Project Report on

CONTENT FILTERING IN SOCIAL MEDIA

Submitted in partial fullfilment of
the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

by

Arun Kuruvila	B090003CS
Nithin V Nath	B090118CS
Vineeth Thomas Alex	B090498CS
Vivek Muraleedharan Nair	B090791CS

Under the guidance of
Ms. Lijiya A

Computer Science and Engineering
NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
Calicut, Kerala 673 601

Winter Semester 2013

Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT

Certificate

This is to certify that the project work entitled ”Content Filtering in Social Media”, submitted by the following students to National Institute of Technology Calicut towards partial fulfillment of the requirements of the award of Degree Of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the work carried out by him under my supervision and guidance.

Arun Kuruvila	B090003CS
Nithin V Nath	B090118CS
Vineeth Thomas Alex	B090498CS
Vivek Muraleedharan Nair	B090791CS

Project Guide
Ms. Lijiya A

Place : Calicut
Date : 29-04-2013

Head of Department

Office Seal

Abstract

Due to increasing collaboration of people from various regions, culture, and backgrounds on the Internet through social networking sites, the implementation of content filtering and censorship has been debated. Efficient methods needs to be devised to filter undesirable content from social media including violence, hate speeches, racist remarks and explicit content. Existing content filtering techniques are mainly focused on general web content primarily using meta-data of websites. There has been attempts in using machine learning in web content filtering such as decision trees, SVM, and neural networks. However with the increased participation in social networking sites like Facebook. Twitter etc, a collaborative strategy could be adopted where training the learning algorithm can be done by the active participation of users. Textual content may be used for classification and filtering in social media. Realistic solutions to this problem can lead to a more healthy networking on the Internet.

Problem Statement

We want to browse through social media which has become a critical part of our online persona without having to be exposed to harmful content such as hate speech, racism or pornography.

The nature of todays social medium requires a dynamic approach in content filtering techniques. Ignoring the need for genuine content filtering has caused large amount of confusion and has been seen as harmful in various scenarios.

The project aims to apply a content filtering mechanism that actively filters tweets from twitter with the help of machine learning techniques.

Contents

Chapter 1

Introduction

The importance of content filtering in social media is well documented. It has become increasingly important for many Web-related interactions. Social networking sites allow people to broadcast their thoughts without inhibtions. However there has been many instances where harmful content became widespread due to ineffective filtering. This is partly due to the fact that existing content filtering techniques cannot be easily extended to popular web-services, such as microblogs (example: twitter, tumblr), forums, etc., due to short nature of textual content. The goal of our work is to automatically classify above mentioned content in real-time into different categories so that users are not exposed to the harmful content.

The project seeks to protect the users from the problem of Internet abuse. Internet abuse is prevalent in the forms of racist speech or inflammatory material that tends to incite hate crimes, obscene text and violence. The problem is tackled using a *machine learning* (ML) approach similar to the one followed in [3], where incoming tweets are categorized into News, Events, Opinion, Deals and Private Messages.

Chapter 2

Text Categorization

Text categorization is the task of assigning a Boolean value to each pair $\langle d_j, c_i \rangle \in D \times C$, where D is a domain of documents and C = $\{c_1, ..., c_{|C|}\}$ is a set of predefined categories.Formally, the task is to approximate the unknown *target function* $\widehat{\Phi} : D \times C \rightarrow \{T, F\}$ (that describes how documents ought to be classified) by means of a function $\Phi : D \times C \rightarrow \{T, F\}$ called the *classifier* such that $\widehat{\Phi}$ and Φ "coincide as much as possible." [2]

2.1 Short Text Classification

With increase in popularity of online communication like chat-messages, tweets, comments, etc. classification of short text has become more relevant than "document" classification. These sources provide rich content but are very often prone to misbehaviour by a minority of users. The need has arisen to find novel techniques to classify short texts and thus improve online experience.

