Content-Based Filtering in On-line Social Networks

M. Vanetti, E. Binaghi, B. Carminati, M. Carullo and E. Ferrari

Department of Computer Science and Communication
University of Insubria
21100 Varese, Italy
{marco.vanetti, elisabetta.binaghi, barbara.carminati,
moreno.carullo, elena.ferrari} @uninsubria.it

Keywords: On-line Social Network, Content-Based Filtering, Message Filtering, Short Text Classification, Filtered Wall, Filtering Policies

Abstract. This paper proposes a system enforcing content-based message filtering 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 labelling messages in support of content-based filtering.

1 Introduction

In the last years, On-line Social Networks (*OSNs*) have become a popular interactive medium to communicate, share and disseminate 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. The huge and dynamic character of these data creates the premise for the employment of web content mining strategies aimed to automatically discover useful information dormant within the data and then provide an active support in complex and sophisticated tasks involved in social networking analysis and management. 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*.

The aim of the present work is to propose and experimentally evaluate an automated system, called *Filtered Wall* (FW), able to filter out unwanted messages from social network user walls. The key idea of the proposed system is the support for content-based user preferences. This is possible thank to the use of a Machine Learning (ML) text categorization procedure [21] able to automatically assign with each message a set of categories based on its content. We believe that the proposed strategy is a key service for social networks in that in today social networks users have little control on the messages displayed on their walls. For example, Facebook allows users to state who is allowed to insert messages in their walls (i.e., friends, friends of friends, or defined groups of friends). However, no content-based preferences are supported. For instance, it is not possible to prevent political or vulgar messages. In contrast, by means of the proposed mechanism, a user can specify what contents should not be displayed on his/her wall, by specifying a set of *filtering rules*. Filtering rules are very flexible in terms of the filtering requirements they can support, in that they allow to specify filtering conditions based on user profiles, user relationships as well as the output of the ML categorization process. In addition, the system provides the support for user-defined blacklists, that is, list of users that are temporarily prevented to post messages on a user wall.

The remainder of this paper is organized as follows: in Sect. 2 we describe work closely related to this paper, Sect. 3 introduces the conceptual architecture of the proposed system. Sect. 4 describes the ML-based text classification method used to categorize text contents, whereas Sect. 5 provides details on the content-based filtering system. Sect. 6 describes and evaluates the overall proposed system with a case study prototype application. Finally, Sect. 7 concludes the paper.

2 Related Work

In the OSN domain, interest in access control and privacy protection is quite recent. As far as privacy is concerned, current work is mainly focusing on privacy-preserving data mining techniques, that is, protecting information related to the network, i.e., relationships/nodes, while performing social network analysis [4]. Work more related to our proposals are those in the field of access control. In this field, many different access control models and related mechanisms have been proposed so far (e.g., [5, 23, 1, 9]), which mainly differ on the expressivity of the access control policy language and on the way access control is enforced (e.g., centralized vs. decentralized). Most of these models express access control requirements in terms of relationships that the requestor should have with the resource owner. We use a similar idea to identify the users to which a filtering rule applies. However, the overall goal of our proposal is completely different, since we mainly deal with filtering of unwanted contents rather than with access control. As such, one of the key ingredients of our system is the availability of a description for the message contents to be exploited by the filtering mechanism as well as by the language to express filtering rules. In contrast, no one of the access control models previously cited exploits the content of the resources to enforce access control. We believe that this is a fundamental difference. Moreover, the notion of blacklists and their management are not considered by any of these access control models.

Content-based filtering has been widely investigated by exploiting ML techniques [2, 13, 19] as well as other strategies [12, 7]. However, 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 [3] 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.

The advantages of using ML filtering strategies over ad-hoc knowledge engineering approaches are a very good effectiveness, flexibility to changes in the domain and portability in different applications. However difficulties arise in finding an appropriate set of features by which to represent short, grammatically ill formed sentences and in providing a consistent training set of manually classified texts.

3 Filtered Wall Conceptual Architecture

The aim of this paper is to develop a method that allows OSN users to easily filter undesired messages, according to content based criteria. In particular, we are interested in defining an automated language-independent system providing a flexible and customizable way to filter and then control incoming messages.

