ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

Context Aware Message Filtering in OSN user Walls

Dactchayani Sukumar[1], Maniraj .S.P[2]

Assistant Professor, Dept. Of CSE, S.R.M. University, Chennai, Tamil Nadu[1]

Assistant Professor, Dept. Of CSE, S.R.M. University, Chennai, Tamil Nadu[2]

ABSTRACT:Unwanted Messages which are sending and received on the user private space are filtered by allowing OSN users to have a direct control on the messages posted on their walls through Filtering Rules and Context Based Filtering Mechanism. A flexible rule-based system, that allows users to customize the filtering criteria to be applied to their walls. Machine Learning based soft classifier uses content based filtering method to filter the unwanted messages from the user walls. We have a system with a more sophisticated approach to decide when a user should be inserted into a BlackList

KEYWORDS: Machine learning, Content based filtering, Black list,Soft Classifier,Online social networks, Rule based system.

I. INTRODUCTION

OSN provides support to prevent unwanted messages on user walls, for example Facebook allows user to state who is allowed to insert messages in their walls(friends, friends of friends are defined groups of friends. we exploit Machine learning text categorization technics to automatically assign with each shortest message a set of categories based on its content. The major effects in building a robust short text classifier are concentrated in the extraction and selection of a set of characterizing and discriminate features.[10]However, the aim of the majority of these proposals is mainly to provide users a classification mechanism to avoid they are overwhelmed by useless data.

The original set of features, derived from endogenous properties of short texts, is enlarged here including exogenous knowledge related to the context from which the messages originate. As far as the learning model is concerned, we confirm in the current paper the use of neural learning which is today recognized as one of the most efficient solutions in text classification. In particular, we base the overall short text classification strategy on Radial Basis Function Networks (RBFN) for their proven capabilities in acting as soft classifiers, in managing noisy data and intrinsically vague classes. Moreover, the speed 2 in performing the learning phase creates the premise for an adequate use in OSN domains, as well as facilitates the experimental evaluation tasks .The first proposal of a system to automatically filter unwanted messages from OSN user walls on the basis of both message content and the message creator relationships and characteristics. The current paper substantially extends [4] for what concerns both the rule layer and the classification module. Major differences include, a different semantics for filtering rules to better fit the considered domain, an online setup assistant (OSA) to help users in FR specification, the extension of the set of features considered in the classification process, a more deep performance evaluation study and an update of the prototype implementation to reflect the changes made to the classification techniques

II. RELATED METHODS

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

A. SHORT TEXT CLASSIFIER

Classification of texts potentially containing a complex and specific terminology requires the use of learning methods that do not rely on extensive feature engineering. In this work we use prediction by partial matching (PPM), a method that compresses texts to capture text features and creates a language model adapted to a particular text. We show that the method achieves a high accuracy of text classification and can be used as an alternative to state-of-art learning algorithms[5].A general framework for building classifiers that deal with short and sparse text & Web segments by making the most of hidden topics discovered from large-scale data collections. The main motivation of this work is that many classification tasks working with short segments of text & Web, such as search snippets, forum & chat messages, blog & news feeds, product reviews, and book & movie summaries, fail to achieve high accuracy due to the data sparseness.

We, therefore, come up with an idea of gaining external knowledge to make the data more related as well as expand the coverage of classifiers to handle future data better. The underlying idea of the frame-work is that for each classification task, we collect a large-scale external data collection called "universal dataset", and then build a classifier on both a (small) set of labelled training data and a rich set of hidden topics discovered from that data collection. The framework is general enough to be applied to different data domains and genres ranging from Web search results to medical text. We did a careful evaluation on several hundred megabytes of Wikipedia (30M words) and MEDLINE (18M words) with two tasks: "Web search domain disambiguation" and "disease categorization for medical text", and achieved significant quality enhancement.

