IJERST

International Journal of Engineering Research and Science & Technology

ISSN : 2319-5991
Vol. 2, No. 4
November 2013

www.ijerst.com

Email: editorijerst@gmail.com or editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 2, No. 4, November 2013

Research Paper

ON LINE SOCIAL NETWORK CONTENT AND IMAGE FILTERING, CLASSIFICATIONS

Prashant Tomer[1]*, Shrikant Lade[1], Manish Kumar Suman[2] and Deepak Patel[3]

*Corresponding Author: **Prashant Tomer** ✉ ptomer40@yahoo.com

Since the textual contents on online social media are highly unstructured, informal, and often misspelled, existing research on message-level offensive language detection cannot accurately detect offensive content, and user-level offensiveness evaluation is still an under researched area. To bridge this gap, This paper elaborate few systems which detect offensive content and identify potential offensive users in social media. enforcing content based message filtering conceived as a key service for On-line Social Networks (OSNs). The system allows OSN users to have a direct control on the messages posted on their walls. This is achieved through a flexible rule-based system, that allows a user to customize the filtering criteria to be applied to their walls, and a Machine Learning based soft classifier automatically producing membership labels in support of content-based filtering.

Keywords: Content filtering, Message classification, Policy-based personalization, Just check

INTRODUCTION

In the last years, On-line Social Networks (OSNs) have become a popular interactive medium to communicate, share a considerable amount of human life information. Daily and continuous communication implies the exchange of several types of content, including free text, image, audio and video data. With the rapid growth of social media, users especially adolescents are spending significant amount of time on various social networking sites to connect with others, to share information, and to pursue common interests. In 2011, 70% of teens use social media sites on daily basis (Ali, 2007) and nearly one in four teens hit their favorite social-media sites 10 or more times a day (Amati and Crestani, 1999). While adolescents benefit from their use of social media by interacting with and learning from others, they are also at the risk of being exposed to large amounts of offensive online contents. A main part of social network content is constituted by short text, a notable example are the messages permanently written by OSN users on particular public/private areas, called in general walls.

[1] Information and Technology Branch of RKDF Institute of Science and Technology Bhopal affiliated by Rajeev Gandhi University Bhopal.

[2] Computer Science and Engineering Branch of Millennium Institute of Technology Bhopal affiliated by Rajeev Gandhi University Bhopal.

[3] Information and Technology Branch of RKDF Institute of Science and Technology Bhopal affiliated by Rajeev Gandhi University Bhopal.

The Children's Internet Protection Act (CIPA) was enacted in early 2001 to address concerns on children's access to visual offensive content over Internet. While CIPA concerns about image contents, offensive languages in the form of unstructured and despicable texts can be as harmful as multimedia materials. To comply with CIPA requirements, administrators of social media often manually review online contents to detect and delete offensive materials. However, the manual review tasks of identifying offensive contents are labor intensive, time consuming, and thus not sustainable and scalable in reality. Some automatic content filtering software packages, such as Appen and Internet Security Suite, have been developed to detect and filter online offensive contents. Most of them simply blocked webpages and paragraphs that contained dirty words. These word-based approaches not only affect the readability and usability of web sites, but also fail to identify subtle offensive messages. For example, under these conventional approaches, the sentence "you are such a crying baby" will not be identified as offensive content, because none of its words is included in general offensive lexicons. In addition, the false positive rate of these word-based detection approaches is often high, due to the word ambiguity problem, i.e., the same word can have very different meanings in different contexts.

CONTENT FILTERING TECHNIQUES

This section, presents methods on offensive content filtering in social media, and then focus on text mining based offensive detection research.

a. Offensiveness Content Filtering Methods in Social Media Popular online social networking sites apply several mechanisms to screen offensive contents. For example, You-tube safety mode, once activated, can hide all comments containing offensive languages from users. But pre-screened content will still appear—the pejoratives replaced by asterisks, if users simply click "Text Comments." And on Facebook, users can add comma-separated keywords to the "Moderation Blacklist." When people include blacklisted keywords in a post and/or a comment on a page, the content will be automatically identified as spam and thus be screened. Twitter client, "Tweeter 1.3," was rejected by Apple Company for allowing foul languages to appear in users' tweets. Currently, Twitter does not prescreen users' posted contents, claiming that if users encounter offensive contents, they can simply block and unfollow those people who post offensive contents. In general, the majority of popular social media use simple lexicon-based approach to filter offensive contents. Their lexicons are either predefined (such as Youtube) or composed by the users themselves (such as Facebook). Furthermore, most sites rely on users to report offensive contents to take actions. Because of their use of simple lexicon-based automatic filtering approach to block the offensive words and sentences, these systems have low accuracy and may generate many false positive alerts. In addition, when these systems depend on users and administrators to detect and report offensive contents, they often fail to take actions in a timely fashion. For adolescents who often lack cognitive awareness of risks, these approaches are hardly effective to prevent them from being exposed to offensive contents. Therefore, parents need more sophisticate software and techniques to efficiently detect offensive contents to protect their adolescents from potential exposure to vulgar, pornographic and hateful languages.