Before illustrating the architecture of the proposed system, we briefly introduce the basic model underlying OSNs. In general, the standard way to model a social network is as directed graph, where each node corresponds to a network user and edges denote relationships between two different users. In particular 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 it. Therefore, 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, such that all the edges in the path have label RT [11].

In general, the architecture in support of OSN services is a three-tier structure. The first layer commonly aims to provide the basic OSN functionalities (i.e., profile and relationship management). Additionally, some OSNs provide an additional layer allowing the support of external Social Network Applications (SNA).[1] Finally, the supported SNA may require an additional layer for their needed graphical user interfaces (GUIs). According to this reference layered architecture, the proposed system has to be placed in the second and third layers (Fig. 1), as it can be considered as a SNA. In particular, users interact with the system by means of a GUI setting up their filtering rules, according to which messages have to be filtered out (see Sect. 5 for more details). Moreover, the GUI provides users with a FW, that is, a wall where only messages that are authorized according to their filtering rules are published.

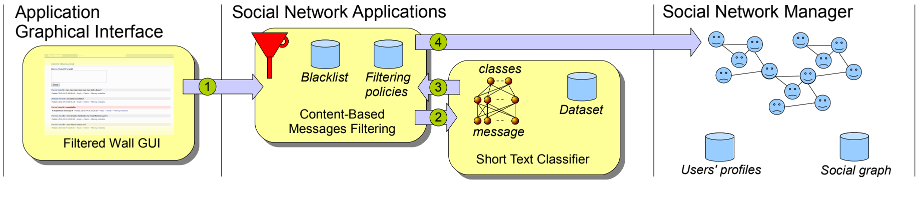


Fig. 1. Filtered Wall Conceptual Architecture

[1] See for example the Facebook Developers documentation, available on-line at http://developers.facebook.com/docs/

The core components of the proposed system are the *Content-Based Messages Filtering* (CBMF) and the *Short Text Classifier* (STC) modules. The latter component aims to classify messages according to a set of categories. The strategy underlying this module is described in Sect. 4. In contrast, the first component exploits the message categorization provided by the STC module to enforce the filtering rules specified by the user. Note that, in order to improve the filtering actions, the system makes use of a *blacklist* (BL) mechanism. By exploiting BLs, the system can prevent messages from undesired users. More precisely, as discussed in Sect. 5, the system is able to detect who are the users to be inserted in the BL according to the specified user preferences, so to block all their messages and for how long they should be kept in the BL.

4 Short Text Classifier

Established techniques used for text classifications work well on datasets with large documents such as newswires corpora [16] 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.

The task of semantically categorizing short texts is conceived in our approach as a multi-class soft classification process composed of two main phases: text representation and ML-based classification.

4.1 Text Representation

The extraction of an appropriate set of features by which representing the text of a given document is a crucial task strongly affecting the performance of the overall classification strategy. Different sets of features for text categorization have been proposed in the literature [21], however the most appropriate feature types and feature representation for short text messages have not been sufficiently investigated. Proceeding from these considerations and basing on our experience documented in previous work [6], we consider two types of features, *Bag of Words* (BoW) and *Document properties* (Dp), that are used in the experimental evaluation to determine the combination that is most appropriate for short message classification (see Sect. 6).

The underlying model for text representation is the Vector Space Model [17] for which a text document d_j is represented as a vector of binary or real weights $d_j = w_{1j}, \ldots, w_{|\mathcal{T}|j}$, where $\mathcal{T}$ is the set of terms (sometimes also called features) that occur at least once in at least one document of the collection of document $\mathcal{T}r$, and $w_{kj} \in [0;1]$ represents how much term t_k contributes to the semantics of document d_j. In the BoW representation, terms are identified with words. In the case of 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 [20], defined as

$$tf - idf(t_k, d_j) = \#(t_k, d_j) \cdot \log \frac{|\mathcal{T}_r|}{\#\mathcal{T}_r(t_k)} \tag{1}$$

where $\#(t_k, d_j)$ denotes the number of times t_k occurs in d_j, and $\#\mathcal{T}_r(t_k)$ denotes the document frequency of term t_k, i.e., the number of documents in $\mathcal{T}r$ in which t_k

occurs. Domain specific criteria are adopted in choosing an additional set of features concerning orthography, known words and statistical properties of messages. In more details:

