Using Compression Models for Filtering Troll Comments

Jorge de-la-Peña-Sordo, Igor Santos, and Pablo G. Bringas
S^{3}Lab, DeustoTech - Computing, University of Deusto, Bilbao, Spain
Email: {jorge.delapenya, isantos, pablo.garcia.bringas}@deusto.es

***Abstract*—Internet is evolving. How the content is generated has changed and currently, users and readers of a site can create content. They can express themselves showing their feelings or opinions commenting diverse stories or other users' comments in social news websites. This fact has led to negative side effects: the appearance of troll users and their contents seeking deliberately controversy. In this paper we propose a new method to filter trolling comments using compression models. Normally, Vector Space Model representation use is quite common but these filters can be attacked. To this end, we validate our approach with data from 'Menéame', a popular Spanish social news site, training several compression models, showing that our method can maintain high accuracy rates whilst making such filters difficult to defeat.**

I. Introduction

The Internet has evolved and now, not only the moderators and administrators of a website generate the content available in it, but also its readers or users, generating valuable information about them. In this particular scenario, social news websites such as Digg[1] or its Spanish variation 'Menéame'[2] are very popular social sites that allow users to send news stories, so other users may read, rate and comment them. The publishing criteria is usually vote-driven, where other users of those systems rate each story them by voting and the most voted stories appear in the frontpage [1].

We focus on 'Menéame'. This social news website has already a method for automatic moderation of comments and stories to automatically filter them. However, it is based on the votes of other users and, therefore, it can be manipulated. To avoid this problem, we have selected a more linguistic and statistical representation of the comments. There are approaches to filter spam in reviews [2], [3], that can be applied to this particular domain.

In our previous work [4], we proposed an approach able to automatically categorise comments in these social news sites using supervised machine-learning algorithms. These approaches model sites using the Vector Space Model (VSM) [5], an algebraic approach for Information Filtering (IF), Information Retrieval (IR), indexing and ranking. This model represents natural language documents in a mathematical manner through vectors in a multidimensional space.

Nevertheless, this method has its shortcomings. For instance, in spam filtering, which is a type of text filtering, *Good Word Attack*, a method that modifies the term statistics by appending a set of words that are characteristic of legitimate e-mails, or tokenisation, that works against the feature selection of the message by splitting or modifying key message features rendering the term-representation no longer feasible [6], have been applied by spammers.

In light of this background, we present the first troll comments filter for social news websites that is based on compression methods for text filtering. These kind of models are very robust to noise, fast to construct and incrementally updateable and they should make such filters difficult to defeat.

In summary, our main contributions are:

- A new method to represent comments in social news websites.
- An adaptation of the compression model approach to comment filtering.
- An empirical validation which shows that our method can maintain high accuracy rates whilst making such filters difficult to defeat.

The remainder of this paper is structured as follows. Section II describes in detail our proposed method. Section III describes the compression algorithms we applied to this particular task. Section IV describes the experimental procedure and discussed the obtained results. Finally, Section V concludes and discusses the main limitations of this work.

II. Method Description

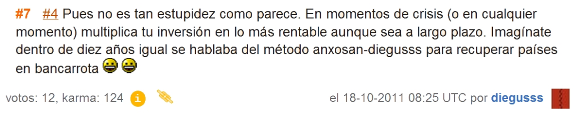
'Menéame' is a Spanish social news website, in which news and stories are promoted. It was developed in later 2005 by Ricardo Galli and Benjamín Villoslada and it is currently licensed as free software. At the beginning, it was focused on scientific and technological topics, but nowadays it is open to any topic such as politics, society or sports. Also, as the number of the users of 'Menéame' grew, so did the quality and quantity of the contributions.

Any user (even if it is not registered in the system) can vote the news stories in the front page or in the pending section, which are news that have not been contrasted yet. Registered users can send news to the system. A news story is held in the pending queue. There, the story will be voted by different readers or users. Registered users can also make a negative vote and comment the news story.