Using Text Mining Techniques to Detect Online Offensive Contents Offensive language identification in social media is a difficult task because the textual contents in such environment is often unstructured, informal, and even misspelled. While defensive methods adopted by current social media are not sufficient, researchers have studied intelligent ways to identify offensive contents using text mining approach. Implementing text mining techniques to analyze online data requires the following phases:

1) Data acquisition and preprocess

2) Feature extraction

3) Classification

The major challenges of using text mining to detect offensive contents lie on the feature selection phrase, which will be elaborated in the following sections.

Message-level Feature Extraction

Most offensive content detection research extracts two kinds of features: lexical and syntactic features. Lexical features treat each word and phrase as an entity. Word patterns such as appearance of certain keywords and their frequencies are often used to represent the language model. Early research used Bag-of-Words (BoW) in offensiveness detection (Trier and Jain, 1995). The BoW approach treats a text as an unordered collection of words and disregards the syntactic and semantic information. However, using BoW approach alone not only yields low accuracy in subtle offensive language detection, but also brings in a high false positive rate especially during heated arguments, defensive reactions to others' offensive posts, and even conversations between close friends. N-gram approach is considered as an improved approach in that it brings words' nearby context information into consideration to detect offensive contents (Pratikakis *et al.,* 2011). N-grams represent subsequences of N continuous words in texts. Bi-gram and Tri-gram are the most popular N-grams used in text mining. However, N-gram suffers from difficulty in exploring related words separated by long-distances in texts. Simply increasing N can alleviate the problem but will slow down system processing speed and bring in more false positives.

Syntactic features: Although lexical features perform well in detecting offensive entities, without considering the syntactical structure of the whole sentence, they fail to distinguish sentences' offensiveness which contain same words but in different orders. Therefore, to consider syntactical features in sentences, natural language parsers (Chen *et al.,* 2008) are introduced to parse sentences on grammatical structures before feature selection. Equipping with a parser can help avoid selecting unrelated word sets as features in offensiveness detection.

User-Level Offensiveness Detection

Most contemporary research on detecting online offensive languages only focus on sentence-level and message-level constructs. Since no detection technique is 100% accurate, if users keep connecting with the sources of offensive contents (e.g., online users or websites), they are at high risk of continuously exposure to offensive contents. However, user-level detection is a more challenging task and studies associated with the user level of analysis are largely missing. There are some limited efforts at the user level. For example, Kontostathis *et al.* [Liou, 2006] propose a rule-based communication model to track and categorize online predators. Pendar [Pratikakis, 2006] uses lexical features with machine learning classifiers to differentiate victims from predators in online chatting environment. Pazienza and Tudorache Collobert R and Weston J A (2008) propose utilizing user profiling features to detect aggressive discussions. They use users' online behavior histories (e.g., presence and conversations) to predict whether or not users' future posts will be offensive. Although their work points out an interesting direction to incorporate user information in detecting offensive contents, more advanced user information such as users' writing styles or posting trends or reputations has not been included to improve the detection rate.

CONTENT CLASSIFICATION

The problem of applying content-based filtering on the varied contents exchanged by users of social networks has received up to now few attention in the scientific community. One of the few examples in this direction is the work by Boykin and Roychowdhury [Gllavata *et al.,* 2004] that proposes an automated anti-spam tool that, exploiting the properties of social networks, can recognize unsolicited commercial e-mail, spam and messages associated with people the user knows. However, it is important to note that the strategy just mentioned does not exploit ML content-based techniques.

Spam classification can be seen as a subset of the larger field of text classification. Text classification (or categorization) is the task of assigning a piece of data, such as a text document d_i to one (or more) predefined categories {C1,C2....Cm}. The value of {$a_{ij\}}$ represents the decision to classify document j under category i. For example, imagine a set of various news articles that needs to be divided into appropriate categories, such as politics or sports. The task of classification is to derive rules that accurately organize these articles into these groups, based, in general, solely on their contents. In the specific case of spam classification, there are only two categories, spam and non-spam (legitimate or ham"). Furthermore, these two categories are exclusive, as an e-mail cannot be both spam and non-spam. This is not always the case in other classification problems, as a certain news article can be filed under multiple categories. Extending the example of a collection of news articles, some derived rules could be:

if (ball AND racquet) OR (Wimbledon), then confidence ("tennis" category) = 0.9 confidence ("tennis" category) = 0.3 * ball + 0.4 * racquet + 0.7 * Wimbledon

Information Retrieval (IR) was the first application of automated classification and motivated much of the early interest in the field (Ali, 2007). This has led to extensive research and ever more accurate techniques for categorizing text documents. In addition to automatic indexing for IR, other techniques have been developed and other applications for this technology have been found, including document filtering and routing (Amati and Crestani, 1999), authorship attribution (Jung *et al.,* 2004), word sense disambiguation and general document organization.

