Content Based SMS Spam Filtering

José María Gómez Hidalgo
Universidad Europea de Madrid
Villaviciosa de Odón
28670 Madrid, SPAIN
34 91 211 5670
jmgomez@uem.es

Guillermo Cajigas Bringas
Group R&D - Vodafone ES
Avenida de Europa, 1
28108 Alcobendas, Madrid, SPAIN
34 610 51 34 93
guillermo.cajigas@vodafone.com

Enrique Puertas Sánz
Francisco Carrero García
Universidad Europea de Madrid
Villaviciosa de Odón
28670 Madrid, SPAIN
34 91 211 5611
enrique.puertas@uem.es
francisco.carrero@uem.es

ABSTRACT

In the recent years, we have witnessed a dramatic increment in the volume of spam email. Other related forms of spam are increasingly revealing as a problem of importance, specially the spam on Instant Messaging services (the so called SPIM), and Short Message Service (SMS) or mobile spam.

Like email spam, the SMS spam problem can be approached with legal, economic or technical measures. Among the wide range of technical measures, Bayesian filters are playing a key role in stopping email spam. In this paper, we analyze to what extent Bayesian filtering techniques used to block email spam, can be applied to the problem of detecting and stopping mobile spam. In particular, we have built two SMS spam test collections of significant size, in English and Spanish. We have tested on them a number of messages representation techniques and Machine Learning algorithms, in terms of effectiveness. Our results demonstrate that Bayesian filtering techniques can be effectively transferred from email to SMS spam.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *information filtering*.
H.3.4 [**Information Storage and Retrieval**]: Systems and Software – *performance evaluation (efficiency and effectiveness).*

General Terms

Experimentation, Security.

Keywords

Spam, junk, Bayesian filter, Receiver Operating Characteristic

1. INTRODUCTION

Mobile spam is a relevant problem in Far East countries since year 2001. In Korea, the volume of mobile spam was already bigger than the volume of email spam at the end of 2003. spam messages are used to advertise dating services, premium rate numbers, or selling drugs and software. Several countries have taken legal and technical measures to control the SMS spam problem. Japan government filed two acts in 2002, which defined and penalized email and mobile abuse. These laws, the effort of self-regulation from Mobile Network operators, and some technical limitations, have helped to reduce the volume (but not to quit) mobile spam. All in all, experts consider that mobile spam can only get controlled through the combination of technical and legal measures.

SMS spam has been regarded as a minor problem in Western countries, mostly because the cost of sending spam messages is much bigger than that of sending email spam. But in Europe, SMS messaging is the fashion: nearly all people over 15 years old own a mobile phone, and an average user sends about 10 SMS a day. That makes SMS messages a perfect target for abuse. Moreover, botnets of zombie PCs are being used to emulate real users when sending SMS messages through free SMS messaging services at e.g. Russia. So, SMS spam cost is decreasing. In other words, mobile spam can pay. In fact, more than 80% of users in EC admit to have received mobile spam

A variety of technical measures against spam email have been already proposed (see below). Most of them can effectively be transferred to the problem of mobile spam. One of the most popular ones is the so called Bayesian filters: programs that, through learning, are able to discriminate legitimate from spam messages. In this paper, we study the possibility of applying Bayesian filtering techniques to the problem of SMS spam filtering. First, we review a number of technical measures against spam email, focusing on Bayesian filtering. Then we present how the Mobile Network operator sees the problem of mobile spam in Europe, and which role can Bayesian filtering can play in reducing or stopping it. After, we describe a series of experiments designed to test to what extent it is effective addressing the mobile spam problem.

2. TECHNICAL MEASSURES AGAINST SPAM EMAIL

Millions of spam email messages are sent every day, advertising pornographic Web sites, drugs or software products, or of fraud (phishing) [2]. Spam email has an important economic impact in end users and service providers. The increasing importance of this problem has motivated the development of a set of techniques to fight it, or at least, providing some relief.

2.1 Spam Email Filtering

The most popular techniques used to reduce spam nowadays include the following ones.

White and black listing. The senders occurring in a black list (e.g. RBL) are considered spammers, and their messages blocked. The messages from senders in a white list (e.g. the address book, or the provider itself – Hotmail) are considered legitimate, and thus delivered.

