Learning to Block Undesired Comments in the Blogosphere

Tiago A. Almeida, Túlio C. Alberto
Department of Computer Science
Federal University of São Carlos – UFSCar
18052-780, Sorocaba, São Paulo, Brazil
talmeida@ufscar.br, tuliocasagrande@gmail.com

Abstract—A significant amount of the available information on web sites comes from the interaction with users, such as news sites and blogs, where readers can post comments and sometimes develop habits of frequenting them. Some blogs specialize in certain subjects, receive the users credibility and become references in the field. However, the ease of inserting content through text comments makes room for unwanted messages, which affect the user experience, reduce the quality of the information provided by the websites and indirectly causing personal and economic losses. Given this scenario, this paper presents a comprehensive analysis of machine learning techniques applied to automatically detect undesired comments posted on blogs. Experiments carried out with a real and public database indicate that support vector machines and logistic regression, trained with both attributes extracted from the text messages and metadata from the posts, are promising in the task of filtering unwanted comments.

Keywords—blog comment spam; undesired messages; classification.

I. INTRODUCTION

The interactivity with users is a strong characteristic of the web 2.0. Nowadays, many websites with user generated content support text comments, such as news and blogs. This interaction can be used both as an indicator of the page’s content, as to assist the author to correct pieces of information that may be incorrect or incomplete [1]. On the other hand, the facility to insert content makes room for undesired messages that disturb readers and reduce the quality of the information provided on the websites.

Undesired comments cause confusion in search engines and can dramatically reduce the pagerank of such websites. As a consequence, blog comment spam can cause injury to the authors by the reduction of the traffic of readers. Traffic statistics are the key point for those who offer ads on websites, thus a reduction in the number of accesses implies the decreasing interest of potential advertisers.

For many authors in the blogosphere, the discussion around a post is an inseparable part of its content and a motivation to keep publishing. According to Mishne and Glance [1], the text comments represent about 30% of the blogs total content and, therefore, it is very important to effectively automatically filter such kind of spam in order to ensure the quality of information to the readers.

One of the few currently available services to filter blog comment spam is the Akismet1. According to information provided on its web site, more than 82 billion comment spam have been detected since it came in operation and, moreover, more than 1.5 million messages are blocked in a single day. The volume of legitimate comments on average is less than 15% of the total published messages. Furthermore, unlike the email spam, most comment spam are not sent by bots, but by people who pose as legitimate users and attempt to post messages with links and advertisements2.

Given this scenario, this paper presents a comprehensive performance evaluation of several well-known machine learning techniques that can be applied to automatically filter such unwanted messages. Our main goal is to find promising methods that can be used to assist the detection of undesired text comments posted on blogs and news websites. Compared to works available in the literature, this paper offers the following main contributions:

1) empirical results achieved by several classifiers including some that have not been evaluated so far, such as boosting, tree-based techniques, logistic regression and support vector machines with different kernel functions;
2) performance evaluation of different feature vectors extracted from comments available in a real and public database; and
3) comparison of the results attained in this work with those achieved by other papers in the literature;

This paper is organized as follows: in Section II we briefly describe the related papers available in the literature. Section III presents the basic concepts of the methods evaluated in this paper. In Section IV we present the database and settings that we used in the experiments. Section V presents the main results. Finally, Section VI describes the conclusions and guidelines for future work.

II. RELATED WORK

Filtering blog comment spam is a relatively recent problem and, consequently, there are still few papers available in the literature that offer significant contributions. In one of the earliest work, Mishne et al. [2] proposed a technique

1Akismet – Available at: https://akismet.com/.
2How we stop comment and trackback spam. Available at: https://akismet.com/how/.
using the KL-divergence method to compute the difference between the language model of the original post and its comments. Since the post provides context to the comment, it is possible to use it to identify undesired comments. This approach creates statistical models of language by means of probability distributions taking into account the occurrence of strings in the text. It aims to verify the presence of links that may take to undesired pages. The authors also created the first public blog spam database that has been used in later studies [1], [3], [4].

Mishne and Glance [1] studied the relations between the post and its comments. The authors estimated that the volume of comments in blogs corresponds to about 30% of the post total content. They also studied the relations between the comments popularity and patterns, comparing the number of views and the number of comments. The authors concluded that the comments are of vital importance to the blog, since search engines also index the comments and, therefore, the more relevant its content, the higher the pagerank of the site.