- *Correct words:* express the amount of terms $t_k \in \mathcal{T} \cap \mathcal{K}$, where t_k is a term of the considered document d_j and $\mathcal{K}$ is a set of known words for the domain language. This value is normalized by $\sum_{k=1}^{|\mathcal{T}|} \#(t_k, d_j)$.
- *Bad words:* are computed similarly to the *Correct words* feature, whereas the set $\mathcal{K}$ is a collection of "dirty words" for the domain language.
- *Capital words:* express 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 the 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).
- *Punctuations characters:* calculated 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 $5/24$.
- *Exclamation marks:* calculated as the percentage of exclamation marks over the total number of punctuation characters in the message. Referring to the aforementioned document the feature value is $3/5$.
- *Question marks:* calculated as the percentage of question marks over the total number of punctuations characters in the message. Referring to the aforementioned document the feature value is $1/5$.

4.2 Machine Learning-based Classification

We address the short text categorization as a hierarchical two-level classification process. The first-level classifier performs a binary hard categorization that labels messages as *Neutral* and *Non-Neutral*. The first-level filtering task facilitates the subsequent second-level task in which a finer-grained classification is performed. The second-level classifier performs a soft-partition of *Non-neutral* messages assigning with a given message a gradual membership to each of the non neutral classes. Among the variety of multi-class ML models well-suited for the text classification, we choose the *Radial Basis Function Network* (RBFN) model [18] for its proven robustness in dealing with inherent vagueness in class assignments and for the experimented competitive behavior with respect to other state-of the-art classifiers. The first and second-level classifiers are then structured as regular RBFNs, conceived as hard and soft classifier respectively. Its non-linear function maps the feature space to the categories space as a result of the learning phase on the given training set constituted by manually classified messages. As will be described in Sect. 6, our strategy includes the availability of a team of experts, previously tuned on the way with which to intend the interpretation of messages and their categorization, provide manually classified examples.

We now formally 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 $D = \{(m_i, \boldsymbol{y}_i), \ldots, (m_{|D|}, \boldsymbol{y}_{|D|})\}$ is composed of the text m_i and the supervised label $\boldsymbol{y}_i \in \{0,1\}^{|\Omega|}$ describing the belongingness to each of the defined classes. The set D is then split into two partitions, namely the training set TrS_D and the test set TeS_D.

Let M_1 and M_2 be the first and second level classifier respectively and $\boldsymbol{y}_1$ be the belongingness to the *Neutral* class. The learning and generalization phase works as follows:

1. each message m_i is processed such that the vector $\boldsymbol{x}_i$ of features is extracted. The two sets TrS_D and TeS_D are then transformed into $TrS = \{(\boldsymbol{x}_i, \boldsymbol{y}_i), \ldots, (\boldsymbol{x}_{|TrS_D|}, \boldsymbol{y}_{|TrS_D|})\}$ and $TeS = \{(\boldsymbol{x}_i, \boldsymbol{y}_i), \ldots, (\boldsymbol{x}_{|TeS_D|}, \boldsymbol{y}_{|TeS_D|})\}$ respectively.
2. a binary training set $TrS_1 = \{(\boldsymbol{x}_j, \boldsymbol{y}_j) \in TrS \mid (\boldsymbol{x}_j, y_j), y_j = \boldsymbol{y}_{j_1}\}$ is created for M_1.
3. a multi-class training set $TrS_2 = \{(\boldsymbol{x}_j, \boldsymbol{y}_j) \in TrS \mid (\boldsymbol{x}_j, \boldsymbol{y}'_j), \boldsymbol{y}'_{j_k} = \boldsymbol{y}_{j_{k+1}},\ k = 2, \ldots, |\Omega|\}$ is created for M_2.
4. M_1 is trained with TrS_1 with the aim to recognize whether or not a message is *Non-Neutral*. The performance of the model M_1 is then evaluated using the test set TeS_1.
5. M_2 is trained with the *Non-Neutral* TrS_2 messages with the aim of computing gradual membership to the *Non-Neutral* classes. The performance of the model M_2 is then evaluated using the test set TeS_2.

To summarize the hierarchical system is then composed of M_1 and M_2, 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. The membership values for each class of a given message computed by f are then exploited by the CBMF module described in the following section.