Postage. For delivering a message, postage is required: economic (e.g. on cent per message), computational (e.g. the computation of a costly function) or Turing-test based (e.g. the verification of the sender being a person instead of a program). See e.g. [22].

Address management. It consists on the usage of temporal, machine-generated addresses, which are managed automatically by the system, and discarded when they begin receiving spam. See e.g. the Channels system [23].

Collaborative filtering. When a user tags a message as spam, this is considered spam for users similar to him/her. Alternatively, the service provider considers that massive messages (for e.g. more than N=20 recipient) are spam [24].

Digital signatures. Messages without a digital signature are considered spam. Digital signatures can be provided by the sender or the service provider [25].

Content-based filtering [14]. The most used method. Each messaged is searched for spam features, like indicative words (e.g. "free", "viagra", etc.), unusual distribution of punctuation marks and capital letters (like e.g. in "BUY!!!!!!"), etc.

Most of the techniques above can be directly applied to the problem of mobile spam, but among them, content-based filtering (and in particular, Bayesian filtering) is playing a key role in reducing spam email. In fact, its success has forced spammers to periodically change their practices, and to disguise their messages, in order to bypass these kinds of filters. In this work, we focus on Bayesian filtering SMS spam in Europe.

2.2 Bayesian Filtering Techniques

Content-based spam filters can be built manually, by hand-engineering the set of attributes that define spam messages. These are often called heuristic filters [4], and some popular filters like SpamAssassin have been based on this idea for years. Content-based filters can also be built by using Machine Learning techniques applied to a set of pre-classified messages [16]. These so-called Bayesian filters are very accurate according to recent statistics [1], and their applicability to SMS spam seems immediate.

Bayesian[1] filters [3] automatically induce or learn a spam classifier from a set of manually classified examples of spam and legitimate (or *ham*) messages (the training collection). The learning process takes as input the training collection, and consists of the following steps [14]:

- *Preprocessing*. Deletion of irrelevant elements (e.g. HTML), and selection of the segments suitable of processing (e.g. headers, body, etc.).
- *Tokenization*. Dividing the message into semantically coherent segments (e.g. words, other character strings, etc.).
- *Representation*. Conversion of a message into an attribute-value pairs' vector [13], where the attributes are the previously defined tokens, and their values can be binary, (relative) frequencies, etc.
- *Selection*. Statistical deletion of less predictive attributes (using e.g. quality metrics like Information Gain).
- *Learning*. Automatically building a classification model (the classifier) from the collection of messages, as they have been previously represented. The shape of the classifier depends on the learning algorithm used, ranging from decision trees (C4.5), or classification rules (Ripper), to statistical linear models (Support Vector Machines, Winnow), neural networks, genetic algorithms, etc.

Each new target message is pre-processed, tokenized, represented and feed into the classifier, in order to take a classification decision on it (whether it is spam or not).

Current methods in Bayesian filter development are focused on the first steps, given that the quality of representation has big impact on the accuracy of the learned model. It is noteworthy that some researchers have developed highly accurate filters by employing character-level tokenization, putting nearly all the intelligence of the filter in the learning method (a for of text compression) [20].

2.3 Application to Mobile Spam

Since having a good term representation is one of the most important parts for getting a good classifier, we have to face the fact that SMS messages have not the same structure and characteristics than email messages. We have described techniques used to filter spam email messages, but we cannot state they can be also effective filtering SMS.

SMS are usually shorter than email messages. Only 160 characters are allowed in a standard SMS text, and that could be a problem because using fewer words means less information to work with.

Also, due to the above constraint, people tend to use acronyms when writing SMS. Moreover, the abbreviations used by SMS users are not standard for a language, but they depend on the users communities. Such language variability provides more terms or features, and a more sparse representation. We have to test if the state of the art methods used to extract terms from email messages are also suitable for SMS texts.

3. SMS SPAM IN EUROPE

The threat of mobile spam is clear, as everybody feels like the mobile handset has become a very personal piece of technology and want to keep it useful, personal and free of invasions like viruses and spam. From the European Mobile Network operator (MNO) point of view, we can classify mobile spam according to

[1] Learning-based content filters would be a more accurate name for this kind of filters, because many of them do not use Bayesian learning methods at all. However, we prefer to use the name Bayesian, as it has spread across the research and technical literature, to avoid misunderstandings. The very first experiment with learning based spam filters is that by Sahami et al. [21].

how it is produced. The final user can roughly receive mobile spam from three main sources:

- MNO or parties that pay MNOs for delivering the SMS to the final user.
- Parties that manage not to pay for the SMS that are finally delivered to the user.
- User originated messages that bother the receiver.