This research has led to more automation and less human interaction, through the application of machine learning techniques to the categorization task. The machine learning approach relies on a corpus of documents, for which the correct classification is known. This corpus is divided in two, non-overlapping sets: (a) the training set: a set of pre-classified documents, that is used to teach the classifier the characteristics that define the category (a category profile); and (b) the test set: a set of documents that will be used to test the effectiveness of the classifier built using the training set.

Furthermore, the training set can consist of both positive and negative examples, several techniques of content-based spam classification. This will include statistical classifiers, collaborative classifiers and combinations of different classifiers.

Statistical Classifiers

The Naive (or simple) Bayesian classifier (NB) is arguably the most well-known and commonly used statistical spam classifier. This is considered a probabilistic classifier, as it estimates the probability that a document d_j belongs to class c_i given the features present or not present in vector of the document (Ali, 2007). For instance, consider the situation of a formal letter beginning with the words to whom it may. The probability that the next word is concern" is intuitively far greater than the probability that the next word is hurricane" or banana. However, this method ignores those interdependencies as a matter of convenience, as both computing and making practical use of the stochastic dependence between terms is computationally hard" (Jung *et al.,* 2004). Despite this simple design, this classifier is surprisingly effective. Although, NB is not the most accurate technique, there are several proposals to improve this method by experimenting with the feature selection stage [Amati and Crestani, 1999]. As with most statistical classifiers, NB requires a training phase using the pre-classified documents in the training set.

Another approach to spam classification uses genetic programming [Gllavata *et al.,* 2004]. Genetic programming models itself after the evolutionary process in nature: survival of the fittest. "In this case, the fittest" is the solution that can most accurately classify the training set of messages. Each solution is represented by a tree, containing the classification rules, such as the presence or absence of a certain feature. A fitness function is used to measure how well each solution performs. The fittest solutions are allowed to reproduce, using a crossover function. A crossover function combines aspects of two trees to produce a child" solution. Additionally, a mutation function is sometimes applied to a child to introduce parts of the solution not inherited from either parent. This process continues until the optimal solution is reached.

In the case of spam classification, feature detectors are used to score an e-mail (Ali, 2007). These take the form of empirical rules that return a numerical value. The fitness function compares the probability that a correct solution would have been reached to the known classifications of the training set and allows the best solutions to continue the evolutionary process until a certain threshold of accuracy is reached. Experiments conducted by Amati and Cristani (1999) show that genetic classifiers can achieve promising results.

A Neural Network (NN) classifier is a network of units connected to each other based on dependence relations (Collobert R and Weston J A, 2008). Document features serve as input units and document categories as output units. Relatively little research has been performed on using the classification power of NN for spam classification. This is due, in part, to the large amount of time needed for feature selection and training. However, (Trier and Jain, 1995) propose an NN-based classifier called LINGER that achieved 100% accuracy in their experiments. In this system, all words were retained that occurred at least twice in the training set. No additional preprocessing was performed, such as stopword removal or word-stemming. Similar experiments by Pratikakis *et al.* (2011) show that removing stopwords and performing word-stemming actually improved classification accuracy.

The k-Nearest Neighbor (kNN) classifier is considered an example-based classifier, that means that the training documents (examples) are used for comparison rather than an explicit category representation, such as the category profiles used by other classifiers. As such, there is no real training phase. When a new document needs to be categorized, the k most similar documents (neighbors) are found and if a large enough proportion of them have been assigned to a certain category, the new document is also assigned to this category, otherwise not Pratikakis *et al.* (2011). Additionally, finding the nearest neighbors can be quickened using traditional indexing methods (Chen, 2008). As all of the training examples are stored in memory, this technique is also referred to as a memory-based classifier (Liu, 2006).

Support Vector Machines (SVM) represent one of the most advanced and promising classifiers to date. This approach can be understood by plotting all positive and negative examples in a multidimensional space and then finding the plane ¾$_i$ that separates positive from negative examples with the widest margin (Ali, 2007). This is visualized in Figure 1. This approach integrates both dimension reduction, as well as classification, thus no separate feature selection phase is needed. Another advantage is that SVMs are insensitive to the relative numbers of each class, as outlying examples will seldom affect the best dividing line between positive and negative examples. If a plane cannot be found that separates the two classes, the hyperspace can be extended to more dimensions, thus insuring the existence of a separating plane (Chen, 2008).

