Volume 4, Issue 2, February 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Text Based Filtering System for OSN User Walls

A. D. Swami, B. S. Khade

Computer Department, BSIOTR (W)

Pune, India

Abstract— As we know, today everyone is using On-line Social Networks (OSNs) to communicate and share information. Therefore one important need in today On-line Social Networks (OSNs) is to give users the ability to control the messages posted on their own private space to avoid that unwanted content is displayed. OSNs provide little support to this requirement up to now. To provide this, we propose a system allowing OSN users to have a direct control on the messages posted on their walls. This is accomplished 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 which automatically produce membership labels in support of content-based filtering.

Keywords— Information Filtering, On-line Social Networks, Short Text Classification, Policy-based Personalization

I. INTRODUCTION

Today the most popular interactive medium to communicate, share and disseminate a considerable amount of human life information are On-line Social Networks (OSNs). Daily and continuous communications imply the exchange of several types of content, including free text, image, and audio and video data. According to Facebook statistics average user creates 90 pieces of content each month, whereas more than 30 billion pieces of content are shared each month. Information filtering can therefore give users the ability to automatically control the messages written on their own walls, by filtering out unwanted messages. Truly, today OSNs provide very little support to prevent unwanted messages on user walls. For example, Facebook lets users to state who is allowed to insert messages in their walls (i.e., friends, friends of friends, or defined groups of friends). However, content-based preferences are not supported. Wall messages are constituted by short text for which traditional classification methods have serious limitations since short texts do not provide sufficient word occurrences.

Therefore the aim of the present work is to propose and experimentally evaluate an automated system, called Filtered Wall (FW), able to filter unwanted messages from OSN user walls. The support for content based user preferences is the key idea of proposed system. This is possible thank to the use of a Machine Learning (ML) text categorization procedure [12] able to automatically assign with each message a set of categories based on its content. Section II reviews related work, whereas Section III presents the conceptual architecture of the proposed system. Section IV describes the ML-based text classification method used to categorize text contents, whereas Section V explains FRs and BLs. Section VI describes the case study. Finally, section VII concludes the paper.

II. LITERATURE SURVEY

Filtering is based on explanations of individual or group information preferences that typically represent long-term interests. Users get only the data that is extracted. Information filtering systems are intended to categorize a stream of dynamically generated information and present it to the user those information that are likely to satisfy user requirements [1]. In this paper the main focus was on to show the similarity between Information filtering and Information retrieval.

Foltz and dumais researched tested methods for predicting which Technical Memos (TMs) best match people's technical interests. Within Bellcore, nearly 150 new TMs are published each month, yet very few are related to any single person's interests. Feedback using previous related abstracts provided an efficient and simple way of demonstrating people's interests [2]. This was totally based on previous feedback. There was no individual based filtering. In the trial filtering system being explored at Autodesk, a user chooses a discrete rating value (e.g., terrible, boring, somewhat interesting, no comment, very interesting) for each document read. A learning algorithm associates these user ratings with document features such as author, subject, selected keywords, organizations and shared ratings from earlier readers to prioritize incoming information [3]. This paper has addressed diversified domains including newswire articles, Internet "news" articles, and broader network resources. This paper focused on just prioritizing information by using rating values. The work by Boykin and Roychowdhury [4] that offers an automated anti-spam tool that, exploiting the properties of social networks, can recognize unsolicited commercial e-mail, spam and messages related with people the user knows. However, it is important to note that the strategy just stated does not exploit ML content-based techniques.

J. Golbeck Offered an application, called FilmTrust, to personalize access to the website. But, such systems do not provide a filtering policy layer by which the user can exploit the result of the classification process to decide how and to which extent filtering out unwanted information [5]. As far as privacy is concerned, current work is mainly focusing on privacy-preserving data mining skills, that is, protecting information related to the network, i.e., relationships/nodes,

while performing social network analysis [6]. In microblogging services such as Twitter, there may arrive a situation where the users may become overwhelmed by the raw data. One solution to this problem is the classification of short text messages [7]. The proposed approach effectively classifies the text to a predefined set of generic classes such as News, Events, Opinions, Deals, and Private Messages. So, in this paper there was a focus on to classify news, opinions and other messages according to their categories.

