Automatic Moderation of Comments in a Large On-line Journalistic Environment

Adriano Veloso, Wagner Meira, Tiago Macambira, Dorgival Guedes and Hélio Almeida
Computer Science Department
Federal University of Minas Gerais
Belo Horizonte, Brazil
{adrianov,meira,tmacam,dorgival,helio}@dcc.ufmg.br

Abstract

On-line journalistic sites publish several news and stories every day. Readers of these sites may comment a story, and, as a consequence, a single story might receive thousands of comments. The quality of these comments may vary a lot, from spams and trolls to truly useful information. Separating good from bad comments is an important task, and is the primary goal of comment moderation. Moderators usually classify and score the comments, promoting high quality ones and, likewise, discouraging low quality ones. However, moderators usually face a very large number of comments, and thus, moderation may require a huge amount of time. In this paper we address the problem of automatic moderation of comments in a large journalistic Web site. Participants of the site may engage in discussions and interact with each other (i.e., friends, fans, enemies etc.), constituting a large social network. We propose a data mining technique which combines underlying patterns that are implicit in the content of the comments with patterns hidden in the social network, and then uses the result for classification and scoring purposes. We evaluate our proposed technique using a real collection of comments collected from the Slashdot forum. The proposed technique is effective, outperforming traditional approaches, such as decision trees and SVMs, in terms of accuracy. Further, the technique shows to be extremely fast, being able to moderate hundreds of comments per minute.

Keywords

Text Categorization, Social Network Analysis, Filtering

1. Introduction

Moderation is the ability to give feedback on the quality of a certain content. Several types of content may be moderated, such as articles, comments, wikis, blogs, and media files. Different methods of moderation may be applied, for instance, content may be classified (e.g., spam, informative etc.) or scored (e.g., rated in a $0-5$ scale). Moderation is extremely important in mass communication systems, sorting good from bad content and helping readers to find useful information.

Moderation is a task that is historically performed by humans, mostly because of subjective criteria that are applied while evaluating the content. The most common strategy is to assign a content to a moderator, who analyzes it and then express her/his opinion about its quality. However, moderators usually face a large amount of content to be analyzed, and, therefore, the moderation task consumes a huge amount of time (sometimes prohibitive). For instance, assume a large journalistic Web site, specifically the Slashdot forum[1]. Slashdot publishes several stories every day, and its readers have the ability to post comments to specific stories. Comments may be very informative and useful. On the other hand, users may have divergent opinions, start a discussion and change the subject of the original story. Thus, comments must be analyzed and moderated.

More than a forum for publishing stories, Slashdot constitutes a large social network, where users may interact with each other. Depending on the comments that were posted by a certain user, she/he may acquire fans, friends, and enemies throughout her/his existence as a participant of Slashdot. This interaction among users may result in communities and groups of users that have similar opinions (i.e., friends and fans) or not (i.e., enemies).

Each published story may receive thousands of comments, resulting in a huge amount of content to be moderated. The immediate alternative is to increase the number of moderators. Still, if we consider the total time that is spent by all moderators, we reach an unwanted work of extraordinary magnitude. What is needed is an automatic technique, in order to reduce the time spent in moderating comments. However, developing such a technique is challenging, since it involves modeling a large hypothesis space, where several pieces of evidence must be combined in order to classify/score a comment. In this paper we propose and evaluate such a technique. We consider the moderation task essentially as a classification problem, and we propose a classifier that finds patterns hidden in the comments as well as in the structure of the social network. These patterns work as different pieces of evidence, that are combined in order to classify and score comments.

We evaluate our proposed technique using a total of 301,278 manually moderated comments, which were collected from the Slashdot forum during two months. We compared the proposed technique against traditional ones, such as C4.5 decision trees [17] and SVMs [6]. We observed that the proposed technique is very effective for scoring comments, reaching more than 96% of accuracy. Classifying comments seems to be a more complex task, and the proposed technique achieves almost 67% of accuracy. Further, the proposed technique is very fast, being able to classify and score hundreds of comments per minute.

[1] http://slashdot.org

The rest of the paper is organized as follows. In the next section we present related work. In Section 3 we explain the moderation mechanism adopted by Slashdot, and we describe the corpus of the comments used in our evaluation. In Section 4 we present our technique for automatic moderation of comments. Extensive experimental results are presented in Section 5. Finally, we present our conclusions in Section 6.