Collaborative Classifiers

Collaborative spam filters are based on the beliefs that most users agree on which messages constitute spam and that many users receive the same spam message (Collobert R and Weston J A, 2008). Collaborative filters capitalize on these understandings to combine the collective classifying power and accuracy from a community of users to form a super-classifier. Essentially, when a user identifies a message as spam, either manually or using an automated classifier, a signature is computed for that particular message and is shared with other users. When a new message arrives at a user, the signature is computed and compared to the shared database of signatures. If a match is detected, the message is immediately treated as spam. Several available collaborative filters are Distributed Checksum Clearinghouse, Vipul's (http://www.rhyolite.com/dcc/) Razor, Cloudmark and SpamWatch (http://www.cs.berkeley.edu/~zf/spamwatch/). Collaborative filters have two basic features in common: (1) a mechanism for creating message signatures; and (2) a mechanism for sharing these signatures with other users in the community.

It is preferable to use message signatures, rather than the original messages, not only in order to lower bandwidth cost, but also to preserve privacy. To create a message signature, an algorithm is needed that is non-reversible and is robust to small differences in text, such as the salutation line of a personalized message. The Approximate Text Addressing (ATA) algorithm provides such functionality (Trier and Jain, 1995).

Another popular method is to compute the hash digest of the e-mail using SHA or similar hash function (J Kong P O *et al*., 2005). Unfortunately, spammers can fool this by inserting a few random characters into the message, thus resulting in a different hash value that will not be matched by the signature database. To counter these hash busters, proposes an open digest technique that (1) does not vary significantly for changes that can be produced automatically; (2) is robust against intentional attacks, and (3) has an extremely low risk of false positives. The hash function proposed produces similar digests for similar documents, despite random text additions, thesaurus substitutions and perceptive substitutions, such as replacing letters with numbers (e.g., s3cur1ty).

Collaborative filters can be implemented both at the ISP level or the user level. An example of an ISP level collaboration is the alliance-based framework presented by (S Brin and L Page 1998). This framework assumes a community of trusted mail servers that exchange a set of global rules, generated by a genetic algorithm. These rules are further improved by user feedback that punishes bad rules and rewards good rules. Experimental results show that the alliance increases classification accuracy; however, spam classification processing time is considered too long.

Many collaborative filters rely on a centralized database of message signatures; however, several proposals have been made for a decentralized model. The CASSANDRA architecture introduced by [Gilayati *et al.,* 2004] uses an adaptive, resilient, decentralized and scalable peer-to-peer (P2P) overlay network to share spam signatures between users. To lower bandwidth consumption, spam signatures are only sent to users that are most likely to need them, based on past interactions and relationships. Similar P2P collaborative filters are discussed by (E Damiani 2004), the latter of which combines agents, whitelisting, distributed hash tables and SVMs.

The collaborative filter introduced by (W Yerazunis 2004) uses global social e-mail networks, instead of a centralized database or P2P network, to achieve very high spam detection rates. Social e-mail networks provide several useful foundations for spam filtering, including an existing network infrastructure with no additional overhead, an existing list of participants and an implied trust and reputation model based on communication history. The proposed system requires no additional costly network connections and the list of peers is maintained by the list of e-mail contacts (address book) of the user. Regarding trust, a distributed power iteration algorithm is used to obtain a trust score for each contact called mailtrust, in much the same way as Google's PageRank (W Yerazunis 2004).

Combining Classifiers

A classifier committee consists of multiple classifiers (usually an odd number) that classify the same documents and then take the majority vote for the appropriate classification. A multilevel classifier committee is referred to as stacking. This technique uses multiple classifiers as members to induce a higher-level classifier with improved performance. This classifier can be thought of as the president of the committee. Each new message is first classified by the members of the committee and then given to the president. The president considers the results of the members along with the results of its own classification to produce the final classification. The benefit of this construction is that different classifiers often make different mistakes and the president can be trained to know when to listen to which classifier.

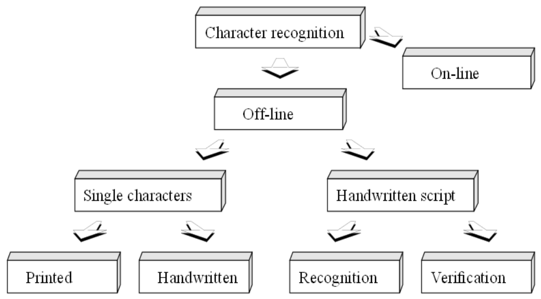
Hybrid classifiers consist of multiple classifiers that are targeted at different parts of a document to reach higher effectiveness. This can combine both intrinsic metadata, such as the results of statistical analysis by one of the classifiers described above, as well as extrinsic metadata, such as the origin of the message. The open source spam filter, Spam Assassin is an example of a hybrid classifier as it combines a rule-based classifier created using genetic programming with an NB classifier trained on user examples. An example of a spam report generated by SpamAssassin, including the rules applied and the points assigned is shown in Figure 2. These rules can include: (1) checks for suspicious words or phrases; (2) content and layout checks, such as hidden HTML code; (3) blacklisting of the sender, and (4) statistical classification performed by an NB classifier trained on user-provided examples.