Cormack et al. [5] evaluated the performance achieved by methods commonly used in email spam filtering applied to classify short text messages, such as SMS and blog comments. The authors concluded that such messages have an insufficient number of features to train the most known approaches, and therefore, it is necessary to employ additional techniques to expand the space of attributes.

Romero et al. [3] performed a comparative study of four machine learning techniques in blog comment spam filtering, by extracting attributes of the message content. Among the evaluated classifiers: Naïve Bayes, k-nearest neighbors, artificial neural networks and support vector machines (SMO), the latter achieved the best result.

Shin et al. [6] proposed a model trained with features extracted from the spammers’ origin information, commenting activities, comment text and URLs. Originally designed to forum messages filtering, this approach has similarities with blogs, as insufficient amount of information in the messages. The results indicate that it is a promising method, however the filter was developed for a single website and may not be effective when used more generally.

Bhattarai and Dasgupta [4] presented a self-supervised approach based on decision trees. According to the authors, the method still needs to be improved but is flexible enough to be used in a mixed way, in which part of the supervision is done by humans and part by the algorithm.

Kantchelian et al. [7] defined spam as any message with uninformative content. Then, they created a metric called content complexity to measure the level of information present in each comment. To enhance the metric, the authors grouped the comments that had the same user, the same IP address, the same links and other data. Using a private database, the authors evaluated a classifier based on logistic regression and the proposed metric.

Wang et al. [8] studied a problem which they called diversionary comments. According to the authors, many political blogs suffer from such comments, which aim to twist the bloggers’ intention and divert the topic to another one. The authors considered the problem of diversionary comments as still novel in the literature and applied well-known methods as KL-divergence to measure the similarity between the contents of the post and its comments.

Chu et al. [9] proposed a technique to automatically detect blog bots – automated programs that fill forms and post comments. Unlike conventional detection methods that require direct user participation, such as CAPTCHA, the authors presented an approach that uses behavioral biometrics, such as the way to use mouse and keyboard, to distinguish between a real person and a blog bot.

### III. Methods

This section presents concepts regarding the following well-known methods that we evaluated in this paper: naïve Bayes classifier, support vector machines, methods based on trees, such as decision trees and random forest, k-nearest neighbor, adaptive boosting of trees, logistic regression and a rule-based classifier. Such methods were chosen because most of them have been presented as the best machine learning and data mining techniques currently available [10].

#### A. Naïve Bayes

The Naïve Bayes (NB) classifier is still the most employed in spam filtering because of its simplicity and high performance [11]. From Bayes’ theorem and the theorem of the total probability, the probability for a message with vector \( \vec{x} = (x_1, \ldots, x_n) \) belongs to a category \( c_i \in \{c_s, c_l\} \): 

\[
P(c_i|\vec{x}) = \frac{P(c_i).P(\vec{x}|c_i)}{P(\vec{x})}.
\]

Since the denominator does not depend on the category, NB classifies each message in the category that maximizes \( P(c_i).P(\vec{x}|c_i) \).

#### B. Support vector machines (SVM)

Support vector machines (SVM) [12] is a machine learning method that can be used for pattern classification, regression and others learning tasks [13], [14]. This method was conceptually implemented following the idea that input vectors are non-linearly mapped to a high dimension feature space. In this feature space is constructed a linear decision surface which separates the classes of the input patterns.

One of the main elements that the SVM uses to separate the patterns of distinct classes is a kernel function. Through it, the SVM constructs a decision surface nonlinear in the input space, but linear in the features space [13]. Table I presents the most popular SVM kernel functions, where \( \gamma \), \( r \) and \( d \) are parameters that must be set by the user.

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>( k(x_i, x_j) = x_i^T x_j )</td>
</tr>
<tr>
<td>RBF</td>
<td>( k(x_i, x_j) = \exp(-\frac{1}{2\gamma}|x_i - x_j|^2), \gamma &gt; 0 )</td>
</tr>
<tr>
<td>Polynomial</td>
<td>( k(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma &gt; 0 )</td>
</tr>
</tbody>
</table>
To assist the choice of the SVM parameters, Hsu et al. [15] recommend the employment of a grid search. For instance, considering the SVM with RBF kernel, in which is necessary to define the regularization parameter $C$ and $\gamma$, the authors suggest that the grid search could be used to evaluate exponential sequences: $C = 2^{-5}, 2^{-4}, 2^{-3}, \ldots, 2^{15}$ and $\gamma = 2^{-15}, 2^{-14}, \ldots, 2^{3}$.