2. Related Work

Comment moderation shares some similarities with opinion retrieval, which is concerned with recognizing, classifying, and understanding opinionated text. An interesting approach to opinion retrieval was presented in [14]. The proposed approach consists of identifying different components (i.e., opinion expression and post quality) that may be used to retrieve the correct opinion. Another interesting approach was presented in [7], in which an automatic method for obtaining opinion-bearing words was extended to identify opinion-bearing sentences. Classifying a comment is essentially predicting the opinion of the moderator, and thus, we believe that our proposed classifiers can also be used for opinion retrieval.

Our work can also be compared with sentiment analysis, which deals with the extraction and characterization of emotions, opinions and other non-factive aspects of text. A key task is to identify positive and negative opinions towards a given topic. Mood classification is an important subject of sentiment analysis. In [15] the authors proposed a model to estimate the state-of-mind of bloggers from the text written by them. In [2] a time-series approach was used to study trends and temporal aspects that may influence mood.

A large-scale analysis of comments is presented in [16]. Important facts were verified, including that comments constitutes a substantial part of the blogosphere. Thus, comments are a valuable source of information, that is often neglected. Thus, there is a lack of study and analysis of comments.

Machine learning techniques for text categorization have been widely studied [18, 6, 21]. A collaborative filtering technique that suggests tags for blog posts was proposed in [13]. The proposed technique showed good accuracy and can simplify the tagging process. In [9] the authors employed SVMs to two important tasks: blog identification and splog detection, being successful in both applications. A content analysis technique [10] was used to detect spam Web pages. Finally, a machine learning technique for splog detection was proposed in [10]. In general, approaches for text categorization (such as the aforementioned ones) are evaluated based only on metrics that quantify classification accuracy. Our experimental evaluation also takes into account the computational performance of the proposed classifiers.

3. Corpus

In this section we present the comment moderation strategy used in the Slashdot forum. Then we describe the comment corpus we studied and how it was built.

3.1 Slashdot Moderation Strategy

Figure 1 shows a story that was published in the Slashdot forum. Some comments about the story are also shown (story and comments were randomly selected). Each comment has an author and a title. The score and category assigned to the comment are next to its title. Comments can be scored in a scale that vary from -1 to 5 and can be classified into 8 descriptive categories (Insightful, Interesting, Informative, Funny, Flamebait, Troll, Off Topic, and Redundant). Classification and scoring of comments are handled by Slashdot moderators. Instead of relying on a small and fixed group of moderators, Slashdot delegates this task to its participants. Any registered user is eligible to become a moderator. Some filters apply, though, as the selection of moderators is not completely random. Factors such as frequency of access, membership time and *karma*, among others, are taken into consideration in this selection.

Karma is a numeric representation of how previous interactions of a user with the site (previous comments, moderation, meta-moderation and story submission) were seen by the community. Every time a comment is moderated up or down, the karma of its author is updated accordingly. Besides being used to select moderators, the karma of a user also influences the default score assigned to her/his comments: comments posted by logged and good karmed users start rated as 1 or 2, whereas comments posted by anonymous and low karmed users start rated as 0 and -1. The remaining points come from the category assigned to the comment (a positive classification will increase the score, and a negative classification will decrease the score).

3.2 Corpus Characterization

We collected a set of 472 stories that were published in the Slashdot forum during two months. In response to these stories, 301,278 comments were posted to the forum. During these two months 42,528 distinct users posted at least one comment to the forum. Figure 2 shows the distribution of comments per user. Around 500 users posted more than 60 comments during these two months. These users are long time regular readers of the forum. Almost 40,000 users posted less than 10 comments during two months.

Figure 3 shows the frequency distribution of the words that appear in the comments. As expected, the number of comments in which a word occurs follows a power-law distribution, with a small number of words appearing in a high number of comments, and a long tail of words appearing in few comments.

Figure 4 shows the number of fans, foes, freaks and friends per user. From all users, 7,671 have at least one fan, 4,720 have at least one foe, 6,372 have at least one freak, and 6,825 have at least one friend. Some participants are very popular, having more than 100 friends and/or fans.

Figure 5 shows the distribution of comments per score. Score 0, 1, and 2 concentrate the vast majority of comments (approximately 90%). Only 10% of the comments received a score higher than 2.

Figure 6 shows the distribution of comments per category. From the 301,278 total number of comments, only 51,494 were classified (the remaining 249,784 comments were moderated just by scoring). The majority of the comments received a positive classification (i.e., insightful, interesting, informative or funny).