5 Content-Based Filtering with Blacklist

In this section, we introduce the rules adopted for filtering unwanted messages. In defining the language for filtering rules specification, we consider three main issues that, in our opinion, should affect the filtering decision. The first aspect is related to the fact that, in OSNs like in everyday life, the same message may have different meanings and relevances based on who writes it. As a consequence, filtering rules should allow users to state *constraints on message creators*. Thus, creators on which a filtering rule applies should be selected on the basis of several different criteria, one of the most relevant is by imposing conditions on user profile's attributes. In such a way it is, for instance, possible to define rules applying only to young creators, to creators with a given religious/political view, or to creators that we believe are not expert in a given field (e.g., by posing constraints on the work attribute of user profile).

Given the social network scenario, we see a further way according to which creators may be identified, that is, by exploiting information on their social graph. This implies to state conditions on type, depth and trust values of the relationship(s) creators should be involved in order to apply them the specified rules.

Another relevant issue to be taken into account in defining a language for filtering rules specification is the support for *content-based rules*. This means filtering rules identifying messages according to constraints on their contents. In order to specify and enforce these constraints, we make use of the two-level text classification introduced in Sect. 4. More precisely, the idea is to exploit classes of the first and second level as well as their corresponding membership levels to make users able to state content-based constraints. For example, it would be possible to identify messages that, with high probability, are neutral or non-neutral, (i.e., messages with which the *Neutral/Non-Neutral* first level class is associated with membership level greater than a given threshold); as well as, in a similar way, messages dealing with a particular second level class.

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 naïve 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.

Definition 1. *A filtering rule fr is a tuple ($creatorSpec$, $contentSpec$, $action$), where:*

- *$creatorSpec$ denotes the set of OSN users to which the rule applies. It can have one of the following forms, possibly combined:* (1) *a set of attribute constraints of the form an OP av, where an is a profile attribute name, av is a profile attribute value, whereas OP is a comparison operator compatible with an's domain;* (2) *a set of relationship constraints of the form $(m,\ rt,\ maxDepth,\ minTrust)$, denoting all the OSN users participating with user m in a relationship of type rt, having a depth less or equal to $maxDepth$, and a trust value greater than or equal to $minTrust$.*
- *$contentSpec$ is a Boolean expression defined on content constraints. In particular, each content constraint is defined as a triple $(C,\ ml,\ T)$, where C is a class of the first or second level, ml is the minimum membership level required to class C to make the constraint satisfied, and T is the tolerance for the constraint.*

- *$action \in \{block,\ publish,\ notify\}$ denotes the action to be performed by the system on the messages matching $contentSpec$ and created by users identified by $creatorSpec$.*

Example 1. The filtering rule $((Bob,\ friendOf,\ 10,\ 0.10),(Sex,\ 0.80,\ 0.05),\ block)$ 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.

As introduced in Sect. 3, we make use of a BL mechanism to avoid messages from undesired creators. BL is managed directly by the system, which according to our strategy is able to: *(1)* detect who are the users to be inserted in the BL, *(2)* block all their messages, and *(3)* decide when users retention in the BL is finished. To make the system able to automatically perform these tasks, the BL mechanism has to be instructed with some rules, hereafter BL rules. In particular, these rules aim to specify *(a)* how the BL mechanism has to identify users to be banned and *(b)* for how long the banned users have to be retained in the BL, i.e., the retention time. Before going into the details of BL rules specification, it is important to note that according to our system design, these rules are not defined by the Social Network manager, which implies that these rules are not meant as general high level directives to be applied to the whole community. Rather, we decide to let the users themselves, i.e., the wall's owners to specify BL rules regulating who has to be banned from their walls. As such, the wall owner is able to clearly state how the system has to detect users to be banned and for how long the banned users have to be retained in the BL. Note that, according to this strategy, a user might be banned from a wall, by, at the same time, being able to post in other walls.