'Menéame' ranks their users depending on their 'karma'. The 'karma' is a value between 0 and 20. When a new user is registered a value of 6 point of 'karma' is given. 'Karma' is computed based on the performed activity in

[1]`http://digg.com/`
[2]`http://www.meneame.net/`

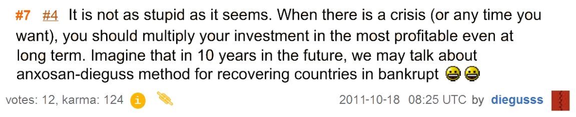
the previous 2 days. To this end, the algorithm combines 4 different components: positives votes received of the sent news, positive votes made, negative votes made and votes received of a user's comments. When a news story is in the pending queue, the 'karma' of the users that vote the story are added to its value and if they surpass a threshold they are published in front page. Otherwise, the stories that accumulate negative votes, will be sent to the discarded section. Usually, these contributions are either irrelevant, old, bothering, sensationalist, spam, replicated, micro-blogging, mistaken or plagiarism.



#7 #4 Pues no es tan estupidez como parece. En momentos de crisis (o en cualquier momento) multiplica tu inversión en lo más rentable aunque sea a largo plazo. Imagínate dentro de diez años igual se hablaba del método anxosan-diegusss para recuperar países en bancarrota

votos: 12, karma: 124 el 18-10-2011 08:25 UTC por diegusss

(a) Example of a comment in Spanish.



#7 #4 It is not as stupid as it seems. When there is a crisis (or any time you want), you should multiply your investment in the most profitable even at long term. Imagine that in 10 years in the future, we may talk about anxosan-dieguss method for recovering countries in bankrupt

votes: 12, karma: 124 2011-10-18 08:25 UTC by diegusss

(b) Translated comment.

Fig. 1. Example of a comment.

Figure 1 shows a comment in 'Menéame'. The first thing that appears is the number of the comment, which in this case is seven. Next, another number appears that references another comment. In this case the user is giving an opinion about a previous comment.

We categorise the comments in three different classifications. In order to make the explanation clearer, we show actual examples from of the different categories for each of the different classifications. These examples have been taken from the story shown in Figure 2.

Title: The Government has been asked to prevent an 'atheist procession' the Maundy Thursday which will be at the same time that the traditional one in Lavapiés

Description: 'HazteOir' group, the political party 'Alternativa Española', churches like 'Santo Miguel Arcángel' and other catholic collectives asked the Government in Madrid to not allow the 'atheist procession' on Maundy Thursday.

Tags: Church, imposition, atheism, prohibition, government

Fig. 2. An example of a news story.

Each one of the three different classifications have several possible classes. They are the following ones:

- **Type of Information:** The type of information indicates what the user is doing in its comment. It can be:
  - *Contribution:* The user contributes by adding new information. Figure 3 shows an example of a contribution comment in the previous story.
  - *Irrelevant:* These comments do not contribute to the main article neither to another previous comment. Figure 4 shows an irrelevant comment.
  - *Opinion:* These comments express the user's particular opinion about the topic discussed in the story. Figure 5 shows an opinion.

#201 #191 By the way, I forgot telling you that in that town I told you before, the one where a church installed a megaphone to say the prayers before the Sunday service and made every inhabitant to get up early even if they don't want to, the major was of the political party PSOE and he didn't do anything to solve it. There is a clear contradiction between the public agenda of his political party and his actions, but some people do like their chair, don't they?

votes: 0, karma: 8, 18 hours and 23 minutes ago by strel

Fig. 3. An example of a contribution.

#225 Good luck, friends!

votes: 0, karma: 6, 12 hours and 46 minutes ago by pavlenka

Fig. 4. An example of an irrelevant comment.

#205 #70 I agree with you, and moreover, I think they should do it even harder, because so much it has cost us to be free of a belief or, at least, reduce it to, now, let them play with our morality and, also, for sociocultural reasons, to be expanded in the future (there is a rapid growth of the population professing such belief)

votes: 1, karma: -1, 18 hours 1 minute ago by RaistlinMajere

Fig. 5. An example of an opinion.

- **Focus of the comment:** The comment can be focused either on the main story or on another comment. Figure 6 shows an example of a comment that focuses on the main story whilst Figure 7 shows an example of a comment focused on another comment.