Table 1 shows the correlation between scores (columns) and categories (lines). As we can see, there is a strong correlation between positive classifications (i.e., Insightful, Informative and Interesting) and higher score values. Similarly, there is a strong correlation between negative classifications (i.e., Flamebait, Off Topic, Redundant and Troll) and lower score values. These correlations are expected, because posi-

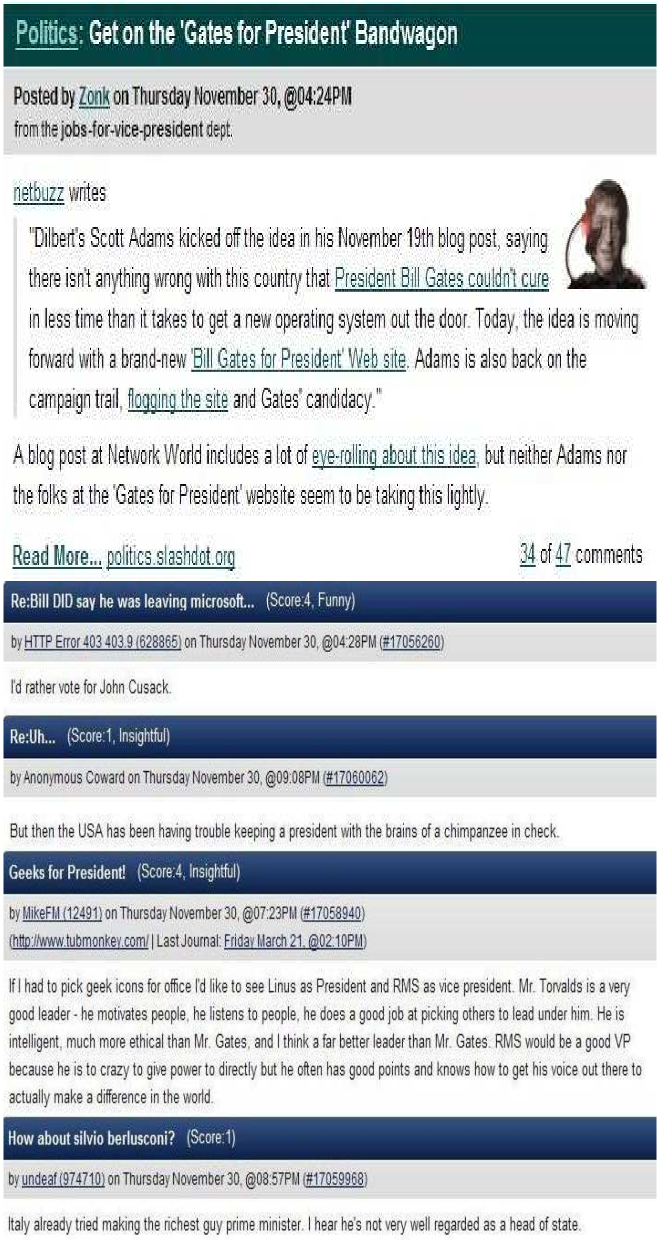

Fig. 1: *A Published Story and Its Comments.*

tive classifications contribute to increase the score, while negative classifications contribute to decrease the score.

It is also interesting to analyze how frequent social relationships among users are. Among the users we analyzed, 90% have fans, but just 35% have foes. Further, 77% have freaks and 60% have friends. Table 2 shows the percentage of users that have (or not) at least one relationship per category, where we can see that just 5% of the users have no relationships, while almost 30% present all kinds of relationships. This information supports our choice of using social relationships as an evidence for sake of scoring and classification and will be key to the success of our approach.