B. PARSING

Parsing is the process of structuring a linear representation in accordance with a given grammar. This definition has been kept abstract on purpose, to allow as wide an interpretation as possible. The “linear representation” may be a sentence, a computer program, a knitting pattern, a sequence of geological strata, a piece of music, actions in ritual behaviour, in short any linear sequence in which the preceding elements in some way restrict† the next element. For some of the examples the grammar is well-known, for some it is an object of research and for some our notion of a grammar is only just beginning to take shape. The bulk of examples of CF grammars originate from programming languages. Sentences in these languages (that is, programs) have to be processed automatically (that is, by a compiler) and it was soon recognized (around 1958) that this is a lot easier if the language has a well-defined formal grammar. The syntaxes of almost all programming languages in use today are defined through a formal grammar. Some authors (for instance, Chomsky) and some parsing algorithms, require a CF grammar to be monotonic. The only way a CF rule can be non-monotonic is by having an empty right-hand side; such a rule is called an ε-rule and a grammar that contains no

Such rules are called ε-free. The requirement of being ε-free is not a real restriction, just a nuisance. Any CF grammar can be made ε-free be systematic substitution of the ε- rules.[5]But this in general does not improve the appearance of the grammar. The issue will be discussed further in Section. The basic property of CF grammars is that they describe things that nest: an object may contain other objects in various places, which in turn may contain ... etc. When during the production process we have produced one of the objects, the right-hand side still “remembers” what has to come after it: n the English grammar, after having descended into the depth of the non-terminal Subject to produce something like the wistful cat, the right-hand side Subject Verb Object still remembers that a Verb must follow. While we are working on the Subject, the Verb and Object remain queued at the right in the sentential form.

C. BACK PROPAGATION

Getting a learning set of various unwanted messages and spam emails, Parsing the individual mails to extract the words of interest. (Porter method) Implementing the Back propagation Algorithm.

The back propagation algorithm looks for the minimum of the error function in weight space using the method of gradient descent. The combination of weights which minimizes the error function is considered to be a solution of the learning problem. Since this method requires computation of the gradient of the error function at each iteration step, we must guarantee the continuity and differentiability of the error function. Obviously we have to use a kind of activation function other than the step function used in perceptron’s,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

We use Backpropagation Algorithm because the composite function produced by interconnected perceptron's is discontinuous, and therefore the error function too. One of the more popular activation functions for backpropagation networks is the sigmoid, a real function SC: IR! (0, 1) defined by the expression

The constant c can be selected arbitrarily and its reciprocal 1/c is called the temperature parameter in stochastic neural networks. The shape of the sigmoid changes according to the value of c, The graph shows the shape of the sigmoid for c = 1, c = 2 and c = 3. Higher values of c bring the shape of the sigmoid closer to that of the step function and in the limit c ! 1 the sigmoid converges to a step function at the origin. In order to simplify all expressions derived in this chapter we set c = 1, but after going through this material the reader should be able to generalize all the expressions for a variable c. In the following we call the sigmoid s1(x) just S(X)

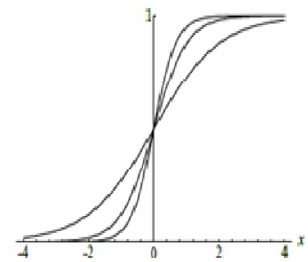

FIG 1

Fig. 7.1. Three sigmoids (for $c = 1$, $c = 2$ and $c = 3$)

Fig.1. Three sigmoid (for c = 1, c = 2 and c = 3)

The derivative of the sigmoid with respect to x, needed later on in this chapter, is

$$\frac{d}{dx}s(x) = \frac{e^{-x}}{(1+e^{-x})^2} = s(x)(1-s(x)).$$

We have already shown that, in the case of perceptron's, a symmetrical activation function has some advantages for learning. An alternative to the sigmoid is the symmetrical sigmoid S(x) defined as

$$S(x) = 2s(x) - 1 = \frac{1-e^{-x}}{1+e^{-x}}.$$

This is nothing but the hyperbolic tangent for the argument x/2 whose shape is shown in Figure 2 (upper right). The figure shows four types of continuous "squashing" functions. The ramp function (lower right) can also be used in

R. Rojas: Neural Networks, Springer-Verlag, Berlin, 1996

TABLE I
LEVELS OF CLASSIFICATION

I LEVEL CLASSIFIER	II LEVEL CLASSIFIER
Health	Health Poison
Boy	Bad Boy
Issue	Political Issue

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

Table 1 explains the Level of Classification were in I Level classifier the unwanted words are identified and as a result of Back Propagation algorithm. It checks for the II Level classifier were the word patterns are searched and if it matches the pattern which is stored in block list than the messages consider as a unwanted message, and it is prevented from sending.

