7 Reject Bad Words using Blacklist and post the filtered message on the wall.

Step 8 Stop.

The above algorithm represents the concept of Machine Learning with the Blacklist. Firstly, a user is showing his interest in posting or commenting in other person's wall regardless of their relationship. He can post any message there without the filtering technique. But the Machine Learning here learns the message which is yet to be posted and finds whether it contains any vulgar or illegal words in it. If it can't find any illegal or vulgar words, then the system allows the message to be posted on the wall. If it finds any illegal or vulgar words in that message while learning it, then it will remove the vulgar words from the message and then insert those words in the Blacklist which stores the indecent words in it. Finally the system prints the message without the indecent words. This mechanism helps in preventing the users to get annoyed by the vulgar words in a public wall of the

Social Networking Sites. It does not prevent the unknown users from posting their messages; rather, it helps in preventing the obscenity with the vulgar words.

4 SHORT TEXT CLASSIFIER

Established techniques used for text classifications work well on datasets with large documents such as newswires corpora but suffer when the documents in the quantity are tiny. In this perspective critical features are the description of a set of characterizing and discriminant features allowing the representation of underlying concepts and the collection of a complete and consistent set of supervised examples. Our study is aimed at designing and evaluating various representation techniques in combination with a neural learning strategy to semantically categorize short texts. The first level task is conceived as a hard classification in which short texts are labelled with crisp Neutral and Non neutral labels. The second-level soft classifier acts on the crisp set of non-neutral short texts.

5 MACHINE LEARNING-BASED CLASSIFICATION

Short text categorization is a hierarchical two-level classification process. The first-level classifier does a binary hard classification that labels messages as Neutral and Non-Neutral. The first-level filtering task enables the subsequent second-level task in which a finer-grained classification is performed. The second-level classifier carries out a soft partition of Non-neutral messages assigning a given message a gradual membership to each of the non-neutral classes.

EXPERIMENTAL EVALUATION

An experiment is also conducted with the Machine Learning Technique with the use of good and bad words in the messages posted on the Social Networking Site Wall. It is conducted with the consideration of the authorized and unauthorized persons taking part in the posts and comments. The graph shows that both the authorized and unauthorized persons use any kind of words in posting the messages

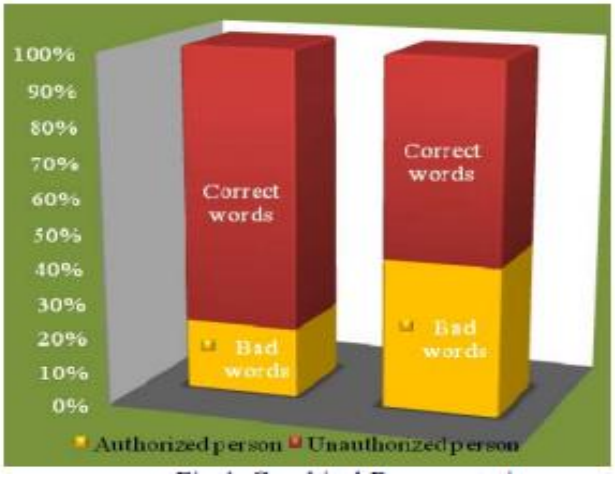

Fig 4 Graph representation

Good Words = ∑ Wi {M} ≠ ∑ BLw i=1 w=1 n n

Bad Words = ∑ Wi {M} = ∑ BLw i=1 w=1

Where, Wi ith word of the message M

M Input Message

BLw with word in the Blacklist

For Example, Blacklist words BLw = {Dog, Monkey, Buffalo, and Donkey}

Input words

M1 Hi Dog

M2 Monkey

M3 Buffalo

M4 Hi da Donkey what doing

OP=> (BL = = M1) = Hi

OP=> (BL = = M2) = No Message Received

OP=> (BL = = M3) = No Message Received

OP=> (BL = = M4) = Hi da what doing

Where, OP Output

Mi Message Input.

CONCLUSION

The proposed approach overcomes the disadvantage of the existing system, where there is no security for the messages that is being posted on the public walls of the user in online social networks. This focuses on preventing the indecent message from posting on the walls of the user, this is done using the machine learning techniques where this gives the result to the system by tracing the message. The user will be able to distinguish the good and bad words, authorized and unauthorized users in the social networking sites.

The machine learning techniques and the content based classification places vital role in the project , that will blacklist the bad words and unauthorized users .the user will have the control over the messages that has to displayed on his wall or not . Whatever the comment that is posted by the user, filtering wall first identifies the bad words based on the categories made and try to classify the bad and restricted words and displays only the message with good words. The bad words those are sent by the user will be blacklisted for permanently or for a particular period of time. So by this approach the obscenity of the user is prevented. This approach provides

more security towards spreading of vulgar or nonsense, indecent messages across online social networks, which causes panic situation among friends and relatives, with this approach more security will be provide for the posting of messages in the online sites.

Future improvement can be made by providing the security to posting of images and videos by steganography, where the message that is encoded in the images can be decoded by steganography that provides security by identifying hidden messages, that leads to terrorist activity.

REFERENCES

[1] Ying Chen, Yilu Zho, "Detecting Offensive Language in Social Media to Protect Adolescent Online Safety", ASE/IEEE International Conference on Social Computing, 2012.

[2] Marco Vanetti, Elisabetta Binaghi, Elena Ferrari, Barbara Carminati, and Moreno Carullo, "A System to Filter Unwanted Messages from OSN User Walls", IEEE Transactions on Knowledge and Data Engineer, February2013

[3] A Review on Filter Undesired Text from Social Networks International Journal of Computer Applications (0975 – 8887) Volume 107 – No 14, December 2014

[4] Filtration of unwanted messages from online social website ,.Multidisciplinary Journal of Research in Engineering and Technology, Volume 2, Issue 2, Pg.398- 402, 2015

[5] Filtering unwanted messages from osn walls 2016 1st International Conference on Innovation and Challenges in Cyber Security (ICICCS 2016)