In defining the language of BL rule specification we have mainly considered the issue of how to identify users to be banned. We are aware that several strategies would be possible, which might deserve to be considered in our scenario. Among these, in this paper we have considered two main directions, postponing as future work a more exhaustive analysis of other possible strategies. In particular, our BL rules make the wall owner able to identify users to be blocked according to their profiles as well as their relationships. By means of this specification, wall owners are able to ban from their walls, for example, users they do not know directly (i.e., with which they have only indirect relationships), or users that are friend of a given person as they may have a bad opinion of this person. This banning can be adopted for an undetermined time period or for a specific time window. Moreover, banning criteria take in consideration also users' behavior in the OSN. More precisely, among possible information denoting users' bad behavior we have focused on two main measures. The first is related to the principle that if within a given time interval a user has been inserted into the BL for several times, say greater than a given threshold, he/she might deserve to stay in the BL for another while, as his/her behavior is not improved. This principle works for those users that have been already inserted in the BL at least one time. To catch new bad behaviors, we use the *Relative Frequency* (RF), defined later in this section. RF let the system be able to detect those users whose messages continue to fail the filtering rules. A BL rule is therefore formally defined as follows.

Definition 2. *A BL rule is a tuple* $(author,\ creatorSpec,\ creatorBehavior,\ T)$*, where:*

- $author$ *is the OSN user who specifies the rule, i.e., the wall owner;*
- $creatorSpec$ *denotes the set of OSN users to which the rule applies. It can have one of the following forms, possibly combined:* (1) *a set of attribute constraints of the form* $an\ OP\ av$*, where* an *is a profile attribute name,* av *is a profile attribute value, whereas* OP *is a comparison operator compatible with* an*'s domain;* (2) *a set of relationship constraints of the form* $(m,\ rt,\ maxDepth,\ minTrust)$*, denoting all the OSN users participating with user* m *in a relationship of type* rt*, having a depth less or equal to* $maxDepth$*, and a trust value greater or equal to* $minTrust$*.*
- $creatorBehavior$ = $RFBlocked\ \vee\ minBanned$*. In particular,* $RFBlocked$ = $(RF,\ mode,\ window)$ *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.*
  - $mode\ \in\ \{myWall,\ SN\}$ *specifies if the messages to be considered for the RF computation have to be gathered from the* $author$*'s wall only (*$mode\ =\ myWall$*) or from the whole community walls (*$mode = SN$*).*
  - $window$ *is the time interval of creation of those messages that have to be considered for* RF *computation;*

  $minBanned$ = $(min,\ mode,\ window)$ *is defined such that* 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 (*$mode\ =\ me$*) or other OSN users (*$mode\ =\ SN$*) in order to satisfy the constraint.*
- T *denotes the time period the users identified by* $creatorSpec$ *or* $creatorBehavior$ *have to be banned from* $author$ *wall.*

Example 2. The BL rule $(Alice,\ (Age\ <\ 16),\ (0.5,\ myWall,\ 1\ week),\ 3\ days)$ 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.

6 A Case Study: DicomFW

In this section we illustrate how our system can be applied in a real OSN, that is, Facebook. In the following we describe the prototype implementation details, we then provide some preliminary experiments in order to evaluate the performance of our system.

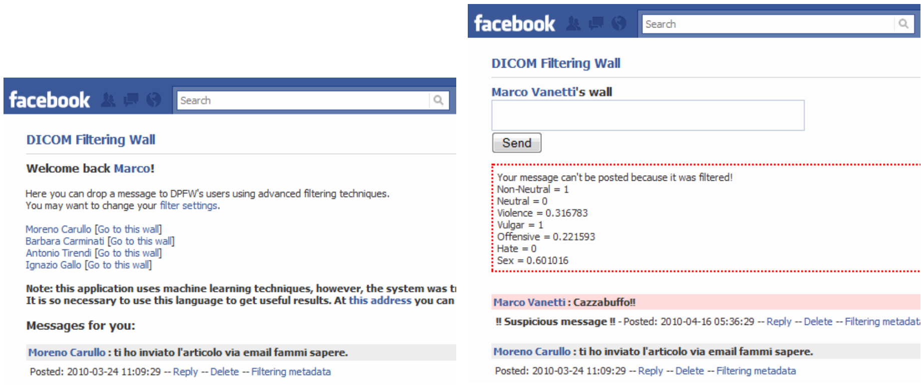
6.1 Problem and Dataset Description

We have built a dataset[2] D of messages taken from Facebook. We have selected an heterogeneous set of publicly visible user groups in italian language. The set of classes

[2] http://www.dicom.uninsubria.it/~marco.vanetti/wmsnsec/

$\Omega = \{Neutral, Violence, Vulgar, Offensive, Hate, Sex\}$ is considered, where $\Omega - \{Neutral\}$ belongs to the second level classes. The set D has 1266 elements, where the percentage of elements in D that belongs to the *Neutral* class is 31%. In order to deal with intrinsic ambiguity in assigning messages to classes, we conceive that a given message belongs to more than one classes. In particular, on the average, a message belongs to two classes ($Vulgar$ and $Offensive$ are the most related classes). Each message has been labeled by a group of five experts and the class membership values $\boldsymbol{y}_j \in \{0,1\}^{|\Omega|}$ for a given message m_j were computed by majority voting. Within *Non-Neutral* classes, the resulting final distribution of the sub-classes is uniform.