The first case seems to be the main responsible of the high number of users admitting they have received spam in Europe. MNOs, third parties and authorities have adopted and enforced the use of opt-out, or even opt-in (for the case of third parties) processes for the user to stop receiving promos or ads. MNOs often disconnect parties that do not comply with MNO policies of legitimate SMS.

The second case is usually worse as it is a fraud, and not only it damages MNO brand but also its revenue stream. MNOs have already installed tools and processes to detect and cut off this kind of sources.

Finally, although the third case is statistically irrelevant, it can produce user complaints and it cannot be easily managed due to SMS content privacy regulations and business commitments acquired with user in SMS service delivery.

In fact, processes at the MNO side and regulations at the authorities' side seem to be effective enough to have lowered the number of user complaints and keep them stable in Europe. To be honest, this maybe can also be imputed to the tolerance raise that the users seem to experience.

Anyhow it also true that this kind of tools and processes are reactive and do leak some of this mobile spam to final users. For instance, from the moment the second kind of abuse begins, to the moment it is terminated, the end users still receive SMS spam, and they perceive a decrease of the quality of service. It is also noteworthy that spam definition is user dependent: what I consider spam could be information to for you. In this context, personal or easily personalized, learning filtering tools could help in reducing even more the final user's complaints, thus helping MNO delivering a better service to them.

4. EXPERIMENTS AND RESULTS

We have conducted a series of experiments, with different attribute definitions, several learning algorithms and a suitable evaluation method, in order to test if Bayesian filtering techniques can be easily transferred to SMS spam filtering. There is some evidence of this [17], but more systematic experiments are required.

4.1 Test Collections

We have built two different collections: one with SMS messages in Spanish and another one with messages in English.

4.1.1 Spanish test database

For this round of experiments we have been provided by Vodafone with a sample of Spanish spam messages, obtained from Vodafone users. The legitimate messages have been taken from a joke by SMS competition. We have built and used a message database consisting of 199 (14.67%) spam messages, and 1,157 (85.32%) legitimate messages. A trivial rejecter (the classifier that always selects the majority class (legitimate or "ham" in our case), would show an accuracy of 0.85.

4.1.2 English test database

We also built an English SMS database by using freely available resources on the Internet. After a detailed search, we have found the following resources:

- A list of 202 legitimate messages, probably collected by Jon Stevenson, according to the HTML code of the Web page. Only the text of the messages is available. We will call this corpus the *Jon Stevenson Corpus* (JSC).
- A collection of about 10,000 legitimate messages collected for research at the Department of Computer Science at the National University of Singapore, called the *NUS SMS Corpus* (NSC). The messages largely originate from Singaporeans and mostly from students attending the University. These messages were collected from volunteers who were made aware that their contributions were going to be made publicly available.
- A collection of 82 SMS spam messages extracted manually from Grumbletext. This is a UK forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the very spam message received. The identification of the text of spam messages in the claims is a very hard and time-consuming task, and it involved carefully scanning 100 web pages.

By using the messages collected by Jon Stevenson, and a random sample of the messages in the NUS SMS Corpus, we have built SMS English database consisting of 1,119 legitimate messages, and 82 spam messages. We believe this collection resembles a realistic scenario, because both the legitimate and the spam messages are real messages; the proportion may be not accurate, but we are not aware of the existence of real world statistics of spam received by cell phone users in the British/Singapore markets.

4.2 Message Processing and Encoding

According to our previous experience and the literature on spam email filtering, the tokenization step is probably the most important one in the process of analysing messages and learning a classifier on them. A bad representation of the problem data may lead to classifier of poor quality and accuracy. So we have scanned carefully the literature in spam email detection, in order to build a set of attributes or parameters that guarantee that learning will be successful.

We have decided to use the following set of attributes to represent SMS messages (either spam or legitimate):

- Words – sequences of alpha-numeric characters in the message text. We consider that any non-alphanumeric character is a separator. Words are the building blocks of any message.
- Lowercased words – lowercased words in the text message, according to the definition of word above. This way, we map different strings to the same attributes, obtaining new token frequencies that should affect the learning stage.