III. FILTERED WALL CONCEPTUAL ARCHITECTURE

The conceptual architecture of OSN services is a three-tier structure (Figure 1). The first layer is Social Network Manager (SNM), commonly aims to provide the basic OSN functionalities (i.e., profile and relationship management), however the second layer provides the support for external Social Network Applications (SNAs). The supported SNAs may in turn need an additional layer for their desired Graphical User Interfaces (GUIs). By considering this reference architecture, the proposed system is placed in the second and third layers. Users interact with the system by means of a GUI to set up and manage their FRs/BLs. Furthermore, the GUI provides users with a FW, that is, a wall where only messages that are authorized according to their FRs/BLs are published. The main components of the proposed system are the Content-Based Messages Filtering (CBMF) and the Short Text Classifier (STC) modules. STC goals to classify messages according to a set of categories.

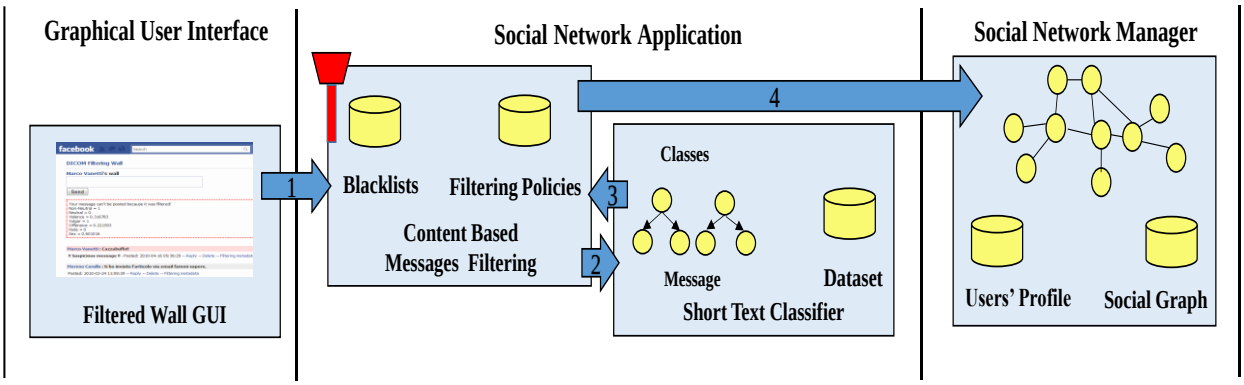

Fig 1. : Filtered Wall Conceptual Architecture

The first component exploits the message categorization provided by the STC module to enforce the FRs specified by the user. As shown in Figure 1, the path followed by a message, from its writing to the possible final publication can be given as follows:

1) The user attempts to post a message after entering the private wall of his/her contacts which is interrupted by FW.
2) A ML-based text classifier extracts metadata from the message content.
3) Metadata together with data extracted from the social graph and users' profiles provided by the classifier is used by FW, to enforce the filtering and BL rules.
4) The message will be published or filtered by FW Depending on the result of the previous step.

IV. SHORT TEXT CLASSIFIER

On datasets with large documents such as newswires corpora, established techniques used for text classification work well but suffer when the documents in the corpus are short. In this context, critical aspects are the definition 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.

From a ML point of view, we approach the task of short text categorization by defining a hierarchical two level strategy assuming that it is better to identify and eliminate "neutral" sentences, then classify "non neutral" sentences. The first level task is considered as a hard classification where short texts are labeled with crisp Neutral and Non-Neutral labels. The second level soft classifier acts on the crisp set of non-neutral short texts and, for each of them, it "simply" produces estimated appropriateness or "gradual membership" for each of the conceived classes, without taking any "hard" decision on any of them. Such a list of grades is then used by the successive phases of the filtering process.