C. Decision trees (C4.5)

The C4.5 [16] is one of the most classical decision tree algorithms and uses both categorical and continuous attributes. It uses a divide-and-conquer approach to increase the predictive ability of the decision trees. Thus, a problem is divided in several sub-problems, by creating sub-trees in the path between the root and the leaves of the decision tree.

D. Random forest

A random forest [17] is a combination of decision trees in which each tree depends on the values of a random vector sampled independently and equally distributed for all trees in the forest. In this method, after generating a large number of trees, each one votes for one class of the problem. Then, the class with more number of votes is chosen.

E. K-nearest neighbors (IBK)

The IBK is an instance-based learning algorithm (IBL) [18]. Such method, derived from the $k$-nearest neighbors (KNN) classifier, is a non-incremental algorithm and aims to keep a perfect consistency with the initial training set. On the other hand, the IBL algorithm is incremental and one of its goals is maximizing classification accuracy on new instances [18].

As well as in the KNN, in IBK, the classification generated for the sample $i$ is influenced by the outcome of the classification of its $k$-nearest neighbors, because similar samples often have similar classifications [18], [19].

F. Adaptive boosting (AdaBoost)

The adaptive boosting [20] is a boosting algorithm widely used in pattern classification problems. In general, as any boosting method, it makes a combination of classifiers. However, it has some properties that make it more practical and easier to implement than the boosting algorithms that preceded it. One of these properties is that it does not require any prior knowledge of the predictions achieved by weak classifiers. Instead, it adapts to the bad predictions and generates a weighted majority hypothesis in which the weight of the each prediction achieved by weak classifiers it is a function of its prediction.

G. Multinomial Logistic Regression

Multinomial logistic regression is a technique generally employed to predict the probability between different possible results of a dependent variable categorically distributed, given a set of independent variables. The term “multinomial” in its name is due to the union of categorically and multinomial distributions.

H. PART

PART [21] is a rule learner that is based on an algorithm for inferring rules by repeatedly generating partial decision trees with C4.5. A major property of this algorithm is that rule sets are learned one rule at a time, without any need for global optimization. Despite of avoiding this optimization present in other rule learners like RIPPER, or decision tree learners like C4.5, it achieves very good performance in terms of accuracy and efficiency. This method was evaluated in SMS spam filtering task by Gómez Hidalgo et al. [22].

IV. DATABASE AND EXPERIMENT SETTINGS

To give credibility to the found results and in order to make the experiments reproducible, all the tests were performed with the real, public and well-known BlogSpam Collection [2]. It is composed by 1,024 labeled comments extracted from 50 different blogs.

The dataset has different types of comment spam, some of which consist of a simple list of common keywords with links pointing to dangerous websites, while others are represented by sophisticated sentences. The messages are divided into classes as follows: 68% spam and 32% ham (legitimate comments).

Each message consists of posting information (metadata - in typewriter) and the text message itself (in italic), as the following example.

comment id="26" post="Posted by: <a target="_blank" title = "http://www.bestukcasinos.co.uk" href="mt-comments.cgi?__mode=red; id=23964" > UK Casinos</a> at January 25, 2004 05:53 PM" Thanks for the great blog!Best Regards,Marry—— ————The best online roulette right here.Play blackjack online today.Online slots gambling.Quality keno gaming action.Play online video poker today!

Given this scenario, we have performed three different experiments for each classifier:

1) with the attributes extracted from the text messages;
2) with the attributes extracted from the posting metadata; and
3) with all the attributes extracted from both the posting metadata and text messages.

To extract the terms (tokens) of the text messages, we have employed a simple tokenizer to get any sequence of characters separated by: blank, tab, newline, period, comma, semicolon or dash. In this way, we intend to keep other symbols that may help to classify the messages. In addition, any pre-processing was performed as stop words removal or stemming, since some research results indicate that such techniques tend to hurt the performance of the spam classifiers [23], [24].