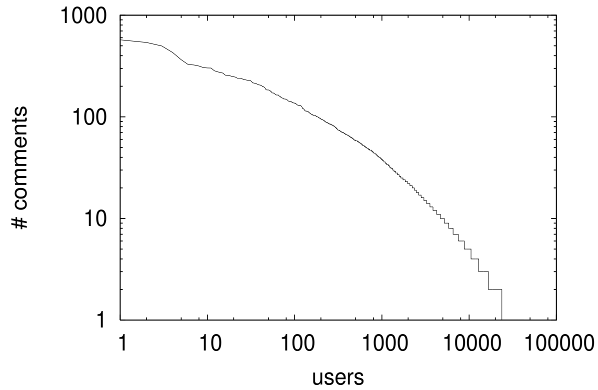

Fig. 2: *Distribution of Comments per User.*

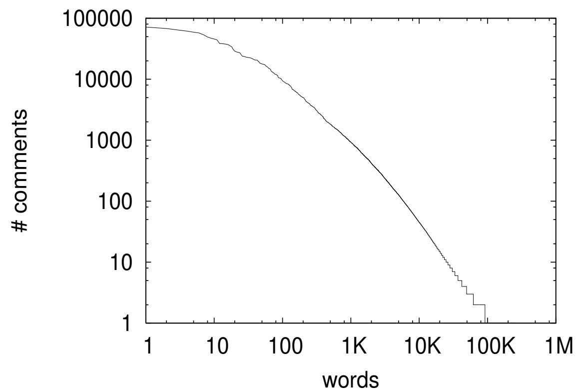

Fig. 3: *Frequency Distribution of Words.*

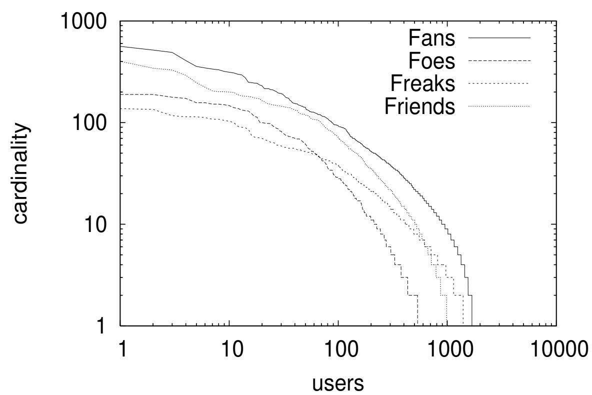

Fig. 4: *Distribution of Fans, Foes, Freaks and Friends.*

4. Automatic Moderation of Comments

In this section we present our technique for automatic moderation of comments. We consider the moderation task essentially as a classification problem, and thus, we start by presenting our classification technique.

	-1	0	1	2	3	4	5
Flamebait	23114	11260	9495	1777	0	0	0
Funny	79	2590	24129	48240	60029	31419	74361
Informative	24	935	20302	48200	76661	32491	59152
Insightful	41	1909	29879	72606	122660	54451	103722
Interesting	25	1135	21529	55319	87580	35178	55998
Off Topic	42658	13414	10529	1572	0	0	0
Redundant	10542	8924	7183	1492	0	0	0
Troll	39455	14686	11194	2146	0	0	0

Table 1: *Correlation between Scores and Categories.*

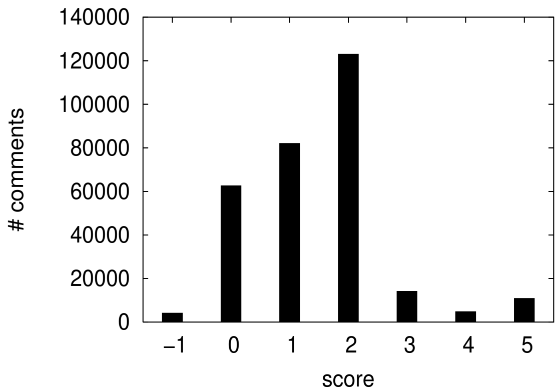

Fig. 5: *Distribution of Scores.*

Fans	Foes	Freaks	Friends	Percentage
				5.6%
			•	0.7%
		•		2.1%
		•	•	0.4%
	•			0.05%
	•		•	0.10%
	•	•		0.10%
	•	•	•	0.16%
•				8.1%
•			•	5.3%
•		•		20.7%
•		•	•	21.8%
•	•			0.16%
•	•		•	2.8%
•	•	•		2.2%
•	•	•	•	29.5%

Table 2: *Social Relationships.*

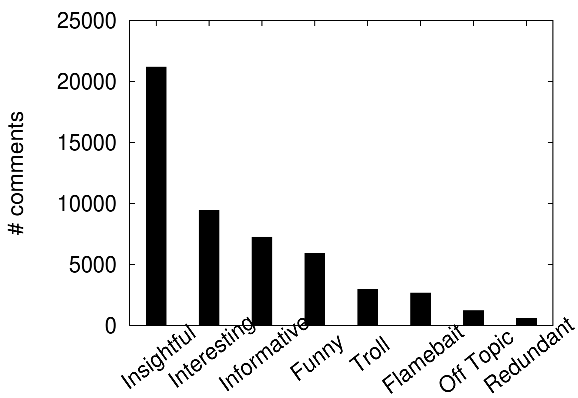

Fig. 6: *Distribution of Categories.*

4.1 Associative Classification

Associative classification [12] is a technique that makes use of association rules [1] to construct a classification model from the training data. Some associative classifiers have been proposed [12, 3, 11, 5], and they have shown to be more accurate than KNNs and SVMs [19]. Recently, it has been formally shown that an associative classifier is more accurate than the corresponding decision tree [20]. In the following we define some key concepts, and then we introduce our associative classifiers.