Classification[1] of short text messages is a hard task due to lack of content and context. Techniques that use word occurrence and its variations as features do not perform as well as it does on larger corpus of text. Using meta data of the text by using online tools like wikipedia or finding relevence of a pair of words in the tweet by using online search engines like google or bing would be infeasible and prone to latency. Hence, we cannot adopt these methods in a real-time context. So there is a need to research beyond using words as features.

[1] Since we are using classification to filter content, the terms will be used interchangeably in this document

Twitter, with its large and diverse user-base and up-to-date content, has established itself as a major social network. The techniques developed for classifying tweets can be extended to other content such as SMS, chat messages, YouTube comments, etc. with only minor modifications. The rest of the document will concentrate on classifying tweets.

2.2 Supervised and Unsupervised Learning

Supervised learning is the machine learning task of inferring a function from labeled training data. Here the key resource is the preclassified documents. In supervised algorithms, documents are classified into a predefned set of categories. On the other hand, unsupervised algorithms is used to seek out similarity between pieces of data in order to determine whether they can be characterized as forming a group (clusters).

2.3 Training Set, Test Set and Validation Set

- *training-and-validation set TV* The classifier Φ for categories is inductively built by observing the characteristics of tweets from *training set*. A *validation set* is used to tune the internal parameters of the classifier.The repeated tests of classifier aimed at parameter optimization are performed on this validation set.

- *test set Te* used for testing the effectiveness of the classifiers. Each $d_j \in Te$ is fed to the classifier, and the classifier decisions $\Phi(d_j, c_i)$ are compared with expert decisions $\widehat{\Phi}(d_j, c_i)$.

2.4 Indexing

Indexing is the procedure that maps a text into a compact representaion of its content. A text d_j is represented as a vector of term *weights* $\overrightarrow{d_j} = \langle w_{1j}, ..., w_{|\mathbf{T}|j} \rangle$ where $\mathbf{T}$ is the set of features and $0 \leq w_{kj} \leq 1$ represents how much the term t_k contributes to the semantics of the document.

2.5 Challenges

There is no perfect classifier. All the classifiers existing are prone to errors because of the intrinsic ambiguity present in natural languages. Most of the traditional ML techniques for the classification of text uses *term frequency* to classify data. These work well with large documents since word occurence is high. However the low word count in microblogs mean that they cannot be as effective. So a new approach based on existing techniques is necessary for accuracy.

Chapter 3

Literature Survey

Content filtering is an application of *text categorization* (TC). As with most cases in text categorization, *machine learning* (ML) is the preferred approach to content filtering.

Fabrizio Sebastiani, in his 2002 survey [2], has discussed in detail the main approaches to text categorization that fall within the machine learning paradigm. The paper formally defines TC and its various subcases and is followed by important applications of TC. He also notes the advantages of the ML approach over *knowledge engineering* approach to TC. The main ideas underlying the ML approach is also given. The major three steps of text classification- *text indexing*, inductive construction of a text classifier and evaluation of text classifiers, are explained in much detail.

The traditional TC methods may not be as efficient in classifying the current social media trend of short texts. Banerjee, et al. (2007) [1] has proposed a method to use Wikipedia to enrich the representation of short texts in blog feeds(RSS/Atom) inorder to cluster them. A Wikipedia dump was downloaded and feature generation was done after indexing it. They conducted an experiment on a snapshot of Google News homepage. The clustering done by using additional features from Wikipedia showed improved accuracy though not significant. The disadvantage of using such a data repository is the inability to capture up-to-date information and is especially unsuitable when the content consists of volatile data.

Another notable work in the field of short text classification is by Bollegala et al. (2006) [4] on measuring semantic similarity between words using web search engines. The work primarily focuses on integrating short text messages with Web search engines like Google, Bing to extract more information about the short text. For each pair of short text, they retrieve statistics on the engine results to determine the similarity score. However, these techniques give rise to ambiguity problems as the same word may have different meaning according to context.