#202 The same that always happens: one gets molested because he/she wants, no because he/she should. If they concentrate more, they would realise that there are different stuffs than themselves and their truth but that damn egocentric point view, thinking that everyone is against me...

votes: 0, karma: 6, 18 hours 13 minutes ago by Natxo-Pistatxo

Fig. 6. An example of a comment focusing in the main story.

#204 #201 Then, you cannot be affiliated to PSOE and be a religious guy?... Indeed, taking into consideration that the PSOE is a center-right party, I do not really get where is the surprising fact. Would you understand it if it was from the PP?

votes: 0, karma: 6, 18 hours 8 minutes ago by xaphoo

Fig. 7. An example of a comment focusing in another comment.

- **Controversy Level:** We categorise the controversy level in three degrees: normal, controversial and highly controversial and, also, an additional one that is used for funny or ironic comments.
  - *Normal:* A normal comment is the one that raises no controversy or irony. Figure 8 shows a normal comment.

#205 #70 I agree with you, and moreover, I think they should do it even harder, because so much it has cost us to be free of a belief or, at least, reduce it (and I say very nuanced), for, now let them to play with our morality and also for sociocultural reasons it will be expanded in the future (there is a rapid growth of the population professing such belief)

votes: 0, karma: 6 for 18 hours 13 minutes ago by Natxo-Pistatxo

Fig. 8. An example of a normal comment.

  - *Controversial:* A comment that, on purpose, seeks controversy with an harmful tone. Figure 9 shows an example of a controversial comment.

#218 A lot of people being proud of their atheism, and the 90% of them made gifts to their children in Xmas..., and rate this comment negative if you dare, I don't mind.

votes: 0, karma: 6, 14 hours 46 minutos ago by canaam

Fig. 9. An example of a controversial comment.

- *Very Controversial:* It seeks to create controversy in an exaggerated way, being hurtful or disrespectful. In other words, a troll user. Figure 10 shows a very controversial comment.

#206 #10 You say that there have not been protests against the use of the burka in Spain? And not only by atheists, of course. When we have the same number of asshole fundamentalist both Christians and Muslims in Spain, we will be the first ones to proclaim our stupidity to the four winds. We are not afraid either of you or them.

votes: 0, karma: 6, 17 hours 57 minutes ago by Despero

Fig. 10. An example of a very controversial comment.

- *Joke:* These comments try to make a joke and be funny. Figure 11 shows an example.

If my colleagues and me walk through the streets handing out oil, grease, wax or other products likely to cause slips, we will have problems. The least they will call us will be vandals or we will be fined, or both.

votes: 0, karma: 6, 13 hours 54 minutes ago by pacorron

Fig. 11. An example of a joke comment.

III. Compression Models

A. Lempel-Ziv 78 (LZ78)

The LZ78 algorithm is among the most popular lossless compression algorithms [7]. It is used as the basis of the Unix compress utility and other popular archiving utilities for PCs. It also has performance guarantees within several analysis models.

Given a sequence $q_1^n \in \sum^n$, LZ78 incrementally parses q_1^n into non-overlapping adjacent 'phrases', which are collected into a phrase 'dictionary'. The algorithm starts with a dictionary containing the empty phrase ϵ. At each step the algorithm parses a new phrase, which is the shortest phrase that is not yet in the dictionary. Clearly, the newly parsed phrase s' extends an existing dictionary phrase by one symbol; that is, $s' = s\sigma$, where s is already in the dictionary (s can be the empty phrase). For compression, the algorithm encodes the index of s' (among all parsed phrases) followed by a fixed code for σ. Also observe that LZ78 compresses sequences without explicit probabilistic estimates.

Here is an example of this LZ78 parsing: if $q_1^{11} = abracadabra$, then the parsed phrases are $a|b|r|ac|ad|ab|ra$. Observe that the empty sequence ϵ is always in the dictionary and is omitted in our discussions.

B. Prediction by Partial Match (PPM)

The Prediction by Partial Match (PPM) algorithm [8] is considered to be one of the best lossless compression algorithms. The algorithm requires an upper bound D on the maximal Markov order of the Variable order Markov Models (VMM) it constructs. PPM handles the zero frequency problem using two mechanisms called escape and exclusion.