6.2 Demo Application



DICOM Filtering Wall

Welcome back Marco!

Here you can drop a message to DPFW's users using advanced filtering techniques.
You may want to change your filter settings.

Moreno Carullo [Go to this wall]
Barbara Carminati [Go to this wall]
Antonio Tirendi [Go to this wall]
Ignazio Gallo [Go to this wall]

Note: this application uses machine learning techniques, however, the system was t
It is so necessary to use this language to get useful results. At this address you can

Messages for you:

Moreno Carullo : ti ho inviato l'articolo via email fammi sapere.
Posted: 2010-03-24 11:09:29 -- Reply -- Delete -- Filtering metadata

(a)

DICOM Filtering Wall

Marco Vanetti's wall

Your message can't be posted because it was filtered!
Non-Neutral = 1
Neutral = 0
Violence = 0.316783
Vulgar = 1
Offensive = 0.221593
Hate = 0
Sex = 0.601016

Marco Vanetti : Cazzabuffo!!
!! Suspicious message !! - Posted: 2010-04-16 05:36:29 -- Reply -- Delete -- Filtering metadat

Moreno Carullo : ti ho inviato l'articolo via email fammi sapere.
Posted: 2010-03-24 11:09:29 -- Reply -- Delete -- Filtering metadata

(b)

Fig. 2. Two relevant use cases of the DicomFW application: (a) start page proposes the list of walls the OSN user can see, (b) a message filtered by the wall's owner filtering rules

Throughout the development of the prototype[3] we have focused our attention on filtering rules, leaving BL implementation as a future improvement. The filtering rules functionality is critical since permits the STC and CBMF components to interact.

To summarize, our application (Fig. 2) permits to: (1) view the list of users' FWs (see Fig. 2(a)), (2) view messages on a FW, (3) post a message on other FWs, (4) define filtering rules for the FWs. When a user tries to post a message on a FW, if it is blocked by a filtering rule, he/she receives an alerting message (see Fig. 2(b)).

6.3 Short Text Classifier Evaluation

Evaluation Metrics Two different types of measures will be used to evaluate the effectiveness of first level and second level classifications. In the first level, the short text

[3] `http://apps.facebook.com/dicompostfw/`

classification procedure is evaluated on the basis of the contingency table approach. In particular the derived well known Overall Accuracy (OA) index capturing the simple percent agreement between truth and classification results, is complemented with the Cohen's KAPPA (K) coefficient thought to be a more robust measure that takes into account the agreement occurring by chance [14]

At second level, we adopt measures widely accepted in the Information Retrieval and Document Analysis field, that is, Precision (P), that permits to evaluate the number of false positives, Recall (R), that permits to evaluate the number of false negatives, and the overall metric F-Measure (F_β), defined as the harmonic mean between the above two indexes [10]. Precision and Recall are computed by first calculating P and R for each class and then taking the average of these, according to the macro-averaging method [21], in order to compensate unbalanced class cardinalities. The F-Measure is commonly defined in terms of a coefficient β that defines how much to favor Recall over Precision. We chose to set $\beta = 1$.

Numerical Results By trial and error we have found a quite good parameters configuration for the RBFN learning model. The best value for the M parameter, that determines the number of Basis Function, seems to be $N/2$, where N is the number of input patterns from the dataset. The value used for the spread σ, which usually depends on the data, is $\sigma = 32$ for both networks M_1 and M_2. As mentioned in Sect. 4.1, the text has been represented with the BoW feature model together with a set of additional features Dp based on document local properties. To calculate the first two features we used two specific italian word-lists, one of these is the CoLFIS corpus [15]. The cardinalities of TrS_D and TeS_D, subsets of D with $TrS_D \cap TeS_D = \emptyset$, were chosen so that TrS_D is twice larger than TeS_D. Table 1 exposes the main results varying used features and term weighting for BoW.