D. ARCHITECTURE OF FILTERING UNWANTED MESSAGE

1) The message which is send through is stored as a chatting documents, this chatting documents is given to parsing for filtering basic verbs & non verbs

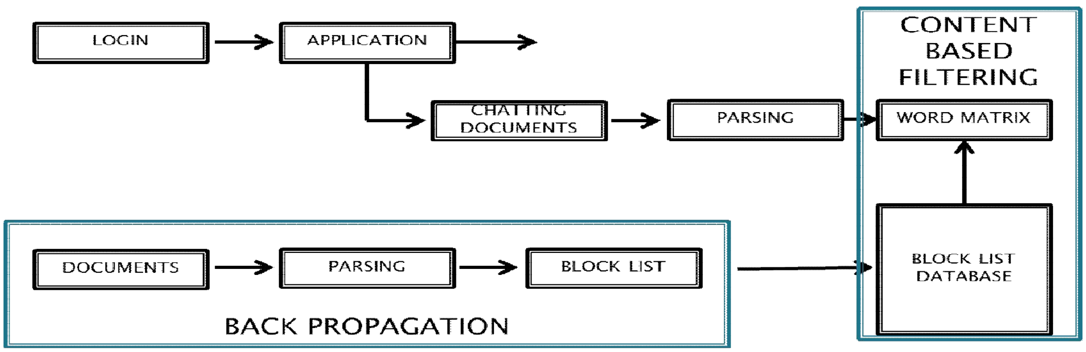

Fig 2: filtered wall architecture

2) If a probability of occurrence of a particular word exceeds than its threshold then the word is stored in word matric

3) Using back propagation the matching pattern of a particular word is stored in the block list

4) The matching word which is present in block list matches the word present in word matrics, then the message containing that word is filtered

E.MACHINE BASED LEARNING

In content-based recommendation methods, the utility u(c, s) of item s for user c is estimated based on the utilities ,i u c s assigned by user c to items is ∈S that are " similar " to item s. For example, in a movie recommendation application, in order to recommend movies to user c, the content-based recommender system tries to understand the commonalities among the movies user c has rated highly in the past (specific actors, directors, genres, subject matter, etc.). Then, only the movies that have a high degree of similarity to whatever user's preferences are would get recomended.The content-based approach to recommendation has its roots in information retrieval and information filtering [1] research. Because of the significant and early advancements made by the information retrieval and filtering communities and because of the importance of 6 several text-based applications, many current content-based systems focus on recommending items containing textual information, such as documents, Web sites (URLs), and Usenet news messages. The improvement over the traditional information retrieval approaches comes from the use of user profiles that contain information about users' tastes, preferences, and needs. The profiling information can be elicited from users explicitly, e.g., through questionnaires, or implicitly – learned from their transactional behavior over time. More formally, let Content(s) be an item profile, i.e., a set of attributes characterizing item s. It is usually computed by extracting a set of features from item s (its content) and is used

to determine appropriateness of the item for recommendation purposes. Since, as mentioned earlier, content-based systems are designed mostly to recommend text-based items, the content in these systems is usually described with keywords. For example, a content-based component of the Fab system [8], which recommends Web pages to users, represents Web page content with the 100 most important words. Similarly, the Syskill&Webert system [1] represents documents with the 128 most informative words. The "importance" (or "informa-tiveness") of word ki in document dj is determined with some weighting measure wij that can be defined in several different ways. One of the best-known measures for specifying keyword weights in Information Retrieval is the term frequency/inverse document frequency (TF-IDF) measure [89] that is defined as follows. Assume that N is the total number of documents that can be recommended to users and that keyword ki appears in ni of them. Moreover, assume that i, j f is the number of times keyword ki appears in document dj. Then i, j TF , the term frequency (or normalized frequency) of keyword ki in document dj, is defined as where the maximum is computed over the frequencies z, j f of all keywords kz that appear in the document dj. However, keywords that appear in many documents are not useful in distinguishing between a relevant document and a non-relevant one. Therefore, the measure of inverse document frequency (IDFi) is often used in combination with simple term frequency (i, j TF). The inverse document frequency for keyword ki is usually defined as