There are several PPM variants distinguished by the implementation of the escape mechanism. In all variants the escape mechanism works as follows. For each context s of length $k \leq D$, we allocate a probability mass $P_k(escape|s)$ or all symbols that did not appear after the context S (in the training sequence). The remaining mass 1-$P_k(escape|s)$ is distributed among all other symbols that have non-zero counts for this context. The particular mass allocation for 'escape' and the particular mass distribution $P_k(\sigma|s)$, over these other symbols σ, determine the PPM variant.

C. Probabilistic Suffix Trees (PST)

The Probabilistic Suffix Tree (PST) prediction algorithm [9] attempts to construct the single 'best' D-bounded VMM according to the training sequence. It is assumed that an upper bound D on the Markov order of the 'true source' is known to the learner.

A PST over $\sum$ is a non empty rooted tree, where the degree of each node varies between zero (for leaves) and $|\sum|$. Each edge in the tree is associated with a unique symbol in $\sum$. These edge labels define a unique sequence s for each path from a node v to the root. The sequence s labels the node v. Any such PST tree induces a 'suffix set' S consisting of the labels of all the nodes. The goal of the PST learning algorithm is to identify a good suffix set S for a PST tree and to assign a probability distribution $P(\sigma|s)$ over $\sum$, for each $s \in S$. Note that a PST tree may not be a tree source. The reason is that the set S is not necessarily proper.

D. An Improved Lempel-Ziv Algorithm (LZ-MS)

There are plenty of variations on the classic LZ78 compression algorithm [10]. The algorithm has two parameters M and S and, therefore, its acronym here is LZ-MS. A major advantage of the LZ78 algorithm is its speed. This speed is made possible by compromising on the systematic counts of all sub-sequences. While for a very long training sequence this compromise will not affect the prediction quality significantly, for short training sequences the LZ78 algorithm yields sparse and noisy statistics. Another disadvantage of the LZ78 algorithm is its loss of context when calculating $P(\sigma|s)$. These are two major deficiencies of LZ78. The LZ-MS algorithm attempts to overcome both these disadvantages by introducing two corresponding heuristics. This algorithm improves the LZ78 predictions by extracting more phrases during learning and by ensuring a minimal context for the next phrase, whenever possible.

The first modification is termed *input shifting*. It is controlled by the S parameter and used to extract more phrases from the training sequence. The second modification is termed *back-shift parsing* and is controlled by the M parameter. *Back-shift parsing* attempts to guarantee a minimal context of M symbols when calculating probabilities. Both *input shifting* and *back-shift parsing* tend to enhance the statistics we extract from the training sequence. In addition, *back-shift parsing* tends to enhance the utilization of the extracted statistics.

E. Dynamic Markov Chain (DMC)

The compression algorithm dynamic Markov compression (DMC) [11] models information with a finite state machine. Associations are built between every possible symbol in the source alphabet and the probability distribution over those symbols. This probability distribution is used to predict the next binary digit. The DMC method starts in a already defined state, changing the state when new bits are read from the entry. The frequency of the transitions to either a 0 or a 1 are summed when a new symbol arrives. The structure can be also be updated using a state cloning method.

F. Binary Context Tree Weighting Method (BICTW)

It is a naive application of the standard binary CTW algorithm over a binary representation of the sequence. The binary representation is naturally obtained when the size of the alphabet is a power of 2. Suppose that $k = log_2|\sigma|$ is an integer. In this case, we generate a binary sequence by concatenating binary words of size k, one for each alphabet symbol. If $log_2|\sigma|$ is not an integer we take $k = [log_2|\sigma|]$. We denote the resulting algorithm by BICTW. A more sophisticated binary decomposition of ctw was considered by [12].

G. Decomposed Context Tree Weighting Method (DECTW)

The DECTW [13] uses a tree-based hierarchical decomposition of the multi-valued prediction problem into binary problems. Each of the binary problems is solved via a slight variation of the binary CTW algorithm.

Let $\sum$ be an alphabet with size $k = |\sum|$. Consider a full binary tree $T_{\sum}$ with k leaves. Each leaf is uniquely associated with a symbol in $\sum$. Each internal node v of $T_{\sum}$ defines the binary problem of predicting whether the next symbol is a leaf on v's left subtree or a leaf on v's right subtree.