- Character bi-grams and tri-grams – sequences of 2 or 3 characters included in any lowercased word. This attributes try to capture morphological variance and regularities in a language-independent way. For instance, if the simple past's suffix "ed" in English, is representative of spam messages, we would be able to capture this knowledge.
- Word bi-grams – sequences of 2 words in a window of 5 words preceding the current word. On one side, it has been demonstrated that most relevant dependences between words in a language are present only in a window of size 5; that is, rarely a word influences another that is far of 5 tokens from itself. On the other side, this kind of word bi-grams have been proven very useful in spam email detection

4.3 Feature Selection

In a data mining inspired approach, we have decided to feed the test with a possibly very big number of attributes, letting the attribute selection step the responsibility for deleting less informative attributes. We use Information Gain (IG) [18, 19] as attribute quality metric. The experience in learning-based text classification is that IG can reduce substantially the number of attributes, without no loss (or even some improvement) of accuracy.

We have made experiments selecting all tokens scoring over 0 (zero) in Information Gain, and sets with 100 and 200 tokens with highest IG. Such tokens may provide information for the spam class (that is, they correlate to it), or for the legitimate class.

4.4 Machine Learning Algorithms

For our experiments we have used the following algorithms[2]:

- Naive Bayes (NB) [10]. This is the most immediate and simple Bayesian learning method. It is based on the Bayes theorem and an independence hypothesis, generating an statistical classifier based on probabilities. Its name has been borrowed by the class of Machine Learning based email spam filters. It is the simplest technique, and it must be used before trying more complex methods.
- C4.5 [12]. A classical learning method that outputs decision trees, with average results on text classification problems. Its main advantage is that decision trees are a simple, clear and expressive formalism.
- PART [7]. This algorithm produces classification rules, based on generating decision rule lists from partial decision trees. It shares the advantages and limitations of C4.5.
- Support Vector Machines (SVM) [6, 9]. An optimisation algorithm that produces (linear) vectors that try to maximally separate the target classes (spam versus legitimate). It is a complex and (relatively) recent algorithm (1997), which has shown excellent results in text classification applications. However, it is difficult to interpret its output.

These algorithms represent a good sample of the current learning algorithms available in the market, and their learning strategies are different, as their learning bias [17].

For this round of experiment, we have also used stratification or re-weighting mechanism, trying to make algorithms more sensitive to the (underrepresented) class spam. The goal is to reduce the number of false negatives (spam classified as Ham), in order to block more spam messages [8].

The mechanism consists of incrementing the weight of (or giving more importance to) spam messages in the training collection [15], what is equivalent to stratification: incrementing the number of spam messages by inserting more copies of available ones. For instance, if spam messages are given a weight of 50 (against a weight of 1 for a legitimate message), we force the algorithm to consider each spam message as 50 messages. Since the algorithms we use try to minimize the error (or to maximize the accuracy), a mistake on a spam message (a false negative) is 50 times more important than a mistake on a legitimate message (a false positive) [11]. Anyway, we use cost-weighting as a mechanism focused on finding the accuracy tolerance of the studied algorithms in the provided test collection. In our tests, we use 10 and 100 weights for spam messages, against a stable weight of 1 for legitimate messages.

4.5 Evaluation Setup

As we need to select the most suitable tokenization and learning methods for an operation environment, we have to perform a evaluation oriented to a number of variable conditions, in terms of class distribution (and balance) and cost of false positives (and negatives), that can not be known in advance.

The most suitable evaluation method for such an imprecise environment is the Receiver Operating Characteristic Convex Hull (ROCCH) method [11]. We have employed this method in previous experiments in spam email filtering, and in Web content filtering, two applications that share this imprecise nature; also, this method has became an standard in spam filtering, according to the most well-known spam email filtering competition nowadays, the TREC spam Track.