Definition 1. [Comments] Let $\mathcal{C}$ denote the set of m comments $\{c_1, c_2, \ldots, c_m\}$, where each comment c_i is composed of a category k and a score s, along with a set of features (i.e., a feature set or feature vector). There are 7 different types of features:

- Author: It is the identifier of the user who posted the comment.
- Title: It is the set of words/symbols appearing in the title of the comment.
- Body: It is the set of words/symbols appearing in the body of the comment.
- Fans: It is the set of fans that were acquired by the author of the comment.
- Foes: It is the set of foes that were acquired by the author of the comment.
- Freaks: It is the set of freaks that were acquired by the author of the comment.
- Friends: It is the set of friends that were acquired by the author of the comment.

Let $\mathcal{F}$ denote the set of all unique features. A feature set is simply a non-empty subset of $\mathcal{F}$, and it may contain different types of features. We use $\mid \mathcal{X} \mid$ to denote the cardinality (or size) of the feature set $\mathcal{X}$. The support (or frequency) of the feature set $\mathcal{X}$ is the fraction of comments in the training data that contain $\mathcal{X}$ as a subset, given by: $\sigma(\mathcal{X}) = \frac{|\{c_i \in \mathcal{C} | \mathcal{X} \subseteq c_i\}|}{m}$. The feature set $\mathcal{X}$ is frequent if $\sigma(\mathcal{X}) \geq \sigma_{min}$, where σ_{min} is a user-specified minimum support threshold which is used to ensure that the feature set appears in a large enough number of comments.

Definition 2. [Association Rules] The rule $\mathcal{X} \xrightarrow{\theta} k_i$ associates a feature set $\mathcal{X}$ to a category k_i. The support of the rule is given by $\sigma(\mathcal{X} \rightarrow k_i) = \sigma(\mathcal{X} \cup \{k_i\})$. The strength of the rule is given in terms of its confidence, defined as the conditional probability of the consequent when the antecedent is known, given by: $\theta = \frac{\sigma(\mathcal{X} \rightarrow k_i)}{\sigma(\mathcal{X})}$. The rule $\mathcal{X} \xrightarrow{\theta} k_i$ is strong if $\theta \geq \theta_{min}$, where θ_{min} is a user-specified minimum confidence threshold. The size of $\mathcal{X} \xrightarrow{\theta} k_i$ is given by $| \mathcal{X} |$+1.

Generation of association rules is a two-step process. In the first step all frequent feature sets are found. In the second step all strong rules are generated from the frequent feature sets. The first step involves accessing the collection of comments in order to calculate the support of feature sets, and thus, is the most complex one. There are several efficient algorithms for mining association rules [22, 1].

The classification model is composed of all strong rules. Each rule $\mathcal{X} \rightarrow k_i$ is considered as a vote of $\mathcal{X}$ for category k_i. Some votes weight more heavily than others, depending on the quality of the rule that is issuing the vote. The weight of each vote represents how many points the vote is worth, and it is calculated based on how frequent and strong the corresponding rule is, given by: $\sigma(\mathcal{X} \rightarrow k_i) \times \theta(\mathcal{X} \rightarrow k_i)$. All votes are summed, and in the end the total number of points of each category is divided by the number of votes for this category. For example, category k_i, which receives n votes, will make $\frac{\sum_{j=1}^{n}(\sigma(\mathcal{X}_| \rightarrow k_i) \times \theta(\mathcal{X}_| \rightarrow k_i))}{n}$ points. The predicted category is the one that acquired the higher number of points.

Table 3 shows a set of five moderated comments from which a set of association rules is induced. Suppose we set σ_{min}=40% and θ_{min}=95%. In this case the classification model includes the following rules:

- if title contains {Re:} and body contains {stupid} then category=Troll (σ=40%, θ=100%)
- if title contains {Automatic, Moderation} then category=Interesting (σ=40%, θ=100%)
- if fans contain {julia} and foes contain {andrew} then score=4 (σ=40%, θ=100%)
- if author is oliver then score=4 (σ=40%, θ=100%)

The most important concern in this process is to select the rules that will compose the model. There are two approaches for model induction. In the eager (traditional) approach, which is the one used by most of the classifiers [12, 3], a single classification model is induced from the training data. In the lazy approach, which is the one used by our proposed technique, a specific model is induced for each comment that is classified.