A. Text Representation

The most appropriate feature set and feature representation for short text messages have not yet been sufficiently investigated. We consider three types of features, BoW, Document properties (Dp) and Contextual Features (CF). The first two types of features, already used in [9], are endogenous. Text representation using endogenous knowledge has a good general applicability, though in operational settings it is appropriate to use also exogenous knowledge. We introduce contextual features (CF) modelling information that characterize the environment where the user is posting. These features play important role in deterministically understanding the semantics of the messages [12].

According to Vector Space Model (VSM) for text representation, a text document d_j is represented as a vector of binary or real weights $d_j = w_{1j}, \ldots, w_{|T|j}$, where T indicates the set of terms that occur at least once in at least one

document of the collection *Tr*, and $w_{kj} \in [0; 1]$ denotes how much term t_k contributes to the semantics of document d_j. In the BoW representation, terms are identified with words. For non-binary weighting, the weight w_{kj} of term t_k in document d_j is computed according to the standard term frequency - inverse document frequency (tf-idf) weighting function, defined as

$$tf - idf(t_k, d_j) = \#(t_k, d_j).\log \frac{|T_r|}{\#T_r(t_k)} \qquad (1)$$

Where $\#(t_k, d_j)$ indicates the number of times t_k occurs in d_j, and $\#T_r(t_k)$ indicates the document frequency of term t_k, i.e., the number of documents in T_r in which t_k occurs. Dp features are heuristically calculated; their definition stems from intuitive considerations, domain specific criteria and in some cases required trial and error procedures. In more details:

a) Correct words: it represents the amount of terms $t_k \in T \cap K$, where t_k is a term of the considered document d_j and K is a set of known words for the domain language. This value is normalized by $\sum_{k=1}^{|T|} \#(t_k, d_j)$.
b) Bad words: they are determined similarly to the correct words feature, where the set K is a collection of "dirty words" for the domain language.
c) Capital words: it represents the amount of words mostly written with capital letters, calculated as the percentage of words within the message, having more than half of the characters in capital case. For example, the value of this feature for the document "To be OR Not to BE" is 0:5 since the words "OR" "Not" and "BE" are considered as capitalized ("To" is not uppercase since the number of capital characters should be strictly greater than the characters count).
d) Punctuations characters: it is computed as the percentage of the punctuation characters over the total number of characters in the message. For example, the value of the feature for the document "Hello!!! How're u doing?" is 5/24.
e) Exclamation marks: it is computed as the percentage of exclamation marks over the total number of punctuation characters in the message. Referring to the aforesaid document, the value is 3/5.
f) Question marks: it is computed as the percentage of question marks over the total number of punctuations characters in the message. Referring to the aforesaid document, the value is 1/5.

B. 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. We select the RBFN model, among the variety of multi-class ML models well-suited for text classification for the experimented competitive behavior with respect to other state of the art classifiers. The first level classifier is then structured as a regular RBFN. In the second level of the classification stage we introduce a modification of the standard use of RBFN.

We now officially describe the overall classification strategy. Let Ω be the set of classes to which each message can belong to. Each element of the supervised collected set of messages $D = \{(m_i, \vec{y}_i), \ldots, (m_{|D|}, \vec{y}_{|D|})\}$ is composed of the text m_i and the supervised label $\vec{y}_i \in \{0,1\}^{|\Omega|}$ describing the belongingness to each of the defined classes. The set D is then into two partitions, namely the training set *TrSD* and the test set *TeSD*. Let *M1* and *M2* be the first and second level classifier, respectively, and $\vec{y}_i$ be the belongingness to the Neutral class. The learning and generalization phase works as follows:

1) we extract the vector of features $\vec{x}_i$ from each message m_i. The two sets *TrSD* and *TeSD* are then converted into $TrS = \{(\vec{x}_i, \vec{y}_i), \ldots, (\vec{x} \mid TrSD|, \vec{y} \mid TrSD|)\}$ and $TeS = \{(\vec{x}_i, \vec{y}_i), \ldots, (\vec{x} \mid TeSD|, \vec{y} \mid TeSD|)\}$, respectively.
2) for *M1*, a binary training set $TrS1 = \{(\vec{x}_j, \vec{y}_j) \in TrS \mid (\vec{x}_j, y_j), y_j = \vec{y}_{j1}\}$ is generated.
3) for *M2*, a multi-class training set $TrS2 = \{(\vec{x}_j, \vec{y}_j) \in TrS \mid (\vec{x}_j, \vec{y}'_j), \vec{y}'_{jk} = \vec{y}'_{jk}+1, k=2, \ldots, |\Omega|\}$ is generated.
4) to recognize whether or not a message is non-neutral, *M1* is trained with *TrS1*. Then using the test set *TeS1* the performance of the model *M1* is calculated.
5) to compute the gradual membership to non-neutral classes, *M2* is trained with the non-neutral *TrS2* messages. Then the performance of the model *M2* is determined using the test set *TeS2*.

Therefore, the hierarchical system is composed of *M1* and *M2*, where the overall computed function f: $R^n \rightarrow R^{|\Omega|}$ is able to map the feature space to the class space, that is, to recognize the belongingness of a message to each of the $|\Omega|$ classes.

V. MANAGEMENT OF FILTERING RULES AND BLACKLIST

In this section, we introduce the rules adopted for filtering unwanted messages. We model a social network as a directed graph, where each node represents a network user and edges represent relationships between two different users. Each edge is labeled by the type of the established relationship (e.g., friend of, colleague of, parent of) and, possibly, the corresponding trust level, which represents how much a given user considers trustworthy with respect to that specific kind of relationship the user with whom he/she is establishing the relationship. We assume that trust levels are rational numbers in the range [0; 1]. There exists a direct relationship of a given type *RT* and trust value *X* between two users, if there is an edge connecting them having the labels *RT* and *X*. Moreover, two users are in an indirect relationship of a given type *RT* if there is a path of more than one edge connecting them.

A. Filtering Rules

We consider three main issues in defining the language for FRs specification. First is related to the fact that, the same message may have different meanings and relevance based on who writes it. Message 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.

Creator specification, defined as follows.

Definition 1. (Creator specification). A creator specification *creatorSpec* implicitly represents a set of OSN users. It may have one of the following forms, possibly combined:

1) a set of attribute constraints of the form an *OP av*, where an is a user profile attribute name, *av* and *OP* are, respectively, a profile attribute value and a comparison operator, compatible with *an*'s domain.
2) a set of relationship constraints of the form (*m*; *rt*; *minDepth*; *maxTrust*), representing all the OSN users participating with user m in a relationship of type *rt*, having a depth greater than or equal to *minDepth*, and a trust value less than or equal to *maxTrust*.

Example 1. The creator specification *CS1* = {*Age* < *18*; *Sex* = *female*} denotes all the females whose age is less than 18 years, whereas the creator specification *CS2* = {*Rose*; *colleague*; *2*; *0:4*} denotes all the users who are colleagues of Rose and whose trust level is less than or equal to 0.4. Finally, the creator specification *CS3* = {(*Rose*; *colleague*; *2*; *0:4*); (*Sex* = *female*)} selects only the female users from those identified by *CS2*.

For the further requirement of FRs to support the specification of content-based filtering criteria, we make use of the two-level text classification to identify messages that, with high probability, are neutral or non-neutral.; as well as, in a similar way, messages dealing with a particular second level class. And the last component of a FR is the action that the system has to accomplish on the messages that fulfill the rule. The possible actions are "*block*" and "*notify*", with the obvious semantics of blocking the message, or notifying the wall owner and wait him/her decision. A FR is therefore can be defined as follows.

Definition 2. (Filtering rule). A filtering rule FR is a tuple (*author*, *creatorSpec*, *contentSpec*, *action*), where:

1) *author* is the user who defines the rule;
2) *creatorSpec* is a creator specification, according to Definition 1;
3) *contentSpec* is a Boolean expression defined on content constraints of the form (*C*, *ml*), where *C* is a class of the first or second level and *ml* is the minimum membership level threshold required for class *C* to make the constraint satisfied;
4) *action* € {*block*; *notify*} represents the action to be performed by the system on the messages matching *contentSpec* and created by users identified by *creatorSpec*.