4.5.1 The ROCCH Method

The Receiver Operating Characteristics (ROC) analysis is a method for evaluating and comparing a classifiers performance [8]. It has been extensively used in signal detection, and it was introduced and extended in [11] for the Machine Learning community. In ROC analysis, instead of a single value of accuracy, a pair of values is recorded for different class and cost conditions a classifier is learned. The values recorded are the False Positive rate (FP) and the True Positive rate (TP), defined in terms of the confusion matrix as:

$$\mathrm{FP} = \frac{\mathrm{fp}}{\mathrm{fp} + \mathrm{tn}}$$

$$\mathrm{TP} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}}$$

In this formulas, "fp" is the number of false positives (legitimate classified as Spam), "tp" is the number of true positives (Spam classified as Spam), etc. The TP rate is equivalent to the recall of the positive class, while the FP rate is equivalent to 1 less the recall of the negative class. Each (FP, TP) pair is plotted as a

[2] The implementation used in our experiments is the one included in the learning package WEKA, with the default parameters.

point in the ROC space. Most ML algorithms produce different classifiers in different class and cost conditions. For these algorithms, the conditions are varied to obtain a ROC curve.

One point on a ROC diagram dominates another if it is above and to the left, i.e. it has a higher TP and a lower FP. Dominance implies superior performance for a variety of commonly performance measures, including Expected Cost, Weighted Accuracy, Weighted Error, recall and others. Given a set of ROC curves for several ML algorithms, the one which is closer to the left upper corner of the ROC space represents the best algorithm.

The ROC analysis allows a visual comparison of the performance of a set of ML algorithms, regardless of the class and cost conditions. This way, the decision of which is the best classifier or ML algorithm can be delayed until target (real world) conditions are known.

Instead of drawing a ROC curve through threshold variation, we can vary the class and cost conditions and obtain for each of them a (FP, TP) point using the Threshold method. This view of ROC curve plotting allows using other methods for making ML algorithms cost-sensitive. For instance, one can use techniques as Stratification or MetaCost [5] applied to a ML algorithm for inducing a set of classifiers for a range of class and cost conditions, and then linking the obtained (FP,TP) points to form a ROC curve.

This is the basis of the method we have applied to obtain ROC curves for a range of methods for making ML algorithms cost-sensitive.

We have computed our ROC curves by 10-fold cross validation. This method consists of dividing the test collection into 10 sets, and using nine for learning and one for testing 10 times. This method is designed to take advantage of all the learning data when it is scarce. ROC curves can be computed in a kind of operational environment, which has been used in the Text Retrieval Conferences Spam Track: the filter classifies the messages one by one, and it is feed with the true class of the message, allowing it to learn on its mistakes or even on all messages.

4.6 Results and Discussion

The output of a ROCCH evaluation is a table or plot showing the ROC points or curves for a number of algorithms, and a table showing the slope ranges in which the classifiers laying on the Convex Hull are optimal.

4.6.1 Results and analysis, English database

First, we show in the Figure 1 the result of experiments for all the tested algorithms and each number of attributes (100, 200 and all with IG over zero) for the English language database, obtained from all the algorithms tested[3].

We must remind the reader that in a ROC plot, the optimal point is (0,1), that represents no False Positives (no legitimate messages classified as spam) and a maximum of True Positives (spam messages classified as spam). The closer a point is to this point (or to the upper left corner of the plot), the better it performs.

For all the attribute set sizes, the dominant classifier is SVM. This fits experiments in the literature on text classification and spam email detection, where this algorithm has performed usually much better than the others tested in this work. This can be also observed in the Table 1, showing the optimal classifiers for the slope ranges presented.

The different classifiers (and their (FP, TP) points) correspond to 10-fold cross validated tests encoded the following way:

- first the code of the algorithm (NB, C45, PART, SVM);
- then, the cost ratio, meaning e.g. 020 that a false positive (classifying a legitimate messages as spam) is 20 times more important than a false negative, and e.g. i030 that a false positive is 30 times less important than a false negative;

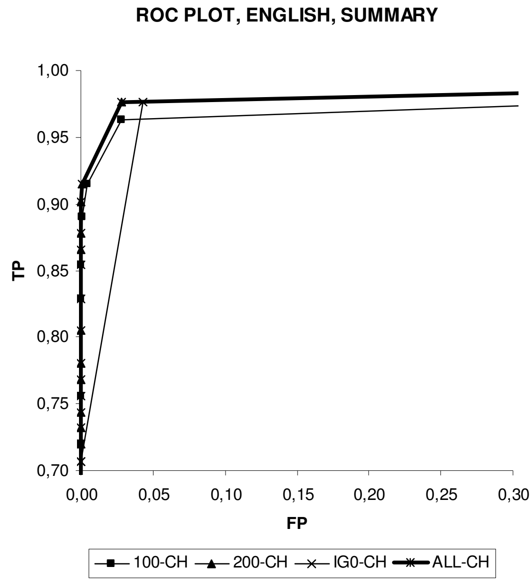

Figure 1. The ROC curves and Convex Hull, English database.