4.1.1 Eager Classifier

The eager classifier (EAC) induces a single model, and uses this model to classify all comments. That is, the model is induced before the comments that will be classified are known. In this case, the challenge for the classifier is in anticipating all the different directions in which it should attempt to generalize its training examples (i.e., what rules must be generated). In practice, the induced model is good on average, but fails in specific cases (i.e., comments with rare words or posted by not so frequent users).

4.1.2 Lazy Classifier

The lazy classifier (LAC) induces a specific model for each comment to be classified. That is, the induction process (i.e., rule generation) is delayed until a comment is given for classification. In this case, the classifier generalizes its training examples exactly as needed to cover the comment to be classified, generally resulting in more accurate predictions. The drawback is that the classifier has to induce several models, one for each comment to be classified, demanding more computational resources than its eager countpart.

In the next section we evaluate and compare eager and lazy associative classifiers.

5. Experimental Evaluation

In this section we present the experimental results for the evaluation of the proposed approaches, EAC and LAC, in terms of both classification effectiveness and computational efficiency. Parameters that were varied include σ_{min}, θ_{min}, and rule size. We divided our evaluation into two distinct tasks: scoring and classifying comments. Our proposed techniques are compared against SVMs [4] and C4.5 decision trees [8], using the corpus described in Section 3. In all experiments we used 10-fold cross-validation and the final results of each experiment represent the average of ten runs. All experiments were performed on a Linux-based PC with a Intel Pentium III 1.5 GHz processor and 2.0 GBytes RAM.

5.1 Comment Scoring

The first task is to estimate an integer value, the score of the comment. The metric used for quantifying classification effectiveness is accuracy, which is calculated based on how far (or how close) the estimated score is from the actual score. That is, the accuracy of each prediction is calculated based on the proximity of the predicted and actual scores, and in the end, the final accuracy is an average of the accuracies of all predictions.

Figure 7 shows the accuracy achieved by the proposed approaches with different values of σ_{min}. We varied σ_{min} from 0.01% to 2%, and observed the corresponding accuracy. As we can see, best accuracy is obtained when σ_{min} is between 0.05% and 0.1%. Lower values of σ_{min} implies in the generation of a large number of rules, but some of these rules may overfit the training data, hurting accuracy. On the other hand, higher values of σ_{min} implies in the generation of few rules, increasing the chance of a scenario in which there is no rule to be applied to score a specific comment.

Figure 8 shows the accuracy achieved by the proposed approaches with different values of θ_{min}. We varied σ_{min} from 80% to 95%, and observed the corresponding accuracy. As expected, best accuracy is obtained when higher values of θ_{min} are employed, since in this case more accurate rules are generated.

Table 4 shows accuracy numbers when different rules sizes are allowed. Shorter rules are simpler and more general, while longer rules are more complex and specific. As can be seem, very simple rules (or very complex ones) are not the best choice. Very general rules may underfit the training data, while very complex rules may overfit the training data. Thus,

Id	Category	Score	Author	Title	Body	Fans	Foes	Friends
1	Troll	0	andrew	Re: Uh...	Dont be stupid!	julia	oliver	ana
2	Interesting	5	jack	Automatic Moderation	Great idea.	oliver	andrew	julia
3	Funny	4	oliver	Uh...	chimpanzees are stupid:)	julia	andrew	
4	Troll	-1	anonymous	Re: to oliver	Stupid comment.			
5	Interesting	4	oliver	Re: Automatic Moderation	Someone is trying it.	julia	andrew	