IV. Empirical Validation

This section describes the validation of our compression-based approach against a comment dataset gathered from 'Menéame'. We gathered a collection of comments from the 5th of April, 2011 to 12th of April, 2011[3]. This dataset of comments comprises one week of stories filled by 9,044 comment instances. We evaluated the precision of our proposed method. To this end, by means of the dataset, we conducted the following methodology:

A. Methodology

- **Cross validation:** This method is generally applied in machine-learning evaluation [14]. In our experiments, we performed a K-fold cross validation with $k = 10$. In this way, our dataset is 10 times split into 10 different sets of learning (90 % of the total dataset) and testing (10 % of the total data).
- **Learning the model:** We accomplished the learning step using different learning algorithms depending on the specific model, for each fold. As discussed above, we employed the implementations of the compression classification provided by the *CompressionTextClassifier* package[4] for machine-learning tool WEKA [15]. In our experiment approaches, we used the following models with their default configurations:
  - *Binary Context Tree Weighting Method (BICTW).*
  - *Decomposed Context Tree Weighting Method (DECTW).*
  - *Dynamic Markov Chain (DMC).*
  - *Lempel-Ziv 78 (LZ78).*
  - *Improved Lempel-Ziv Algorithm (LZ-MS).*
  - *Prediction by Partial Match (PPM).*
  - *Probabilistic Suffix Trees (PST).*
- **Testing the models:** To test the approach, we measured the accuracy (Acc) of the models (i.e., percent of correctly classified comments or the total number of hits of the classifiers divided by the number of instances in the whole dataset).

B. Results

To evaluate the contribution of the Compression Models to categorise trolling comments, we compared the filtering capabilities of our method with the ones obtained with a classic VSM model in our previous work [4]. In order to represent the previous comment dataset, we developed two different procedures to construct the VSM of the comment body: (i) VSM with words and terms, and (ii) n-grams with different values of n (n = 1, n = 2, n = 3). Furthermore, we removed every word devoid of meaning in the text, called stop words, (e.g., 'a','the','is') [16]. To this end, we employed an external stop-word list of Spanish words[5]. These words do not provide any semantic information and add noise to the model [17].

To represent the information contained in the comment body we have used an Information Retrieval (IR) model. It can be defined as a 4-tuple $[\mathcal{C}, F, \mathcal{Q}, R(q_i, c_j)]$ [18] where $\mathcal{C}$, is a set of representations of comments; F, is a framework for modelling comments, queries and their relationships; $\mathcal{Q}$, is a set of representations of user queries; and, finally, $R(q_i, c_j)$ is a ranking function that associates a real number with a query q_i ($q_i \in \mathcal{Q}$) and a comment representation c_j ($c_j \in \mathcal{C}$).

As $\mathcal{C}$ is the set of comments c, $\{c : \{t_1, t_2, ...t_n\}\}$, each comprising n terms $t_1, t_2, \ldots, t_n$, we define the weight $w_{i,j}$ as the number of times the term t_i appears in the comment c_j, if t_i is not present in c, $w_{i,j} = 0$. Therefore, a comment c_j can be represented as the vector of weights $\vec{c_j} = (w_{1,j}, w_{2,j}, ...w_{n,j})$.

On the basis of this formalisation, IR systems commonly use the Vector Space Model (VSM) [18], which represents comments algebraically as vectors in a multidimensional space. This space consists only of positive axis intercepts. Comments are represented by a term-by-comment matrix, where

[3]The labelled dataset can be downloaded at http://paginaspersonales.deusto.es/isantos/resources/Data_Meneame_dot_net_from_April_5th_to_April_12th.rar

[4]The package CompressionTextClassifier can be download at http://paginaspersonales.deusto.es/isantos/resources/CompressionTextClassifier-0.4.3.zip

[5]The list of stop words can be downloaded at: http://paginaspersonales.deusto.es/isantos/resources/stopwords.txt

the $(i,j)^{th}$ element illustrates the association between the $(i,j)^{th}$ term and the j^{th} comment. This association reflects the occurrence of the i^{th} term in comment j. Terms can represent diverse textual units (e.g., words or n-grams) and can also be individually weighted, allowing the terms to become more or less important within a comment or the collection $\mathcal{C}$ as a whole.