- finally, the number of attributes used for learning, meaning IG0 those attributes with an IG score over zero.

For a given class and cost distribution, we can compute the corresponding slope and find the optimal classifier. For instance, given the current class distribution in the database: P(+) =0,07, P(-) =0,93, and a cost distribution of 1/1 (that means balanced error costs, C(+,-) = C(-,+)), the distribution slope S is:

$$S = \frac{P(-)\times C(+,-)}{P(+)\times C(-,+)} = \frac{0{,}93}{0{,}07} = 13{,}28$$

Given this slope value, the optimal classifiers are:

- SVM-001 (Support Vector Machines with Cost Ratio of one), for the cases with 100 and 200 attributes, or those attributes IG-scoring over zero.
- SVM-001-200 (the previous one but with 200 attributes) in general, because it achieves a better TP rate for the same or less FP rate.

[3] C4.5 has not performed optimally in any case, and thus it will not be shown in any table or plot.

Table 1. Slope ranges for various settings, English database

Slope ranges, English, 100 attributes		
Slope Range	**ROC point**	**Classifier**
[0.000,0.038]	1.000,1.000	AllPos
[0.038,2.000]	0.028,0.963	NB-i090
[2.000,8.333]	0.004,0.915	SVM-i010
[8.333,36.000]	0.001,0.890	SVM-001
[36.000,Inf]	0.000,0.854	SVM-005
Slope ranges, English, 200 attributes		
[0.000,0.025]	1.000,1.000	AllPos
[0.025,2.259]	0.028,0.976	NB-005
[2.259,13.000]	0.001,0.915	SVM-i005
[13.000,Inf]	0.000,0.902	SVM-001
Slope ranges, English, attributes IG>0		
[0.000,0.025]	1.000,1.000	AllPos
[0.025,6.256]	0.043,0.976	NB-1000
[6.256,Inf]	0.000,0.707	SVM-001
Slope ranges, English, summary		
[0.000,0.025]	1.000,1.000	AllPos
[0.025,2.259]	0.028,0.976	NB-005-200
[2.259,13.000]	0.001,0.915	SVM-i005-200
[13.000,Inf]	0.000,0.902	SVM-001-200

It is noteworthy that for 100 attributes, SVM makes one FP, capturing an 89% of spam messages. The optimal situation, represented by the SVM-001-200 classifier, allows detecting over 90% of spam messages, in a quite safe-for-the-user environment.

These results demonstrate that SVMs are optimal under the Newman-Pearson criterion for the most reasonable operating scenario. This criterion consists of setting a maximum acceptable FP rate FPMAX, which corresponds to a vertical line in a ROC graph, and selecting the best classifier with FP rate under FPMAX, that is, the one with optimal TP rate.

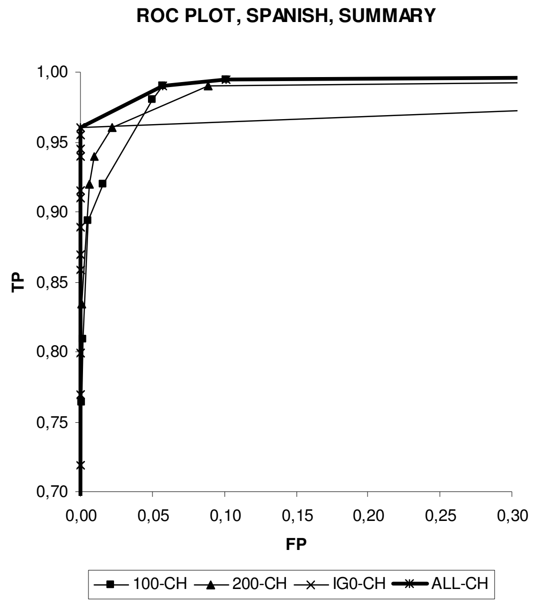

Figure 2. The ROC curves and Convex Hull, Spanish database.