Sriram,et al (2010) experimented a new approach in short text (*tweets*) classification to improve information filtering [3]. To address the limitations of traditional classification methods such as "Bag-Of-Words"(BOW), they proposed to use a set of domain-specific features extracted from the author's profile and text. They extracted 8 features in total which consisted of 7 binary features (Shortening of words and slangs (Binary), Time-event information (Binary), Opinions (Binary), Emphasis on words (Binary), Currency statistical information (Binary), Reference to another user at beginning of tweet (Binary), Reference to another user within tweet (Binary)) and authorship information. He conducted experiments with an available implementation of Naïve Bayes classifier in WEKA. The results obtained showed better accuracy when compared to BOW.

Joachims in his 1998 paper [7] explores the use of Support Vector Machines (SVMs) for learning text classifiers from examples. SVMs are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. His paper provides both theoretical and empirical evidence that SVMs are well suited for text categorization. In the conducted experiments the SVMs outperformed all other methods significantly. [7] concludes that SVMs make a very promising and easy to use method for text classification.

A Re-examination of Text Categorization Methods by Yiming Yang and Xin Liu [9] reports a controlled study with statistical significance tests on five text categorization methods: the Support Vector Machines (SVM), a k-Nearest Neighbor (kNN) classifier, a neural network (NNet) approach, the Linear Least-squares Fit (LLSF) mapping and a Naive Bayes (NB) classifier. They focus on the robustness of the methods in dealing with a skewed category distribution, and their performance as function of the training-set category frequency. The results show that SVM, kNN and LLSF significantly outperform NNet and NB when the number of positive training instances per category are small (less than ten), and that all the methods perform comparably when the categories are sufficiently common (over 300 instances).

In the comparative study of feature selection methods conducted by Yiming Yang and Jan Pederson [10], the focus is on aggressive dimensionality reduction. The study showed that the availability of a simple but effective means for aggressive feature space reduction may significantly ease the application of more powerful and computationally intensive learning methods to very large text categorization problems which are otherwise intractable.

Most of the existing techniques on short texts classification are based on web query or construction of a database. Web queries are prone to latency related problem and database has the problmm of out-dated information. Also feature expansion using web based techniques result in "curse-of-dimensionality". So classification has to be done using the internal features and minimal external features.

Chapter 4

Design

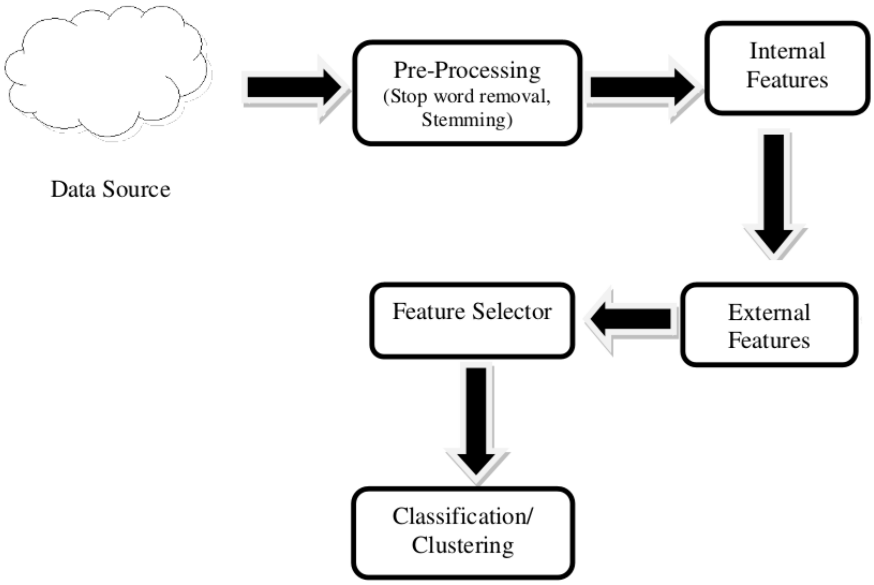

Figure 4.1: General design of short text classifier

4.1 Introduction

The project can be broadly divided into two stages

- Dataset collection and preprocessing.
- Construction of Classifier

4.2 Dataset Collection and Preprocessing

Dataset collection is a key step in all ML techniques. Preprocessing includes feature selection and extraction.