A. Settings

To address the algorithms performance, all the experiments were performed using the traditional 10-fold cross-validation. Thus, we divided each simulation in 10 parts and calculated the arithmetic mean and standard deviation of the following well-known measures:

- **Accuracy (Acc)** – proportion of correctly classified samples;
- **Spam Caught (SC)** – proportion of correctly classified spam samples;
- **Blocked Ham (BH)** – proportion of incorrectly classified ham samples;
- **F-measure** – the harmonic mean between precision and recall [19];

In the following, we describe the main parameters used for each classifier evaluated in this work.

1) **SVM**: we implemented the SVM using the LIBSVM library [14]. We performed simulations with the linear, RBF and polynomial kernel functions and we used grid search for tuning the parameters. However, for the SVM with polynomial kernel, that have a large number of parameters, we performed the grid search only on the parameters C and γ. In this case, we set the default values of the LIBSVM for the remaining parameters.

We performed the grid search using the random subsampling validation with 80% of the samples for training and 20% for test. Then, we chosen the best parameters – those that achieved the highest F-measures, and we used them to perform the experiments with SVMs. Table II presents the parameters used in the simulations with the SVMs.

<table>
<thead>
<tr>
<th>Kernel</th>
<th>C</th>
<th>γ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>1.0</td>
<td>—</td>
</tr>
<tr>
<td>RBF</td>
<td>1.0</td>
<td>0.1</td>
</tr>
<tr>
<td>Polynomial</td>
<td>1.0</td>
<td>0.1</td>
</tr>
</tbody>
</table>

2) **Other evaluated methods**: we implemented all the remaining classifiers using the WEKA tool [25] with the default parameters. With respect to the k-nearest neighbors algorithm, we have tested traditional values for k (1, 3 and 5). As the adaptive boosting technique is very time-consuming, we have evaluated just the fast classifiers such as the boosted decision trees (C4.5) and boosted naïve Bayes.

V. RESULTS

For each evaluated method and feature set, we have performed a simulation using unbalanced classes, as originally provided by Mishne et al. [2], with 692 (≈68%) samples of spam and 332 (≈32%) ham. Moreover, to evaluate if the unbalance of the data impacts on the classifiers accuracy, we have also performed simulations using the same number of samples in each class. For this, we used the SMOTE (Synthetic Minority Oversampling Technique) [26] to create other 360 synthetic samples of ham in order to match the number of representatives in each class.

Table III and Table IV present the results achieved by each classifier exploring the text-based features, posting metadata and the combination of both using unbalanced and balanced data, respectively. In each table, the results are sorted by F-measure for each feature set and bold values indicate the highest score for each performance measure. Further, values preceded by the symbol “*” indicate the highest score considering all the simulations.

Due to space limit, the name of each method was abbreviated as follows: naïve Bayes as N.Bayes; SVM with linear, RBF and polynomial kernel as SVM-L, SVM-R and SVM-P, respectively; decision trees as C4.5; random forest as R.Forest; k-nearest neighbors as IB-k; boosted naïve Bayes as B.NB, boosted decision trees as B.C4.5, and multinomial logistic regression as Logistic.

In general, the evaluated classifiers achieved good results, since they have detected a significant amount of spam comments with high accuracy, regardless of the used feature set. Furthermore, the results indicate that multinomial logistic regression and support vector machines (with linear and polynomial kernel) presented the best overall results. Considering the lower bound, such methods were able to detect more than 92% of the spam messages with an accuracy rate superior than 85%. However, it is important to observe that the feature set extract from the posting metadata has high influence in the class separability. It is evidenced by the significant improvement in the accuracy rate achieved by the remarkable reduction of blocked ham.

It is also important to note that, since the data we used in our experiment is unbalanced, the results also indicate that all the evaluated techniques are superior when trained with the same amount of samples of each class. It is because the models tend to be biased to the benefit of the class with the largest amount of samples. Therefore, the experiments carried out with more spam samples than ham ones presented results with an unacceptable rate of blocked ham (most above 10%). On the other hand, experiments performed with balanced training set and with features extracted from the posting metadata attained an impressive low rate of blocked ham (less than 1% for the best methods).

If we compare the used set of features, we can observe that the best average result for all classifiers was achieved when the combination of text and posting metadata-based features were employed. However, the highest F-measures were achieved by SVM with polynomial kernel function (0.974) and multinomial logistic regression (0.973) with features extracted just from the posting metadata (Table IV). However, there is no statistical difference of such results from those ones attained for the same methods when all the feature sets were employed (0.972). The main difference is that, in this last scenario, the amount of blocked ham was clearly reduced with the price of wrong classify more spam as legitimate messages.