$$TF_{i,j} = \frac{f_{i,j}}{\max_z f_{z,j}}$$

where the maximum is computed over the frequencies z, j f of all keywords kz that appear in the document dj. However, keywords that appear in many documents are not useful in
distinguishing between a relevant document and a non-relevant one. Therefore, the measure of inverse document frequency (IDFi) is often used in combination with simple term frequency
(i, j TF). The inverse document frequency for keyword ki is usually defined as

$$TF_{i,j} = \frac{f_{i,j}}{\max_z f_{z,j}}$$

ed over the frequencies $f_{z,j}$

Then the TF-IDF weight for keyword ki in document dj is defined as i, j i, j i w = TF × IDF and the content of document dj is defined as Content(dj) = (w1j, …wkj). As stated earlier, content-based systems recommend items similar to those that a user liked in the past. In particular, various candidate items are compared with items previously rated by the user, and the best-matching item(s) are recommended. More formally, let ContentBasedProfile(c) be the profile of user c containing tastes and preferences of this user. These profiles are obtained by analysing the content of the items previously seen and rated by the user and are usually constructed using keyword analysis techniques from information retrieval. For example, ContentBasedProfile(c) can be defined as a vector of weights (wc1, …,wck), where each weight wci denotes the importance of keyword ki to user c and can be computed from individually rated content vectors using a variety of techniques. For example, some averaging approach, such as Rocchio algorithm [85], can be used to compute ContentBasedProfile(c) as an "average" vector from an individual content vectors [8, 56]. Onthe other hand, [3] use a Bayesian classifier in order to estimate the probability of a document. To work well, a ML-based classifier needs to be trained with a set of sufficiently complete and consistent pre classified data. The difficulty of satisfying this constraint is essentially related to the subjective character of the interpretation process with which an expert decides whether to classify a document under a given category. In order to limit the effects of this phenomenon, known in literature under the name of inter indexer inconsistency [6], our strategy contemplates the organization of "tuning sessions" aimed at establishing a consensus among experts through discussion of the most controversial interpretation of messages.

1) *Content Based Methods :*Content-based filtering relies on creating associations between items in a collection. When a user shows a preference for specific items, the system compares those items to others in the collection. Items with a high degree of similarity are presented as recommendations. Pure content-based recommendations ignore the preferences of other users (Schein, Popescul, &Ungar, 2002).There are a number of methods that can be used to generate a list of similar items. In the simplest form,this can be thought of as grouping items based upon their genre or subject matter. However, content-based filtering takes this concept further by increasing the number of terms that may be considered to compare items. For example, a collection of movies could be compared based on genre, actors, director, subject, parental guidance rating, or review ratings. This allows the filtering system to recommend items based on a much larger range of aspects than searching alone would allow.In the case of article and website recommendations, a slightly different system is used. One method weights articles based on the number of times specific keywords appear in the article compared to the overall rarity of that keyword in the articles indexed. Therefore, items that contain terms that relate well to the searchand which are statistically less likely to be common are suggested first (Adomavicius&Tuzhilin, 2005).

In content-based filtering, each user is assumed to operate independently. As a result, a content-based filtering system selects information items based on the correlation between the content of the items and the user preferences as opposed to a collaborative filtering system that chooses items based on the correlation between people with similar preferences [8], [9]. Content-based filtering techniques suffer from the problems of limited content analysis, overspecialization, and new users. Limited content analysis refers to the set of attributes that any given item in the collection has attached to it being unacceptably small [4]. In cases where materials are easily indexed by computers (digital articles, websites, etc.) or where rules and standards explicitly define the attributes associated with each item (MARC records, library holdings) this is usually not a problem. However, multimedia, videos, music, and pictures can be much moredifficult to accurately describe, and often these descriptions are subjective to the person doing the cataloguing. Without accurate methods of comparing these types of materials, content-based filtering is ineffective.