4.2.1 Dataset Collection

The initial dataset to be used will be tweets from Twitter. We collect tweets from a diverse group of users in fields ranging from sports, religion, politics, news, students, product dealers etc. We then preprocess these tweets and remove tweets satisfying the following conditions.

- Tweets which are not in English.
- Tweets which does not have any url and is less than three words.

4.2.2 Preprocessing

Stop-word removal and stemming are the main steps in preprocessing. Stop-word removal is done by checking the tweet with a corpus containing English stopwords. The resulting tweet then undergoes stemming. Stemming is the process of grouping words that share the same morphological root. [8] Porter's algortihm is the popular stemming algorithm used [8]. The result may vary with the efficiency of the stemming.

4.2.3 Feature Selection

We follow a feature selection method similar to the one followed in [3]. We have identified certain features which may effectively discriminate messages into the above mentioned categories. Our initial set of features were as follows

1. Authorship Information
2. Retweet
3. Presence of spam words
4. Presence of explicit words
5. Presence of words relating to targetted groups
6. Presence of hate words
7. Presence of URL

8. Presence of reference to another user
9. Presence of emphasized words
10. Shortening of words

In the authorship information, whether the author is a verified twitter user or not as the tweets from them tend to belong to safe category. Retweets may provide an indication that the tweet is not harmful as people generally retweets only the good tweets. Spams and obscene content almost always comes with a url. We intend to test the design on these feature set and optimize the feature set based on the output accuracy.

4.3 Construction of Classifier

The tweets in test set cannot participate in the inductive construction of the classifiers. After the evaluation is performed, the classifier is retrained on the entire initial corpus to enhance effectiveness. This is called *train-and-test* approach.[2]

A supervised learning technique is adopted for the filtering in this project. There has been instances where unsupervised learning was applied to text categorization [5], but the results were not encouraging.

As mentioned earlier, SVM is proven to be the best choice of learning method available. Support vector machines are based on the Structural Risk Minimization principle from computational learning theory. [7] argues that SVM offers two important advantages

- term selection is often not needed, as SVMs tend to be fairly robust to overfitting and can scale up to considerable dimensionalities
- no human and machine effort in parameter tuning on a validation set is needed, as there is a theoretically motivated, default choice of parameter settings, which has also been shown to provide the best effectiveness.

4.3.1 Support Vector Machine

The machine learning technique being used is SVM using Sequential Minimal Optimization (SMO) [11]. SMO is an algorithm for effectively solving optimization problem which arises during the training of support vector machines. SMO breaks the optimization problem in binary classification into a series of smallest possible sub-problems, which are then solved analytically.

Kernel Methods approach the problem of pattern analysis by mapping the data into a high dimensional feature space, where each coordinate corresponds to one feature of the data items, transforming the data into a set of points in a Euclidean space. In that space, a variety of methods can be used to find relations in the data.

Using *Kernel trick*, mapping does not need to be ever computed. If the algorithm can be expressed only in terms of a inner product between two vectors, all that is needed is to replace this inner product with the inner product from some other suitable space. Wherever a dot product is used, it is replaced by the Kernel function. The kernel function denotes an inner product in feature space.

Radial Basis Function (RBF) or the Gaussian Kernel is the most popular and versatile kernel function used in support vector machine. The performance of SVM c lassifier depends on the choice of the regularization parameter C and the kernel parameters. For RBF kernel the bandwidth parameter γ is the only kernel parameter. The standard radial Gaussian kernel is

$$k(x, z) = exp(-\gamma\|x - z\|^2) = exp(\frac{\|x - z\|^2}{2\sigma^2})$$

4.3.2 Categories

For the puposes of filtering out undesirable content, we can classify the tweets into two categories.