Table 1. Results for the two stages of the proposed hierarchical classifier

Configuration		First level		Second Level		
Features	BoW TW	OA	K	P	R	F_1
BoW	binary	72.9%	28.8%	69%	36%	48%
BoW	tf-idf	73.8%	30.0%	75%	38%	50%
BoW+Dp	binary	73.8%	30.0%	73%	38%	50%
BoW+Dp	tf-idf	75.7%	35.0%	74%	37%	49%
Dp	-	69.9%	21.6%	37%	29%	33%

Network M_1 has been evaluated using the OA and the K value. Precision, Recall and F-Measure were used for the M_2 network because, in this particular case, each pattern can be assigned to one or more classes.

Table 1 shows how different features configuration and term weighting (for the BoW features) impact on the results. The numbers prove that, for the first classification stage, Dp features are important in order to distinguish neutral messages from others. BoW features better support the classification task if used with the term weighting as seen in Table 1. The last consideration that we can do on the results is that the network M_2 works better using only the BoW features. This happens because Dp features are too general in order to contribute significantly in the second stage classification, where there are more than two classes, all of non-neutral type, and it is required a greater effort in order to understand the semantics of the message.

Table 2 exposes detailed results for the best classifier (BoW+Dp with tf-idf term weighting for the first stage and BoW with tf-idf term weighting for the second stage). Precision, Recall and F-Measure values, related to each class, show that the most problematic cases are the $Hate$ and $Offensive$ classes. Messages with hate and offensive contents often hold quite complex concepts that hardly may be understood using a term based approach. The behavior of the system on the *Non-Neutral* classes is to be interpreted in light of the intrinsic difficulty of short message semantics.

Table 2. Results of the proposed model in term of Precision, Recall and F-Measure values for each class

	First level		Second Level				
Metric	Neutral	Non-Neutral	Violence	Vulgar	Offensive	Hate	Sex
P	77%	69%	92%	69%	86%	58%	75%
R	92%	38%	32%	53%	27%	26%	52%
F_1	84%	49%	47%	60%	41%	36%	62%

6.4 Overall Performance and Discussion

In order to provide an overall assessment of how effectively the system will apply a filtering rule, we look again at Table 2. This table allows us to estimate the Precision and Recall of our filtering rules, since values reported in Table 2 have been computed for filtering rules with content specification component set to $(C, 0.5, 0.0)$, where $C \in \{Neutral, Non-Neutral, Violence, Vulgar, Offensive, Hate, Sex\}$. Let us suppose that the system applies a given rule on a certain message. As such, Precision reported in Table 2 is the probability that the decision taken on the considered message (that is blocking it or not) is actually the correct one. In contrast, Recall has to be interpreted as the probability that, given a rule that must be applied over a certain message, the rule is finally enforced. Let us now discuss, with some examples, the results presented in Table 2, which reports Precision and Recall values. The second column of Table 2 represents the Precision and the Recall value computed for the filtering rule with $(Neutral, 0.5, 0.0)$ content constraint. In contrast, the fifth column stores the Precision and the Recall value computed for the filtering rule with $(Vulgar, 0.5, 0.0)$ constraint.

Results obtained for the content-based specification component, on the first level classification, can be considered good enough and aligned with those obtained by well-known information filtering techniques [12]. Results obtained for the content-based specification component on the second level must be interpreted in view of the intrinsic difficulties in assigning to a messages a semantically most specific category (see the discussion in Sect. 6.3). As such we are optimistic that after having improved the text classifier strategies such to overcome these difficulties, results on second level will be aligned with those on the first level. More precisely, improvements we are planning and carrying on focus on reducing the inconsistency in the collection of manually classified examples and improving the message representation with the inclusion of contextual information.

7 Conclusions

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

This work is the first step of a wider project. The early encouraging results we have obtained on the classification procedure prompt us to continue with other work that will aim to improve the quality of classification. Additionally, we plan to enhance our filtering rule system, with a more sophisticated approach to manage those messages caught just for the tolerance and to decide when a user should be inserted into a BL. For instance, the system can automatically take a decision about the messages blocked because of the tolerance, on the basis of some statistical data (e.g., number of blocked messages from the same author, number of times the creator has been inserted in the BL) as well as data on creator profile (e.g., relationships with the wall owner, age, sex). Further, we plan to test the robustness of our system against different adversary models. The development of a GUI to make easier BL and filtering rule specification is also a direction we plan to investigate.