More than one filtering rule can apply to the same user. Therefore, a message can be published only if it is not blocked by any of the filtering rules that apply to the message creator.

B. Online Setup Assistant for FRs Thresholds

By conceiving and implementing within FW, an Online Setup Assistant (OSA) procedure, we address the problem of setting thresholds to filter rules. OSA presents the user with a set of messages selected from the dataset. For each message, the user expresses 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 permits to calculate customized thresholds representing the user attitude in accepting or rejecting certain contents.

According to the following process such messages are selected. A certain amount of non-neutral messages taken from a fraction of the dataset and not belonging to the training/test sets, are categorized by the ML in order to have, for each message, the second level class membership values. Class membership values are then quantized into a number of *qC* discrete sets and, for each discrete set, we select a number n_C of messages, obtaining sets M_C of messages with $|M_C| = n_C q_C$, where C € Ω - {*Neutral*} is a second level class. For instance, for the second level class Violence, we select 5 messages belonging to 8 degrees of *violence*, for a total of 40 messages. For each second level class *C*, messages belonging to M_C are shown. For each displayed message *m*, the user is asked to tell the decision *ma* € {*Filter*, *Pass*}. This decision indicates the willingness of the user to filter or not filter the message. Together with the decision ma the user is asked to express the degree of certainty *mb* € {0, 1, 2, 3, 4, 5} with which the decision is taken, where *mb* = 5 represents the highest certainty, whereas *mb* = 0 represents the lowest certainty.

Example 2. Suppose that Alice is an OSN user and she wants to always block messages having a high degree of violence content. Through the session with OSA, the threshold representing the user attitude for the violence class is set to 0:7. Now suppose that Alice wants to filter only messages coming from indirect friends, whereas for direct friends such messages should be blocked only for those users whose trust value is below 0.4.

.

C. Blacklists

BLs are directly managed by the system, and should be able to determine the users to be inserted in the BL and decide user's retention in the BL is finished. Such information are given to the system through a set of rules, called BL rules. We let 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, by, at the same time, being able to post in other walls.

Similar to FRs, our BL rules make the wall owner able to identify users to be blocked based on their profiles as well as their relationships in the OSN. Therefore, by means of a BL rule, wall owners can be able to ban from their walls users they do not directly know BL for another while, as his/her behavior is not improved. This principle works for those users that have been already inserted in the considered BL at least one time.

Definition 3. (BL rule). A BL rule is a tuple (*author*, *creatorSpec*, *creatorBehavior*, *T*), where:

1) 1)*author* is the OSN user who states the rule, i.e., the wall owner;
2) *creatorSpec* is a creator specification, specified according to Definition 1;
3) *creatorBehavior* consists of two components *RFBlocked* and *minBanned*. *RFBlocked* = (*RF*, *mode*, *window*) is defined such that:
   - $RF = \frac{\#bMessages}{\#tMessages}$, where #tMessages is the total number of messages that each OSN user identified by creatorSpec has tried to publish in the author wall (*mode* = *myWall*) or in all the OSN walls (*mode* = *SN*); whereas *#bMessages* is the number of messages among those in *#tMessages* that have been blocked;
   - *window* is the time interval of creation of those messages that have to be considered for RF computation; *minBanned* = (*min*, *mode*, *window*), where min is the minimum number of times in the time interval specified in window that OSN users identified by *creatorSpec* have to be inserted into the BL due to BL rules specified by author wall (*mode* = *myWall*) or all OSN users (*mode* = *SN*) in order to satisfy the constraint.
4) *T* indicates the time period the users identified by *creatorSpec* and *creatorBehavior* have to be banned from author wall.