- Safe
- Unsafe

4.4 Application Design

The above design is implemented to develop an interactive Twitter application. The application allows users to login and view their timeline and messages. The tweets displayed will be only those classifeid as safe.

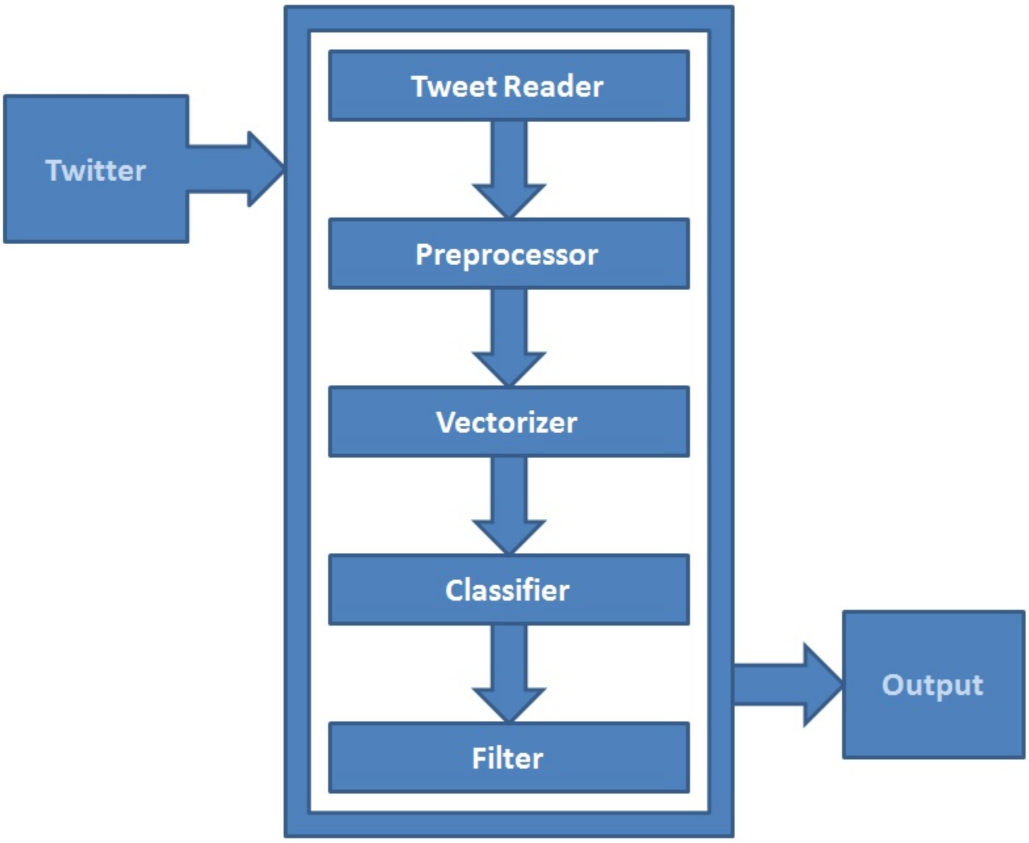

Figure 4.2: Components

4.4.1 Components

Figure 4.2 shows various components that comprise the application. The final aim of the application is to filter out unwanted tweets and display only those categorised as *safe*.

A user logs into the application and authenticates the use of their Twitter account. The application retrieves the tweets from Twitter. The tweets are then passed through the following logical components.

Tweet Reader

Tweet reader component reads the tweets and the associated fields it contains. A tweet contains several fields in addition to the status and username such as co-ordinates, retweet-count, withheld-in-countries,etc. The fields relevant to the application is selected and forwarded to the next component.

User information is also retrieved during this stage.

Preprocessor

In this component, various pre-processing tasks are done. Stop word removal is done by matching the words in the tweet with a predefined corpus containing English stopwords. Stemming is done by implementing a version of Porter's algorithm.