We used the *Term Frequency – Inverse Document Frequency* (TF–IDF) [17] weighting schema, where the weight of the i^{th} term in the j^{th} comment, denoted by $weight(i,j)$, is defined by: $weight(i,j) = tf_{i,j} \cdot idf_i$ where *term frequency* $tf_{i,j}$ is defined as: $tf_{i,j} = n_{i,j} / \sum_k n_{k,j}$ where $n_{i,j}$ is the number of times the term $t_{i,j}$ appears in a comment c, and $\sum_k n_{k,j}$ is the total number of terms in the comment c. The inverse term frequency idf_i is defined as: $idf_i = |\mathcal{C}| / |\mathcal{C} : t_i \in c|$ where $|\mathcal{C}|$ is the total number of comments and $|\mathcal{C} : t_i \in c|$ is the number of comments containing the term t_i.

As the terming schema we have employed two different alternatives. First, we used the word as term. Second, we used a n-gram approach. N-gram is the overlapping subsequence of n words from a given comment.

Next, we compared our method with some of the most used supervised machine-learning algorithms. Specifically, we use the following models:

- *Bayesian networks (BN):* We used different structural learning algorithms: K2 [19] and Tree Augmented Naïve (TAN) [20]. Moreover, we also performed experiments with a Naïve Bayes Classifier [14].
- *Support Vector Machines (SVM):* We performed experiments with a polynomial kernel [21], a normalised polynomial Kernel [22], a Pearson VII function-based universal kernel (PUK) [23] and a radial basis function (RBF) based kernel [24].
- *K-nearest neighbour (KNN):* We launched experiments with $k = 1, 2, 3, 4, 5$.
- *Decision Trees (DT):* We executed experiments with J48 (the *Weka* [15] implementation of the *C4.5* algorithm [25]) and Random Forest [26], an ensemble of randomly constructed decision trees. In particular, we employed $N = 10, 50, 100, 200$.

TABLE I
RESULTS IN TERMS OF ACCURACY (%) OF THE COMMENT CATEGORISATION FOR COMPRESSION MODELS.

Dataset	Type Info.	Focus Comm.	Controversy Level
BICTW	30.60	49.93	36.18
DECTW	52.79	69.85	32.22
DMC	75.43	68.51	64.53
LZ78	73.24	64.52	65.82
LZ-MS	73.84	64.57	66.73
PPM	74.96	75.03	53.01
PST	70.38	90.10	28.06

Table I shows the obtained results when the compression models are applied to our dataset. Table II shows the results with n-grams as tokens and supervised learning and Table III shows the results with words as tokens and supervised learning.

Regarding the results obtained in the classification *Type of Information*, the best classifier was SVM with a polynomial kernel when n-grams as tokens are applied: 76.56%, achieving

TABLE II
RESULTS IN TERMS OF ACCURACY (%) OF THE COMMENT CATEGORISATION FOR N-GRAM VSM.

Dataset	Type Info.	Focus Comm.	Controversy Level
KNN K = 1	54.88	82.43	60.80
KNN K = 2	59.24	78.64	67.70
KNN K = 3	50.96	80.44	67.49
KNN K = 4	54.11	78.51	68.65
KNN K = 5	48.49	79.80	69.09
Bayes K2	63.56	91.73	69.92
Bayes TAN	67.26	92.88	70.58
Naïve Bayes	28.67	81.87	30.44
SVM: PolyKernel	76.56	93.81	71.13
SVM: Norm. PolyKernel	72.96	91.40	71.20
SVM: PUK	76.55	92.13	71.03
SVM: RBFK	70.74	58.76	70.31
J48	72.25	91.20	70.45
Random Forest N = 10	74.60	93.22	67.01
Random Forest N = 50	75.18	93.86	67.39
Random Forest N = 100	75.26	93.89	67.43
Random Forest N = 200	75.30	93.96	67.43

TABLE III
RESULTS IN TERMS OF ACCURACY (%) OF THE COMMENT CATEGORISATION FOR WORD VSM.