Hybrid design offers the ability to combine the best strengths of different classifiers. This strength is demonstrated in Figure 2 as the message was correctly classified as spam, despite the NB classifier component incorrectly classifying it as non-spam. Another hybrid is proposed by (D Gavrilis, 2006) that combines genetic programming for feature selection with a neural network classifier. A similar approach is followed by SpamGuru (http://spamassassin.apache.org), that combines an NB classifier, an advanced pattern matching algorithm and an origin-based filter.

Optical Character Recognition

Content of the user can be filter and classify by above mention method. But as the social media communicate by text and image as well. So one of the medium of offensive language could be image. Now it is necessary to check these images for the same things. So to check these images one has to apply Optical Character Recognition (OCR) as by this characters of image can be identify can and distinguish. Once that words are identify then it is easy to find that whether that image is to reject or not.

Figure 1: Characters Recognition Fields



The first true OCR reading machine was installed at Reader's Digest in 1954. This equipment was used to convert typewritten sales reports into punched cards for input to the computer.

First Generation OCR

The commercial OCR systems appearing in the period from 1960 to 1965 may be called the first generation of OCR. This generation of OCR machines were mainly characterized by the constrained letter shapes read. The symbols were specially designed for machine reading, and the first ones did not even look very natural. With time multifont machines started to appear, which could read up to 10 different fonts. The number of fonts were limited by the pattern recognition method applied, template matching, which compares the character image with a library of prototype images for each character of each font.

Second Generation OCR

The reading machines of the second generation appeared in the middle of the 1960's and early 1970's. These systems were able to recognize regular machine printed characters and also had hand-printed character recognition capabilities. When hand-printed characters were considered, the character set was constrained to numerals and a few letters and symbols. The first and famous system of this kind was the IBM 1287, which was exhibited at the World Fair in New York in 1965. Also, in this period Toshiba developed the first automatic letter sorting machine for postal code numbers and Hitachi made the first OCR machine for high performance and low cost. In this period significant work was done in the area of standardization. In 1966, a thorough study of OCR requirements was completed and an American standard OCR character set was defined; OCR-A. This font was highly stylized and designed to facilitate optical recognition, although still readable to humans. A European font was also designed, OCR-B, which had more natural fonts than the American standard. Some attempts were made to merge the two fonts into one standard, but instead machines being able to read both standards appeared

Third Generation OCR

For the third generation of OCR systems, appearing in the middle of the 1970's, the challenge was documents of poor quality and large printed and hand-written character sets. Low cost and high performance were also important objectives, which were helped by the dramatic advances in hardware technology. Although more sophisticated OCR-machines started to appear at the market simple OCR devices were still very useful. In the period before the personal computers and laser printers started to dominate the area of text production, typing was a special niche for OCR. The uniform print spacing and small number of fonts made simply designed OCR devices very useful. Rough drafts could be created on ordinary typewriters and fed into the computer through an OCR device for final editing. In this way word processors, which were an expensive resource at this time, could support several people and the costs for equipment could be cut.

OCR – Pre-Processing

These are the pre-processing steps often performed in OCR Binarization

– Usually presented with a grayscale image, binarization is then simply a matter of choosing a threshold value.

Morphological Operators

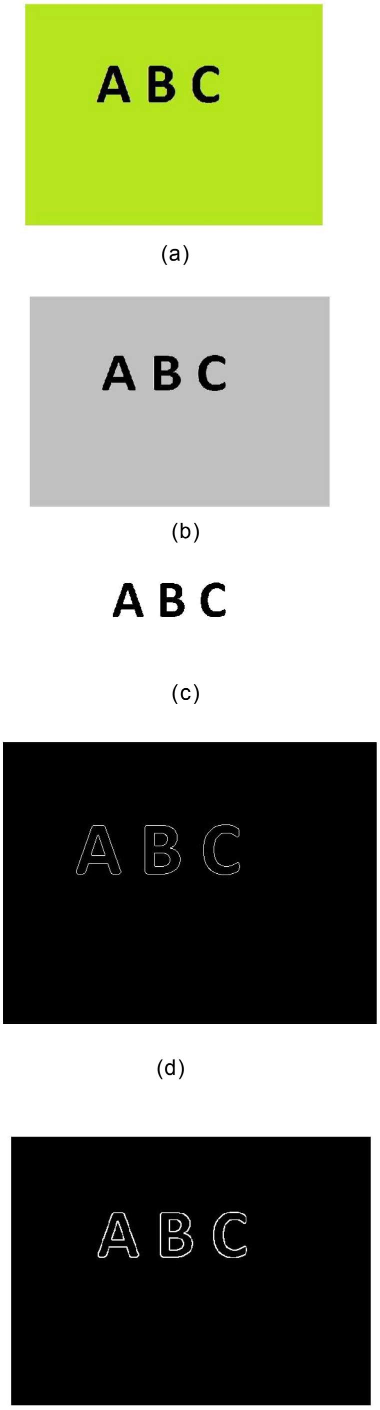
– Remove isolated specks and holes in characters, can use the Majority operator.