To illustrate the good potential of the results found in this work, in Table V we show a comparison between the best
results presented in this paper and the top performance machine learning techniques currently available in the literature of blog comment spamming.

It is important to note that the SVM with polynomial kernel function and multinomial logistic regression techniques achieved very high performances and clear outperformed the methods considered the state-of-the-art in the literature of blog comment spam. Therefore, it is conclusive that such machine learning methods can be successfully employed to automatically filter undesired comments.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a performance evaluation of different established machine learning algorithms used to automatically filter blog comment spam based on features extracted from messages posted by users. For this, we employed a real, public and non-encoded database composed by samples represented by text-based features, posting

<table>
<thead>
<tr>
<th>Table III: Results achieved for each combination of classifier and feature set using balanced data.</th>
<th>Acc%</th>
<th>SC%</th>
<th>BH%</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Message text-based features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM-L</td>
<td>93.75 ± 2.1</td>
<td>96.68 ± 1.4</td>
<td>12.35 ± 3.8</td>
<td>0.954 ± 0.015</td>
</tr>
<tr>
<td>SVM-P</td>
<td>92.68 ± 1.4</td>
<td>94.51 ± 1.7</td>
<td>11.14 ± 4.4</td>
<td>0.946 ± 0.010</td>
</tr>
<tr>
<td>B.NB</td>
<td>92.09 ± 2.9</td>
<td>94.85 ± 2.4</td>
<td>13.86 ± 10.1</td>
<td>0.932 ± 0.020</td>
</tr>
<tr>
<td>SVM-P</td>
<td>90.82 ± 2.9</td>
<td>98.82 ± 1.7</td>
<td>21.68 ± 38.9</td>
<td>0.935 ± 0.019</td>
</tr>
<tr>
<td>Logistic</td>
<td>90.33 ± 2.0</td>
<td>93.06 ± 2.8</td>
<td>15.36 ± 4.5</td>
<td>0.929 ± 0.015</td>
</tr>
<tr>
<td>PART</td>
<td>88.38 ± 2.1</td>
<td>90.46 ± 2.6</td>
<td>15.98 ± 5.9</td>
<td>0.913 ± 0.015</td>
</tr>
<tr>
<td>R.Forest</td>
<td>87.70 ± 2.7</td>
<td>93.93 ± 2.3</td>
<td>25.29 ± 7.8</td>
<td>0.912 ± 0.018</td>
</tr>
<tr>
<td>B.C4.5</td>
<td>87.80 ± 2.2</td>
<td>91.63 ± 3.4</td>
<td>20.16 ± 3.0</td>
<td>0.910 ± 0.017</td>
</tr>
<tr>
<td>N.Bayes</td>
<td>86.92 ± 2.1</td>
<td>87.25 ± 2.1</td>
<td>16.20 ± 3.0</td>
<td>0.897 ± 0.018</td>
</tr>
<tr>
<td>IBk-1</td>
<td>83.79 ± 3.4</td>
<td>97.84 ± 1.6</td>
<td>40.26 ± 17.8</td>
<td>0.891 ± 0.017</td>
</tr>
<tr>
<td>IBk-3</td>
<td>78.71 ± 2.5</td>
<td>99.13 ± 1.0</td>
<td>63.89 ± 7.4</td>
<td>0.863 ± 0.014</td>
</tr>
<tr>
<td>IBk-5</td>
<td>75.98 ± 2.3</td>
<td>97.71 ± 0.6</td>
<td>73.51 ± 6.6</td>
<td>0.849 ± 0.012</td>
</tr>
<tr>
<td>SVM-R</td>
<td>72.07 ± 1.5</td>
<td>97.71 ± 0.6</td>
<td>85.56 ± 4.3</td>
<td>0.828 ± 0.008</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table IV: Results achieved for each combination of classifier and feature set using balanced data.</th>
<th>Acc%</th>
<th>SC%</th>
<th>BH%</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Message text-based features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logistic</td>
<td>93.79 ± 2.8</td>
<td>92.62 ± 3.9</td>
<td>5.06 ± 3.0</td>
<td>0.937 ± 0.029</td>
</tr>
<tr>
<td>SVM-L</td>
<td>93.57 ± 2.0</td>
<td>91.90 ± 4.5</td>
<td>5.20 ± 2.4</td>
<td>0.932 ± 0.022</td>
</tr>
<tr>
<td>IBk-1</td>
<td>92.92 ± 2.3</td>
<td>90.90 ± 5.0</td>
<td>5.05 ± 2.4</td>
<td>0.927 ± 0.026</td>
</tr>
<tr>
<td>R.Forest</td>
<td>92.49 ± 2.9</td>
<td>91.47 ± 2.9</td>
<td>6.50 ± 4.2</td>
<td>0.924 ± 0.029</td>
</tr>
<tr>
<td>B.C4.5</td>
<td>89.67 ± 1.5</td>
<td>90.60 ± 3.8</td>
<td>11.27 ± 4.0</td>
<td>0.898 ± 0.015</td>
</tr>
<tr>
<td>PART</td>
<td>87.72 ± 2.6</td>
<td>89.88 ± 5.3</td>
<td>14.46 ± 4.5</td>
<td>0.879 ± 0.028</td>
</tr>
<tr>
<td>C4.5</td>
<td>86.78 ± 1.8</td>
<td>88.00 ± 4.4</td>
<td>14.45 ± 2.6</td>
<td>0.869 ± 0.021</td>
</tr>
<tr>
<td>N.Bayes</td>
<td>85.41 ± 3.5</td>
<td>88.66 ± 5.5</td>
<td>18.06 ± 5.4</td>
<td>0.859 ± 0.035</td>
</tr>
<tr>
<td>IBk-3</td>
<td>83.68 ± 3.4</td>
<td>94.07 ± 2.9</td>
<td>26.72 ± 4.7</td>
<td>0.852 ± 0.030</td>
</tr>
<tr>
<td>B.NB</td>
<td>82.88 ± 3.2</td>
<td>97.11 ± 2.3</td>
<td>31.35 ± 6.6</td>
<td>0.851 ± 0.025</td>
</tr>
<tr>
<td>SVM-R</td>
<td>84.90 ± 1.1</td>
<td>72.40 ± 5.3</td>
<td>2.60 ± 1.8</td>
<td>0.826 ± 0.039</td>
</tr>
<tr>
<td>IBk-5</td>
<td>76.38 ± 3.2</td>
<td>94.51 ± 3.0</td>
<td>41.76 ± 4.2</td>
<td>0.800 ± 0.027</td>
</tr>
<tr>
<td>SVM-P</td>
<td>76.45 ± 4.7</td>
<td>90.89 ± 3.3</td>
<td>37.99 ± 6.3</td>
<td>0.795 ± 0.039</td>
</tr>
</tbody>
</table>