Dataset	Type Info.	Focus Comm.	Controversy Level
KNN K = 1	64.27	82.43	64.54
KNN K = 2	69.62	78.64	69.33
KNN K = 3	65.32	80.44	69.12
KNN K = 4	68.36	78.51	70.11
KNN K = 5	64.90	79.80	70.36
Bayes K2	66.38	91.73	71.21
Bayes TAN	69.58	92.88	71.24
Naïve Bayes	26.08	81.87	28.61
SVM: PolyKernel	72.44	93.81	71.30
SVM: Norm. PolyKernel	72.62	91.40	71.21
SVM: PUK	74.25	92.13	71.40
SVM: RBFK	70.68	58.76	69.96
J48	70.61	91.20	70.86
Random Forest N = 10	73.29	93.20	67.08
Random Forest N = 50	73.45	93.89	66.99
Random Forest N = 100	73.45	93.93	67.04
Random Forest N = 200	73.53	93.96	67.07

the best compression algorithm the value 75.43%. In terms of *Focus of the Comment*, both VSM approaches achieved the best result: 93.96% by the Random Forest (with $k = 200$) algorithm, whilst the compression classifier PST obtained 90.10%. In the case of the last categorisation, *Controversy Level*, the better performance was obtained by the algorithm SVM with a *Pearson VII* kernel, when words as tokens are applied: 71.40%; however, in the compression classification, the LZ-MS algorithm achieved 66.73%. In summary, the compression models achieved results close to the supervised machine learning algorithms.

V. CONCLUSIONS AND FUTURE WORK

In this work, our main contribution is the first troll comments filter for social news websites that is based on compression methods for text filtering, due to exists a special problem for automated text categorization, of which the defining characteristic is that filters face an active adversary, which constantly attempts to evade filtering. In this case, the compression models not enhance the results obtained by VSM approaches, but these kind of models are very robust to noise, fast to construct and incrementally updateable and they should make such filters difficult to defeat. There are several topics of discussion in which we will focus in future versions of this system.

There is an issue derived from Natural Language Processing (NLP) when dealing with semantics: *Word Sense Disambiguation* (WSD). A troll user may evade our method by explicitly exchanging the key words of the comment with other polyseme terms and thus avoid detection. Thereby, WSD is considered necessary to perform most natural language processing tasks [27]. Hence, we propose the study of different WSD techniques (a survey of different WSD techniques can be found in [28]) able to provide a semantics-aware moderation tool. However, such a semantic approach for moderation should have to deal with the semantics of different languages [29] and, therefore, be language dependant.

Our technique has several limitations due to the representation of comments. For instance, in the context of spam filtering, most of the filtering techniques are based on the frequencies with which terms appear within messages and spammers have started modifying their techniques to evade such filters. These techniques can be applied by a troll user of a social news website. For example, *Good Word Attack* is a method that modifies the term statistics by appending a set of words that are characteristic of legitimate, thereby bypassing filters. Nevertheless, we can adopt some of the methods that have been proposed in order to improve spam filtering, such as *Multiple Instance Learning* (MIL) [30]. MIL divides an instance or a vector in the traditional supervised learning methods into several sub-instances and classifies the original vector based on the sub-instances [31]. Zhou et al. [32] proposed the adoption of multiple instance learning for spam filtering by dividing an e-mail into a bag of multiple segments and classifying it as spam if at least one instance in the corresponding bag was spam. We can adapt this approach to the our comment moderation tool. Another attack, known as *tokenisation*, works against the feature selection of the comment by splitting or modifying key message features, which renders the term representation as no longer feasible [6]. All of these attacks, which spammers have been adopting, should be taken into account in the construction of future filtering or moderation systems.

References

[1] K. Lerman. User participation in social media: Digg study. In Web Intelligence and Intelligent Agent Technology Workshops, 2007 IEEE/WIC/ACM International Conferences on (pp. 255-258). IEEE.

[2] N. Jindal and B. Liu. Review spam detection. In Proceedings of the 16th international conference on World Wide Web (pp. 1189-1190). ACM, 2007.

[3] N. Jindal and B. Liu. Opinion spam and analysis. In Proceedings of the 2008 International Conference on Web Search and Data Mining (pp. 219-230). ACM.

[4] I. Santos, J. De-La-Peña-Sordo, I. Pastor-López, P. Galan-García and P. G. Bringas. Automatic categorisation of comments in social news websites. Expert Systems with Applications, 39(18), 13417-13425, 2012.