However, we aware that a new GUI could not be enough, representing only the first step. Indeed, the proposed system may suffer of problems similar to those in the specification of privacy settings in OSN. In this context, many empirical studies [22] show that average OSN users have difficulties in understanding also the simple privacy settings provided by today OSNs. To overcome this problem, a promising trend is to exploit data mining techniques to infer the best privacy preferences to suggest to OSN users, on the basis of the available social network data [8]. As future work, we intend to exploit similar techniques to infer BL and filtering rules.

References

1. Ali, B., Villegas, W., Maheswaran, M.: 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 (2007)
2. Amati, G., Crestani, F.: Probabilistic learning for selective dissemination of information. Information Processing and Management 35(5), 633–654 (1999)
3. Boykin, P.O., Roychowdhury, V.P.: Leveraging social networks to fight spam. IEEE Computer Magazine 38, 61–67 (2005)
4. Carminati, B., Ferrari, E.: Access control and privacy in web-based social networks. International Journal of Web Information Systems 4, 395–415 (2008)

5. Carminati, B., Ferrari, E., Perego, A.: Enforcing access control in web-based social networks. ACM Trans. Inf. Syst. Secur. 13(1), 1–38 (2009)
6. Carullo, M., Binaghi, E., Gallo, I.: An online document clustering technique for short web contents. In: Pattern Recognition Letters. vol. 30, pp. 870–876 (July 2009)
7. Churcharoenkrung, N., Kim, Y.S., Kang, B.H.: Dynamic web content filtering based on user's knowledge. International Conference on Information Technology: Coding and Computing 1, 184–188 (2005)
8. Fang, L., LeFevre, K.: Privacy wizards for social networking sites. In: WWW '10: Proceedings of the 19th international conference on World wide web. pp. 351–360. ACM, New York, NY, USA (2010)
9. Fong, P.W.L., Anwar, M.M., Zhao, Z.: A privacy preservation model for facebook-style social network systems. In: Proceedings of 14th European Symposium on Research in Computer Security (ESORICS). pp. 303–320 (2009)
10. Frakes, W., Baeza-Yates, R. (eds.): Information Retrieval: Data Structures & Algorithms. Prentice-Hall (1992)
11. Golbeck, J.A.: Computing and Applying Trust in Web-based Social Networks. Ph.D. thesis, PhD thesis, Graduate School of the University of Maryland, College Park (2005)
12. Hanani, U., Shapira, B., Shoval, P.: Information filtering: Overview of issues, research and systems. User Modeling and User-Adapted Interaction 11, 203–259 (2001)
13. Kim, Y.H., Hahn, S.Y., Zhang, B.T.: Text filtering by boosting naive bayes classifiers. In: SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 168–175. ACM, New York, NY, USA (2000)
14. Landis, J.R., Koch, G.: The measurement of observer agreement for categorical data. Biometrics 33(1), 159–174 (March 1977)
15. Laudanna, A., Thornton, A., Brown, G., Burani, C., Marconi, L.: Un corpus dell'italiano scritto contemporaneo dalla parte del ricevente. III Giornate internazionali di Analisi Statistica dei Dati Testuali 1, 103–109 (1995)
16. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research (2004)
17. Manning, C., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge, UK (2008)
18. Moody, J., Darken, C.: Fast learning in networks of locally-tuned processing units. In: Neural Computation. vol. 1, pp. 281–294 (1989)
19. Pérez-Alcázar, J.d.J., Calderón-Benavides, M.L., González-Caro, C.N.: Towards an information filtering system in the web integrating collaborative and content based techniques. In: LA-WEB '03: Proceedings of the First Conference on Latin American Web Congress. p. 222. IEEE Computer Society, Washington, DC, USA (2003)
20. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management 24(5), 513–523 (1988)
21. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
22. 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)
23. Tootoonchian, A., Gollu, K.K., Saroiu, S., Ganjali, Y., Wolman, A.: Lockr: social access control for web 2.0. In: WOSP '08: Proceedings of the first workshop on Online social networks. pp. 43–48. ACM, New York, NY, USA (2008)