| Table V: Comparison between the best results achieved in this paper and the results available in the literature. |
|-------|-------|-------|-------|
| Classifiers | Accuracy (%) | F-measure | F-measure |
| KL-divergence [2] | 83.00 | — | — |
| SMO [3] | 84.61 | — | — |
| C4.5 [4] | 86.00 | 0.890 | — |
| C4.5 auto-supervised [4] | 71.58 | 0.825 | — |

The best results presented in this paper.

SVM-P (metadata) – balanced | 97.47 | 0.974 | |
Logistic (metadata) – balanced | 97.33 | 0.973 | |
Logistic (metadata + text) – balanced | 97.33 | 0.972 | |
SVM-P (metadata + text) – balanced | 97.25 | 0.972 | |
metadata-based features and the combination of both. The results indicated that, in general, the evaluated techniques achieved good performance, regardless of the used attribute. This indicates that, besides to be efficient, they have good capability of generalization. However, the results clearly show that the feature set extract from the posting metadata has a high influence in the class separability. It can be confirmed by the significant improvement in the accuracy rate achieved by the great reduction of blocked ham.

Since the data we used in our experiment is unbalanced, the results indicated that the evaluated techniques are superior when trained with the same amount of samples of each class. It is because the models tend to be biased to the benefit of the class with the largest amount of samples.

Among all evaluated classifiers, the SVM with polynomial and linear kernel functions and the multinominal logistic regression algorithms achieved the best performances, demonstrating to be promising to identify spam comments and consequently they are recommended to be used as a good baseline for further comparison.

For future work, we intend to adapt the most promising methods in order to optimize their performances and we aim to study and propose new set of features in order to enhance the capacity of the classifiers prediction.

ACKNOWLEDGMENT

The authors would like to thank the financial support of Brazilian agencies FAPESP, Capes and CNPq.

REFERENCES