Vectorizer

The feature set defined in previous section is used to convert the text data into a vector of 10 fields. This is done with the help of corpora for spam words, explicit words, hate words and words relating to targetted groups.

Classifier

The classifier categorizes the vector into one of the categories. SVM classifier is chosen for this purpose.

Filter

This is a simple component that filters out tweets that are not categorized as safe.

Chapter 5

Work Done

Python is being used as the programming language for the project. The work can be broadly divided into following interdependent parts.

5.1 Dataset Collection

Dataset collection is the most time consuming and tedious part of the project. It also serves as the foundation on which the classifier is built.
Dataset was collected by identifying the users from diverse fields. Users whose tweets contain sufficient examples of each of the categories were identified. This list of users is then fed into a crawler that copies the latest tweets as well as details of these users and stores the results in a database for future use.

A total of **6320** tweets were collected of which only **5221** tweets were fit to be used in the training set. The remaining were either in languages other than English or contained too few characters. Of the 5221 tweets, 2837 were manually found out to be safe and 2384 were in the unsafe category.

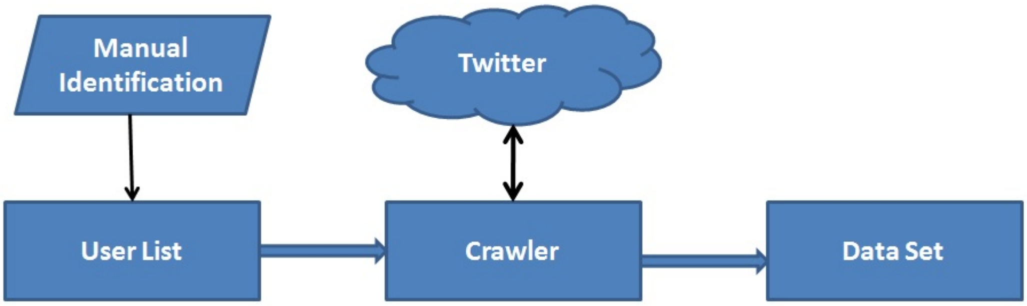

Figure 5.1: Components

The Tweet-crawler uses OAuth protocol to send secure authorized requests to the Twitter API. After establishing a connection with the server, a fixed number of tweets each from user is pulled. The code for this has been completed and is performing as expected.

5.1.1 Training Set and Testing Set

For generating the training set, we manually labeled selected items from the data set as to whether these items are to be filtered or not. An auxiliary application coded in Python was developed to assist in manually labeling large number of items from the training data set. While labeling is labor-intensive and unlike a programmers work it is one of the central ingredients to the competent functioning of the classifier. The effectiveness of the machine learning phase depends entirely on how it is trained and the diversity and consistency of the labeled elements in the training set.

5.2 Preprocessing

The preprocessing stage mainly involves natural language processing and the code for this stage has been completed.
Stopword removal is done with the help of WordNet Corpus on English stopwords available in the Natural Language Toolkit of python. A list of important words is finally forwarded from this part.
Stemming is an important and difficult stage of preprocessing. Converting each word into its morphological root is challenging and often inaccurate. An implementation of Porter's algorithm in python has been used to achieve this.

5.3 Machine Learning

5.3.1 Feature Vector

All the features were succesfully extracted from the tweet obtained after preprocessing.

Presence of spam words, Presence of explicit words, Presence of words relating to targetted groups and Presence of hate words are identified using the help of various corpuses that has been compiled specifically for the purpose of this project. Special care has been taken to match the words and phrases in the corpus to the usage found in online media.

Presence of URL and Presence of emphasized words are detected using regular expressions. Presence of emphasized words are found by the usage of repetition of letters.

Authorship Information, Retweet, Presence of reference to another user and Retweet count are taken directly from the fields provided by Twitter API.