[5] G. Salton, A. Wong and C. S. Yang. A vector space model for automatic indexing. Communications of the ACM, 18(11), 613-620, 1975.

[6] G. L. Wittel and S. F. Wu. On Attacking Statistical Spam Filters. In CEAS, 2004.

[7] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. Information Theory, IEEE Transactions on, 24(5), 530-536, 1978.

[8] J. G. Cleary and I. Witten. Data compression using adaptive coding and partial string matching. Communications, IEEE Transactions on, 32(4), 396-402, 1984.

[9] D. Ron, Y. Singer, and N. Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. Machine learning, 25(2-3), 117-149, 1996.

[10] T. C. Bell, J. G. Cleary, and I. H. Witten. Text compression. Prentice-Hall, Inc., 1990.

[11] G. V. Cormack and R. N. S. Horspool. Data compression using dynamic Markov modelling. The Computer Journal, 30(6), 541-550, 1997.

[12] T. J. Tjalkens, P. A. Volf and F. M. Willems. A context-tree weighting method for text generating sources. In Data Compression Conference. DCC'97. Proceedings (p. 472). IEEE, 1997.

[13] P. A. J. Volf. Weighting techniques in data compression: Theory and algorithms. Technische Universiteit Eindhoven, 2002.

[14] C. M. Bishop. Neural networks for pattern recognition. Oxford university press, 1995.

[15] S. R. Garner. Weka: The waikato environment for knowledge analysis. In Proceedings of the New Zealand computer science research students conference (pp. 57-64), 1995.

[16] W. J. Wilbur and K. Sirotkin. The automatic identification of stop words. Journal of information science, 18(1), 45-55, 1992.

[17] G. Salton and M. J. McGill. Introduction to modern information retrieval, 1983.

[18] R. Baeza-Yates, and B. Ribeiro-Neto. Modern information retrieval (Vol. 463). New York: ACM press, 1999.

[19] G. F. Cooper, and E. Herskovits. A Bayesian method for constructing Bayesian belief networks from databases. In Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence (pp. 86-94). Morgan Kaufmann Publishers Inc., 1991.

[20] N. Friedman, D. Geiger and M. Goldszmidt. Bayesian network classifiers. Machine learning, 29(2-3), 131-163, 1997.

[21] S. I. Amariand S. Wu. Improving support vector machine classifiers by modifying kernel functions. Neural Networks, 12(6), 783-789, 1999.

[22] S. Maji, A. C. Berg and J. Malik. Classification using intersection kernel support vector machines is efficient. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (pp. 1-8). IEEE.

[23] B. stn, W. J. Melssen and L. M. C. Buydens. Visualisation and interpretation of support vector regression models. Analytica chimica acta, 595(1), 299-309, 2007.

[24] B. H. Cho, H. Yu, J. Lee, Y. J. Chee, I. Y. Kim and S. I. Kim. Nonlinear support vector machine visualization for risk factor analysis using nomograms and localized radial basis function kernels. Information Technology in Biomedicine, IEEE Transactions on, 12(2), 247-256, 2008.

[25] J. R. Quinlan. C4. 5: programs for machine learning (Vol. 1). Morgan kaufmann, 1993.

[26] L. Breiman. Random forests. Machine learning, 45(1), 5-32, 2001.

[27] N. Ide and J. Vronis. Introduction to the special issue on word sense disambiguation: the state of the art. Computational linguistics, 24(1), 2-40, 1998.

[28] R. Navigli. Word sense disambiguation: A survey. ACM Computing Surveys (CSUR), 41(2), 10, 2009.

[29] M. Bates and R. M. Weischedel. Challenges in natural language processing. Cambridge University Press, 2006.

[30] T. G. Dietterich, R. H. Lathrop and T. Lozano-Prez. Solving the multiple instance problem with axis-parallel rectangles. Artificial intelligence, 89(1), 31-71, 1997.

[31] O. Maron and T. Lozano-Prez. A framework for multiple-instance learning. Advances in neural information processing systems, 570-576, 1998.

[32] Y. Zhou, Z. Jorgensenand M. Inge. Combating good word attacks on statistical spam filters with multiple instance learning. In Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on (Vol. 2, pp. 298-305). IEEE.