Table 3: *An Illustrative Set of Comments used as an Example.*

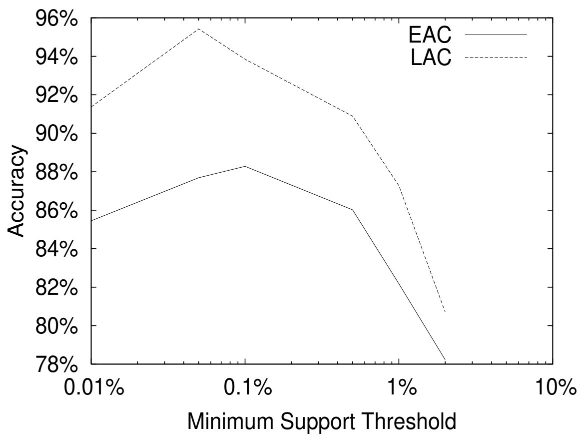

Fig. 7: *Accuracy with σ_{min} (θ_{min}=95%).*

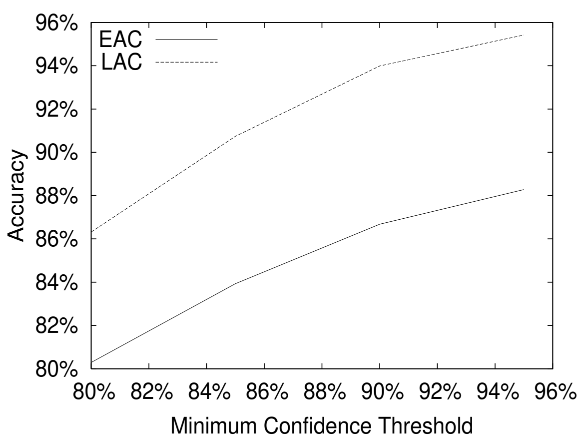

Fig. 8: *Accuracy with θ_{min} (σ_{min}=0.05%).*

the choice of the proper rule size is a trade-off between underfitting and overfitting.

Average Rule Size	EAC	LAC
1	88.29%	88.08%
2	86.71%	91.97%
3	87.51%	93.72%
4	88.28%	95.42%
5	86.95%	94.69%
6	84.83%	94.22%

Table 4: *Accuracy with Rule Size.*

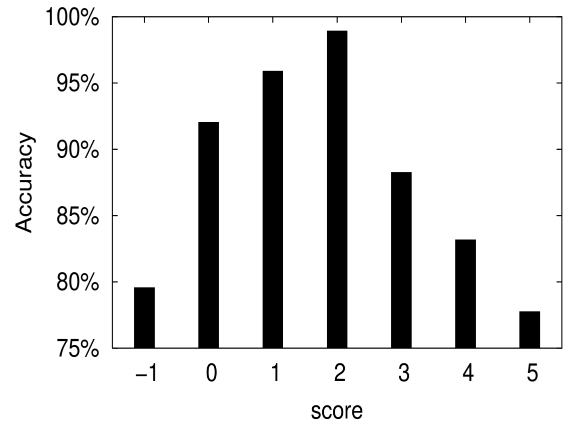

Fig. 9: *Accuracy per Score.*

5.2 Comment Classification

The second task is to predict the correct category of a given comment. We quantify the classification effectiveness through the conventional accuracy, precision and recall. Precision p is defined as the proportion of correctly classified comments in the set of all comments. Recall r is defined as the proportion of correctly classified comments out of all the comments having the target category. F_1 is a combination of precision and recall, defined as the harmonic mean $\frac{2pr}{p+r}$.

Technique	Accuracy	Exec. Time
LAC	66.30%	11,289 secs
EAC	45.23%	8,785 secs
SVM	59.09%	17,917 secs
C4.5	40.82%	8,219 secs

Table 5: *Accuracy and Execution Time (σ_{min}=0.1% and θ_{min}=95%).*

Classifying comments is a very hard task, because different moderators may have different opinions about similar comments. Table 5 shows a comparison of the different classification techniques. Similarly to the first task, LAC classifier is the best performer in terms of accuracy. C4.5 decision tree classifier is the faster one, but achieves just 40.82% of accuracy.

Table 6 shows precision, recall and F_1 numbers obtained by the LAC classifier, so that we can see how accuracy is distributed among the categories. As can be seem, in general higher precision, recall, and F_1 numbers are associated with more frequent categories.

Category	Precision	Recall	F_1
Insightful	71.67%	86.29%	0.78
Interesting	68.11%	65.91%	0.67
Informative	58.40%	51.64%	0.55
Funny	77.81%	74.61%	0.76
Flamebait	87.33%	29.17%	0.44
Troll	64.67%	62.81%	0.64
Off Topic	21.28%	41.94%	0.28
Redundant	46.81%	62.83%	0.54

Table 6: *Precision, Recall and F_1 for LAC classifier (σ_{min}=0.1% and θ_{min}=95%).*

6. Conclusions and Future Work

In this paper we deal with automatic moderation of comments in a journalistic social environment. More specifically, the moderation task consists in score and classify comments posted to a forum. We considered automatic moderation essentially as a classification problem, and we proposed approaches to address this task. The proposed techniques are based on associative classification. We evaluated two different associative classifiers, using a collection of comments posted to the Slashdot forum during two months. Participants of this forum can interact with each other, constituting a large social network. The classification model uses patterns hidden in the content of the comment and also patterns hidden in the structure of the social network. The eager classifier (EAC) builds a single classification model that is used to moderate all comments. The lazy classifier (LAC) builds a specific classification model for each comment to be moderated. The lazy classifier shows to be more accurate and faster than a SVM classifier.