5.3.2 Classifier

A classifier was built using SVM with Radial Basis Function (RBF) as Kernel. Hyperparameter (C and γ) optimization was done using grid search. Grid search is an exhaustive searching through a manually specified subset of the hyperparameter space. Cross-validation on the training was used as the performance metric. The optimum values were found as $C = 4.0$ and $\gamma = 0.5$

The initial feature set gave only 73% of accuracy with only 23% of the unsafe tweets correctly classified as unsafe. An attribute evaluation was done after which the features Shortening of Words and Presence of Emphasis were found to have a detrimental effect on classification. Adding two new features *presence of photo* in the tweet and *retweet count* also helped to improve the overall performance.

5.4 Twitter API

Communicating with Twitter is done using the Twitter API. An application was registered with Twitter to allow tweet retrieval and other actions. The application does the following.

- The application uses the consumer token and consumer secret provided by Twitter to generate a URL
- The user authenticates access to the application by visiting the page and entering the PIN displayed onto the application.
- The application verifies the entered PIN and Twitter grants the application access to the user's profile.
- The application retrieves tweets from the user's timeline.

Chapter 6

Result

The classifier achieved an accuracy of 85% during 10 fold cross validation. The precision was 85.9% and recall 86%. The number of support vectors used to build the SVM was 4307.

Class	TP Rate	FP Rate	Precision	Recall
Safe	0.915	0.247	0.878	0.915
Unsafe	0.753	0.085	0.82	0.753
Weighted	0.86	0.192	0.859	0.86

Table 6.1: Result Summary

Tweets implying racial discrimination and hate speech were found to be the majority in falsely classified instances. Most of the tweets containing profanity and explicit content were correctly classified.

The features *Verified User* and *Profanity* were found to be the most useful among all the features. *Emphasis*, *Shortening of Words* and *Presence of URL* were found to be least useful.

The diversity in the usage of shortforms of words and mistakes in spelling proved to be one of the main obstacles in correctly identifying unsafe tweets. Detecting sarcasm and reference to certain events, etc. also proved too difficult for the classifier to identify. A classifier which is supplied with more background information will be able to perform better.

Chapter 7

Future Work

A more efficient set of features may be identified with repeated analysis of the data available about each tweet. Analysing the correlation between each of the feature and the class allotted can prove to be very useful to improve the classifier. This work can be extended to other areas like Youtube Comments, Facebook, Forums etc. Usage of an external source for extracting background knowledge about tweets will help in the classification of tweets pertaining to global events and current affairs.

As the final stage a fully fledged web application (user interface) can be deployed. The application will allow users to view their Twitter timeline akin to the orginal site. The level of filtering may also be set by the user. Viability of a feedback system that actively responds to error in classification and incorporates it to the application is also to be checked.

References

[1] Banerjee, S., Ramanthan, K., and Gupta, A. *Clustering short texts using Wikipedia.* In Proc. SIGIR (Amsterdam, The Netherlands, July 2007), 787-788.

[2] Sebastiani, F., *Machine learning in automated text categorization* ACM computing surveys(CSUR) (Volume 34 Issue 1, March 2002), 1-47.

[3] Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., Demirbas, M., *Short text classification in twitter to improve information filtering.* In Proc. SGIR (Geneva, Switzerland July 2010), 841-842.

[4] *Measuring semantic similarity between words using web search engines*

[5] Ko, Y., Seo, J., *Automatic text categorization by unsupervised learning.* In Proc. COLING (Saarbrücken, Germany, August 2000), 453-459.

[6] Lewis, D., *Naive (Bayes) at forty: The independence assumption in information retrieval.* In Proc. ECML (Chemnitz, Germany, April 1998)

[7] Joachims, T., *Text categorization with Support Vector Machines: Learning with many relevant features.* In Proc. ECML (Chemnitz, Germany, April 1998)

[8] http://en.wikipedia.org/wiki/Stemming

[9] Yiming Yang and Xin Liu, *A re-examination of text categorization methods*

[10] Yiming Yang and Jan O. Pedersen *A Comparative Study on Feature Selection in Text Categorization*

[11] John C. Platt, *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines.* ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING 1998