Feature Extraction Process

After pre-processing words can be identify by following steps used for cropping the characters

1. Read the image (Figure 2a)
2. Convert the image into gray scale (Figure 2b)

Figure 2: Different stages of Character Recongnsation (a) Original Image (b) Gray Scale (c) Binary Image (d) Edge detection (e) Dilation (f) Filling (g) Object Detection (h) Croping (i) Getting Character



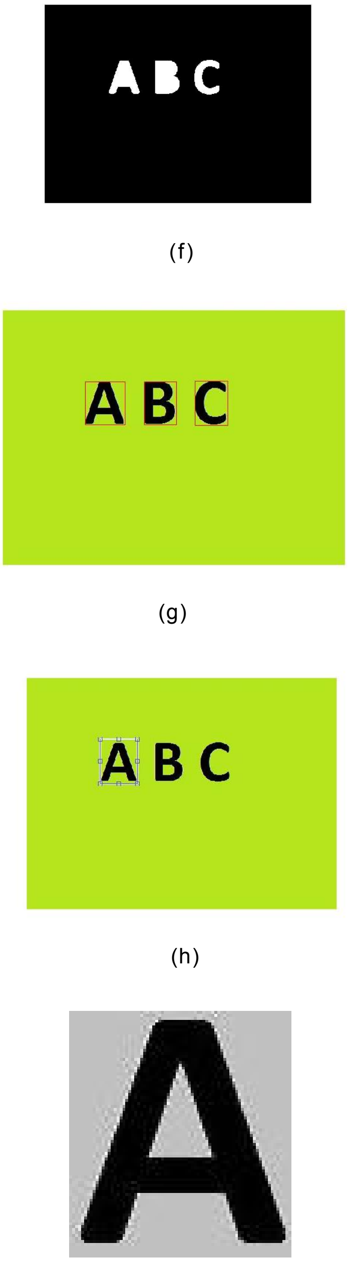
(a)

(b)

(c)

(d)

(e)



(f)

(g)

(h)

(i)

3. Convert the gray image into binary image (Figure 2c)
4. Detect the edge of the image (Figure 2d)
5. Dilate the image for finding the connected components which are disconnected by handwritten spaces (Figure 2e)

6. Fill the image after dilation (Figure 2f)
7. Analyze blobs, i.e., find all the objects present in the image and its properties (Figure 2g)
8. Plot the object (character) location (Figure 2h)
9. Crop the characters on the plotted area for data set. (Figure 2i)

With this technique images which are upload by the user can also be check but upto some text it can be identify. As different color makes binarization difficult by which more noise make word unidentified.

PROFILE RULES

Another issue we believe it is worth being considered is related to the difficulties an average OSN user may have in defining the correct threshold for the membership level. To make the user more comfortable in specifying the membership level threshold, we believe it would be useful allowing the specification of a tolerance value that, associated with each basic constraint, specifies how much the membership level can be lower than the membership threshold given in the constraint. Introducing the tolerance would help the system to handle, in some way, those messages that are very close to satisfy the rule and thus they might deserve a special treatment. In particular, these messages are those with a membership level less than the membership level threshold indicated in the rule but greater or equal to the specified tolerance value. As an example, we might have a rule requiring to block messages with violence class with a membership level greater than 0:8. As such messages with violence class with membership level of 0:79 will be published, as they are not filtered by the rule. However, introducing a tolerance value of 0:05 in the previous content-based constraint allows the system to automatically handle these messages. How the system has to behave with messages caught just for the tolerance value is a complex issue to be dealt with that may entail several different strategies. Due to its complexity and, more importantly, the need of an exhaustive experimental evaluation, in this paper we adopt a native solution according to which the system simply notifies the user about the message asking for him/her decision. We postpone the investigation of these strategies as future work.