Classifying comments seems to be much more hard than scoring them. This is mainly because classifying comments is a very subjective task, that only depends on the personal opinion of the moderator, and different moderators may have different opinions about similar comments. Scoring comments, on the other hand, is a task that can be well modeled, since there are some predefined (and fixed) rules imposed by Slashdot.

As future work we will explore other application scenarios such as opinion retrieval, spam detection and sentiment analysis.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. of the International Conference on Management of Data (SIGMOD03)*, pages 207–216. ACM Press, 1993.

[2] K. Balog and M. de Rijke. Decomposing bloggers' moods: Towards a time series analysis of moods in the blogosphere. In *Proc. of the 3rd Annual Workshop on the Weblogging Ecosystem (WWE06)*, 2006.

[3] E. Baralis and S. Chiusano. Essential classification rule sets. *ACM Transactions on Database Systems*, 29(4):635–674, 2004.

[4] C. Chang and C. Lin. LIBSVM: A library for support vector machines. *Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm*, 2001.

[5] G. Dong, X. Zhang, L. Wong, and J. Li. CAEP: Classification by aggregating emerging patterns. In *Proc. Intl. Conf. on Discovery Science*, pages 30–49, 1999.

[6] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proc. of the 10th European Conference on Machine Learning (ECML98)*, pages 137–142, 1998.

[7] S. Kim and E. Hovy. Automatic detection of opinion bearing words and sentences. In *Proc. of the 2nd International Joint Conference on Natural Language Processing (IJCNLP05)*, 2006.

[8] R. Kohavi, D. Sommerfield, and J. Dougherty. Data mining using MLC++: A machine learning library in C++. In *Tools with Artificial Intelligence*, pages 234–245, 1996.

[9] P. Kolari, T. Finin, and A. Joshi. Svms for the blogosphere: Blog identification and splog detection. In *AAAI Spring Symposium on Computational Approaches to Analysing Webogs*, 2006.

[10] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi. Detecting spam blogs: A machine learning approach. In *Proc. of the 21st National Conference on Artificial Inteligence (AAAI06)*, 2006.

[11] W. Li, J. Han, and J. Pei. Accurate and efficient classification based on multiple class-association rules. In *Proc. of the International Conference on Data Mining (ICDM01)*, pages 369–376, 2001.

[12] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proc. of the 4th International Conference on Knowledge Discovery and Data Mining (SIGKDD98)*, pages 80–86, 1998.

[13] G. Mishne. Autotag: A collaborative approach to automated tag assignment for weblog posts. In *Proc. of the 15th International World Wide Web Conference (WWW06)*, pages 953–954, 2006.

[14] G. Mishne. Multiple ranking strategies for opinion retrieval in blogs. In *Proc. of the Text Retrieval Conference (TREC06)*, 2006.

[15] G. Mishne and M. de Rijke. Capturing global mood levels using blog posts. In *AAAI Spring Symposium on Computational Approaches to Analysing Webogs*, 2006.

[16] G. Mishne and N. Glance. Leave a reply: An analysis of weblog comments. In *Proc. of the 3rd Annual Workshop on the Weblogging Ecosystem (WWE06)*, 2006.

[17] J. Quinlan. C4.5: Programs for machine learning. *Morgan Kaufmann*, 1993.

[18] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surveys*, 34(1):1–47, 2002.

[19] A. Veloso, W. Meira Jr., M. Cristo, M. Gonçaves, and M. Zaki. Multi-evidence, multi-criteria, lazy associative document classification. In *Proc. of the Conference on Information and Knowledge Management (CIKM06)*, 2006.

[20] A. Veloso, W. Meira Jr., and M. Zaki. Lazy associative classification. In *Proc. of the International Conference on Data Mining (ICDM06)*, 2006.

[21] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2):69–90, 1999.

[22] M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li.

New algorithms for fast discovery of association rules. In *Proc. of the International Conference on Knowledge Discovery and Data Mining (SIGKDD97)*, pages 283–290, 1997.