In addition, overspecialization occurs when these systems judge individual items based on a limited number of features. This results in the system recommending items that are either too alike to what the user has seen in the past or items which are content identical to other recommendations shown at the same time. Occasionally, this problem can be resolved by introducing a small amount of randomness. Lastly, new users present a problem because the system has yet to receive adequate user profile information about them to recommend reliably (Adomavicius&Tuzhilin, 2005).

2) *Black List :*A further component of our system is a BL mechanism to avoid messages from undesired creators, independent from their contents. BLs is directly managed by the system, which should be able to determine who are the users to be inserted in the BL and decide when users retention in the BL is finished. To enhance flexibility, such information is given to the system through a set of rules, hereafter called BL rules. Such rules are not defined by the SNM, therefore they are not meant as general high level directives to be applied to the whole community. Rather, we decide to let the users themselves, i.e., the wall's owners to specify BL rules regulating who has to be banned from their walls and for how long. Therefore, a user might be banned from a wall, 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 according to their profiles as well as their relationships in the OSN. Therefore, by means of a BL rule, wall owners are for example able to ban from their walls users they do not directly know (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 may also take into account users' behaviour in the OSN. More precisely, among possible information denoting users' bad behaviour 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 a 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 behaviour is not improved. This principle works for those users that have been already inserted in the considered BL at least one time. In contrast, to catch new bad behaviours, we use the Relative Frequency (RF) that let the system be able to detect those users whose messages continue to fail the FRs. The two measures can be computed either locally, that is, by considering only the messages and/or the BL of the user specifying the BL rule or globally, that is, by considering all OSN users walls and/or BLs.

III. CONCLUSION

In this paper ,we provide capability for the system to Filter unwanted messages from OSN user walls. The development of a GUI and a set of related tools to make easier BL and FR specification is also a direction we plan to investigate, since usability is a key requirement for such kind of applications. In particular, we aim at investigating a tool able to automatically recommend trust values for those contacts user does not personally known. We do believe that such a tool should suggest trust value based on users actions, behaviors, and reputation in OSN, which might imply to enhance OSN with audit mechanisms.Thus this paper provides two levels of Filtering capabilities

REFERENCES

1. A. Adomavicius and G. Tuzhilin, "Toward the Next Gen-eration ofRecommender Systems: A Survey of the State-of-the-Art andPossible Extensions," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, pp. 734-749, June 2005.
2. S. Zelikovitz and H. Hirsh, "Improving Short Text Classifi-cationUsing Unlabeled Background Knowledge," Proc. 17th Int'l Conf.Machine Learning (ICML '00), P. Langley, ed., pp. 1183-1190, 2000.
3. Marco Vanetiti,ElisabettaBinaghi,"A system to Filter un-wanted messages from OSN user walls

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

4. M. Chau and H. Chen, "A Machine Learning Approach to WebWebPage Filtering Using Content and Structure Analysis," DecisionSupport Systems, vol. 44, no. 2, pp. 482-494, 2008.
5. WWW.GOOGLE.COM
6. C. Cleverdon, "Optimizing Convenient Online Access to BibliographicDatabases," Information Services and Use, vol. 4, no. 1,pp. 37-47, 1984.
7. C. Cleverdon, "Optimizing Convenient Online Access to BibliographicDatabases," Information Services and Use, vol. 4, no. 1,pp. 37-47, 1984.
8. P.J. Denning, "Electronic Junk," Comm. ACM, vol. 25, no. 3,pp. 163-165, 1982.
9. P.W. Foltz and S.T. Dumais, "Personalized Information Delivery:An Analysis of Information Filtering Methods," Comm. ACM,vol. 35, no. 12, pp. 51-60, 1992.
10. R.J. Mooney and L. Roy, "Content-Based Book RecommendingUsing Learning for Text Categorization," Proc. Fifth ACM Conf. Digital Libraries, pp. 195-204, 2000.

BIOGRAPHY

S.DACTCHAYANI has finished the bachelor degree in computer science & Engineering in Anand Institute of Higher Technology and currently pursuing Master degree in SRM University, in this topic she as attended conference

S.P.MANIRAJ has finished the Master degree in Computer science in Anna University (Thirunelveli). Currently working as Assistant Professor in SRM University