The last component of a filtering rule is the action that the system has to perform on the messages that satisfy the rule. The possible actions we are considering are “block”, “publish” and “notify”, with the obvious semantics of blocking/publishing the message, or notify the user about the message so to wait him/her decision.

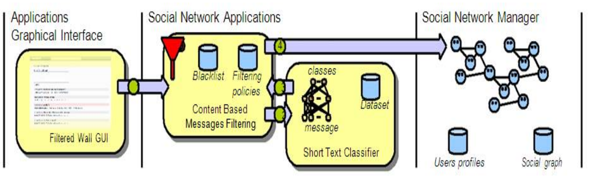
A filtering rule is therefore formally defined as follows:

Rule 1: Here user define an trust level for his friends in profile to send non neutral messages. Some range of user such as by age, some area or relation such as by state or close friend, friend, family, etc. This will decide that if user is increasing rules for his profile from unwanted messages, then check will be perform on his wall every time when friend make messages.

Example 1: The filtering rule ((Bob; friend Of; 10; 0:10); (Sex; 0:80; 0:05); block) blocks all the messages created by those users having a direct or indirect friendship relationship with Bob at maximum distance 10 and minimum trust level 0:10. In particular, it blocks only those messages with which the Sex second level class has been associated with a membership level greater than 0:80; whereas those with membership level greater than 0:75 and less than 0:80 are notified to the wall's owner.

Rule 2: A Black List rule is introduce for those user who cross some limit for the number of messages of non neutral category. It will be decide by the service provider (Figure 3).

Figure 3: Filtering Steps of User Messages



Example 2: The BL rule (Alice; (Age < 16); (0:5; my Wall; 1 week); 3 days) inserts into the BL associated with Alice's wall those young users (i.e., with age less than 16) that in the last week have a relative frequency of blocked messages greater than or equal to 0.5. Moreover, the rule specifies that these banned users have to stay in the BL for three days.

RESULTS AND DISCUSSION

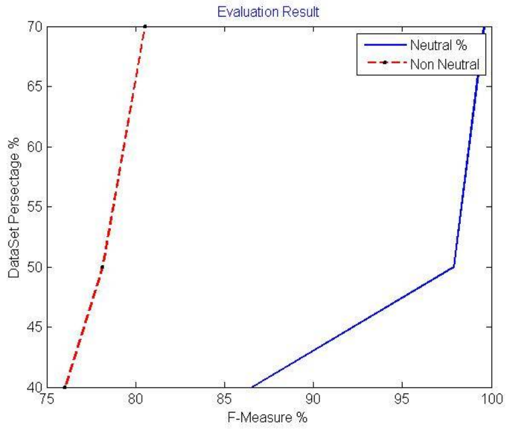
The analysis of related work has highlighted the lack of a publicly available benchmark for comparing different approaches to content-based classification of OSN short texts. To cope with this lack, we have built and made available a data set D of messages taken from Facebook.

The data set, called WmSnSec 2, which has 1166 messages from publicly accessible Italian groups have been selected and extracted by means of an automated procedure that removes undesired spam messages and, for each message, stores the message body and the name of the group from which it originates. The messages come from the group's webpage section, where any registered user can post a new message or reply to messages already posted by other users.

For distinguish messages into two category, RBFN network is use in which input is taken in form of TFIDF values that are generated from message and for clarification of Italian words one dictionary CoLFIS corpus is used, which help to find that message is from normal neutral or non neutral words.

Measures for the finding are precion, recall, F-measure. Which is the standard measure for the text mining. In that we obtain good results for the text recongnization in image as well. Results are optimize with the setting of the threshold in the RBFN for message classification. With different percentage of training dataset we obtain better result in identification.

Figure 4: Learning Percentage at Different Datasat Size of 40%, 50%, 70%



From Table 1 we find that using RBFN network classification of the messages are remarkable as in every measure more than 65% of the messages are identifiable, from the category which it belong.

Text Retreived from the image is also classify as normal text messages. Although it not need

Table 1: Measures of the Class for Different Percentage of Dataset

S. No.	Neutral %			Non Neutral %		
	Presion	Recall	F Measure	Presion	Recall	F Measure
I	68.13	97.3	81.04	100	76.31	86.56
II	64.46	99.2	78.15	98.59	97.22	97.90
III	67.41	99.3	80.53	100	99.23	99.61

any kind of separate training, as the message from image is retrieve then put it in vector for testing in the trained RBFN network. So no separate measure are require for it.

CONCLUSION

As the craze of online social networking is increase day by day, chance of getting unacceptable message also increases. So classify those messages in neutral, non neutral is done by the trained network is done in this work by RBFN neural network, classification measures are good for different level of training parameters as shown in results, this work categorize words from message as well as from the image. In future for image categorization more work required as detecting text from highly dense colored image is difficult, and time taken.