Example 3. The BL rule:
(*Bob*; (*Age* < *18*); (*0*:*5*; *myWall*; *1 week*); *2 days*) inserts into the BL associated with Bob's wall those young users (i.e., with age less than 18) that in the last week have a relative frequency of blocked messages on *Bob*'s wall greater than or equal to 0:5. Furthermore, the rule states that these banned users have to stay in the BL for two days. If *Bob* adds the following component (*2*,*SN*, *1 week*) to the BL rule, he enlarges the set of banned users by inserting also the users that in the last week have been inserted at least two times into any OSN BL.

VI. DICOMFW

It is a prototype Facebook application8 that imitates a personal wall where the user can apply a simple combination of the proposed FRs. our focus is only on the FRs, leaving BL implementation as a future development. However, the implemented functionality is critical, since it permits the STC and CBMF components to interact. Since this application is considered as a wall and not as a group, the contextual information (from which CF are extracted) linked to the name of the group are not directly accessible.

To summarize, our application allows to:

1) view the list of users' FWs;
2) view messages and post a new one on a FW;
3) define FRs using the OSA tool.

VII. CONCLUSION

In this paper, we have presented a system to filter unwanted messages from OSN walls. The system exploits a ML soft classifier to enforce customizable content-dependent FRs. Furthermore, the flexibility of the system in terms of filtering options is enhanced through the management of BLs.

The first concerns the extraction and/or selection of contextual features that have been shown to have a high discriminative power. The second task includes the learning phase. As the underlying domain is dynamically changing, the collection of pre-classified data may not be representative in the longer term.

The present batch learning strategy, based on the preliminary collection of the entire set of labeled data from experts, permitted an accurate experimental evaluation but needs to be developed to include new operational requirements. We plan to address this problem by investigating the use of on-line learning paradigms able to include label feedbacks from users in future work. The proposed system may suffer of problems similar to those encountered in the specification of OSN privacy settings. We plan to investigate the development of a GUI and a set of related tools to make easier BL and FR specification, as usability is a key requirement for such kind of applications.

REFERENCES

[1] N. J. Belkin and W. B. Croft, "Information filtering and information retrieval: Two sides of the same coin?" *Communications of the ACM*, vol. 35, no.12, pp. 29–38, 1992.
[2] P. W. Foltz and S. T. Dumais, "Personalized information delivery: An analysis of information filtering methods," *Communications of the ACM*, vol. 35, no. 12, pp. 51–60, 1992.
[3] P. E. Baclace, "Competitive agents for information filtering," *Communications of the ACM*, vol. 35, no. 12, p. 50, 1992.

[4] Boykin, P.O., Roychowdhury, V.P.: Leveraging social networks to fight spam. IEEE Computer Magazine 38, 61–67 (2005).

[5] J. Golbeck, "Combining provenance with trust in social networks for semantic web content filtering*," in Provenance and Annotation Data, ser. Lecture Notes in Computer Science*, L. Moreau and I. Foster, Eds. 2006

[6] Carminati, B., Ferrari, E.: Access control and privacy in web-based social networks. International Journal of Web Information Systems 4, 395–415 (2008)

[7] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in twitter to improve information filtering," 2010P. J. Denning, "Electronic junk," *Communications of the ACM*, vol. 25, no. 3, pp. 163–165, 1982.

[8] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," Journal of Machine Learning Research, 2004.

[9] M. Vanetti, E. Binaghi, B. Carminati, M. Carullo, and E. Ferrari, "Content-based filtering in on-line social networks," *in Proceedings of ECML/PKDD Workshop on Privacy and Security issues in Data Mining and Machine Learning (PSDML 2010), 2010.*

[10] S. Pollock, "A rule-based message filtering system," *ACM Transactions on Office Information Systems*, vol. 6, no. 3, pp. 232–254, 1988.

[11] Strater, K., Richter, H.: Examining privacy and disclosure in a social networking community. In: SOUPS '07: *Proceedings of the 3rd symposium on Usable privacy and security.* pp. 157– 158. ACM, New York, NY, USA (2007)

[12] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