REFERENCES

1. Ali B, Villegas W and Maheswaran M (2007), “A trust based approach for protecting user data in social networks. In: Proceedings of the 2007 conference of the center for advanced studies on Collaborative research”, pp. 288-293, ACM, New York, NY, USA.

2. Amati G and Crestani F (1999), “Probabilistic learning for selective dissemination of information”, *Information Processing and Management*, Vol. 35, No. 5, pp. 633-654.

3. GIlavata J, Ewerth R and Freisleben B (2004), “Text detection in images based on unsupervised classification of high-frequency wavelet coefficients,” in *Proc. 17th Int. Conf. Pattern Recognition (ICPR’04)*, Cambridge, UK, pp. 425-428.

4. Jung K, Kim K I and Jain A K (2004), “Text information extraction in images and video: A survey,” *Pattern Recogn.*, Vol. 37, No. 5, pp. 977-997.

5. Trier O D and Jain A K (1995), “Goal-directed evaluation of binarization methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 17, No. 12, pp. 1191-1201, Dec.

6. Pratikakis, Gatos B and Ntirogiannis K (2011), “ICDAR 2011 document image binarization contest (DIBCO 2011),” in *Proc. Int. Conf. Document Anal. Recognit.*, Sep., pp. 1506-1510.

7. Chen Q, Qun, Pheng Ann H and Xia D (2008), “A double-threshold image binarization method based on edge detector,” *Pattern Recognit.*, Vol. 41, No. 4, pp. 1254-1267.

8. Liu Y X, Goto S and Ikenaga T (2006), “A contour-based robust algorithm for text detection in color images,” *IEICE Trans. Inf. Syst.*, Vol. E89-D, No. 3, pp. 1221-1230.

9. Collobert R and Weston J A (2008), Unified aRchitecture for Atural Language Processing: Deep Neural Networks With Multitask Learning, In ICML.

10. Gray and Haahr M (2004), “Personalised, Collaborative Spam FIltering”, Proceedings of 1st Conference on Email and Anti-Spam.

11. Distributed Checksum Clearinghouse, http://www.rhyolite.com/dcc/

12. Vipul's razor, http://razor.sourceforge.net/

13. Cloudmark, http://www.cloudmark.com/en/home.html

14. Spamwatch, http://www.cs.berkeley.edu/~zf/spamwatch/

15. Zhou F, Zhuang L, Zhao B Y, Huang L, Joseph A D and Kubia Towicz J (2003), "Approximate Object Location and Spam Filtering on Peer-to-peer Systems", Lecture Notes in Computer Science, pp. 1{20.

16. Ghose S, Jung J G and Jo G S (2004), "Collaborative Detection of Spam in Peer To-peer Paradigm Based on Multi-agent Systems", Lecture Notes in Computer Science, pp. 971-974.

17. Damiani E, di Vimercati S D C, Paraboschi S and Samarati P (2004), "An Open Digest-based Technique for Spam Detection", The 2004 International Workshop on Security in Parallel and Distributed Systems, Vol. 41, pp. 74-83.

18. Chiu Y F, Chen C M, Jeng B and Lin H C (2007), "An Alliance-based Antispam Approach", Natural Computation, ICNC, Vol. IV, Third International Conference on 4, 2007.

19. Damiani E, De Capitani di Vimercati S, Paraboschi S, Samarati P, di Tecnologie dell'Informazione D and Crema I (2004), "P2p-based Collaborative Spam Detection and FIltering", Proceedings of Fourth International Confer- ence on Peer-to-Peer Computing, pp. 176-183.

20. Kong J, Boykin P O, Rezaei B A, Sarshar N and Roychowdhury V P (2005), "Scalable and Reliable Collaborative Spam FIlters: Harnessing the Global Social Email Networks", Conference on Email and Anti-Spam.

21. Brin S and Page L (1998), "The Anatomy of a Large-scale Hypertextual Web Search Engine", *Computer Networks and ISDN Systems*, Vol. 30, Nos. 1-7, pp. 107-117.

22. Yerazunis W (2004), "The Spam FIltering Accuracy Plateau at 99.9 Percent Accuracy and How to Get Past it", Proceedings of the MIT Spam Conference.

23. Gavrilis D, Tsoulos I G and Dermatas E (2006), "Neural Recognition and Genetic Features Selection for Robust Detection of E-mail Spam", Lecture Notes in Computer Science, pp. 39-55, 498-501.

24. The apache spam assassin project, http://spamassassin.apache.org

IJERST

International Journal of Engineering Research and Science & Technology
Hyderabad, INDIA. Ph: +91-09441351700, 09059645577
E-mail: editorijlerst@gmail.com or editor@ijerst.com
Website: www.ijerst.com

ISSN 2319-5991
04
9 772319 599001