In safe-for-the-user environment, we set the FP rate to zero (no false positives allowed), finding also the same optimal classifier, SVM-001-200.

4.6.2 Results and analysis, Spanish database

Next, we show in Figure 2 the result of experiments for all the tested algorithms and each number of attributes (100, 200 and all with IG over zero) for the Spanish language database.

The optimal classifiers for the different attribute set sizes, and slope ranges, are given in the Table 2. This table follows the notation of the previous section.

The most noteworthy fact is that, again, Support Vector Machines perform over the other algorithms most of the time. Given that the distribution of messages is more balanced in this collection, allowing finding more predictive attributes, accuracy can score over 95% if we set the FP rate to zero, or even over 99% if we allow few FPs.

Table 2. Slope ranges for various settings, Spanish database.

Slope ranges, Spanish, 100 attributes		
Slope Range	**ROC point**	**Classifier**
[0.000,0.006]	1.000,1.000	AllPos
[0.006,0.114]	0.101,0.995	SVM-i200
[0.114,1.429]	0.057,0.990	SVM-i060
[1.429,1.765]	0.050,0.980	SVM-i040
[1.765,2.364]	0.016,0.920	SVM-i005
[2.364,28.333]	0.005,0.894	SVM-001
[28.333,45.000]	0.002,0.809	SVM-010
[45.000,206.00]	0.001,0.764	SVM-030
[206.00,Inf]	0.000,0.558	SVM-300
Slope ranges, Spanish, 200 attributes		
Slope Range	**ROC point**	**Classifier**
[0.000,0.011]	1.000,1.000	AllPos
[0.011,0.011]	0.532,0.995	PART-i900
[0.011,0.448]	0.089,0.990	SVM-i500
[0.448,1.667]	0.022,0.960	SVM-i030
[1.667,5.000]	0.010,0.940	SVM-i005
[5.000,17.200]	0.006,0.920	SVM-001
[17.200,256.00]	0.001,0.834	SVM-020
[256.00,Inf]	0.000,0.578	SVM-400
Slope ranges, Spanish, attributes IG>0		
Slope Range	**ROC point**	**Classifier**
[0.000,0.040]	1.000,1.000	AllPos
[0.040,Inf]	0.000,0.960	SVM-i050
Slope ranges, Spanish, summary		
Slope Range	**ROC point**	**Classifier**
[0.000,0.006]	1.000,1.000	AllPos
[0.006,0.114]	0.101,0.995	SVM-i200-100
[0.114,0.526]	0.057,0.990	SVM-i060-100
[0.526,Inf]	0.000,0.960	SVM -i050-IG0

Let us examine the slope corresponding to the actual distribution of classes and costs in the collection. Given that P(+)=0,146 and P(-)=0,853, and for a cost ratio of one, the slope value is S = 5,81. The most accurate (and appropriate) classifier for this conditions is SVM-i050-IG0 (Support Vector Machines trained with all attributes with IG over zero, in a cost ratio of 1/50 – a false negative is 50 times more important that a false positive))[4]. It may seem counter-intuitive to over-weight false negatives, being these less desirable than false positives (legitimate messages classified as spam), but we must note that the ROCCH method involves testing the algorithms over a full range of (cost) conditions. Also, an even not being on the Convex Hull (or optimal performance plot), the SVM-020-IG0 reaches a TP rate of 0,955 (that is, 95,5% of spam messages detected) with a FP rate of zero, very close to the optimal one.

Finally, the SVM-i050-IG0 is the optimal classifier under the Newman-Pearson criterion, when setting the maximum allowed FP rate to zero.

5. CONCLUSIONS

From this series of experiments, we can derive the following conclusions:

- Given the short size of messages, and the literature on spam email filtering, it is reasonable to define a wide range of attribute types, and let the attribute selection using IG process to select those most promising for classification. However, the number of selected attributes can not be known in advance, although it seems proportional to the spam messages. It may be valuable to test other kinds of features (e.g. encoding all numbers, or marking telephone numbers).
- The most suitable learning algorithm for a prototype is, after in-depth evaluation, Support Vector Machines. This is supported by our and others' previous work in spam email detection and in text classification. Also, although we have not demonstrated this empirically, the running time of learning with Support Vector Machines has been comparable to Naïve Bayes, and much smaller than the running time for learning rules or decision trees.

6. ACKNOWLEDGMENTS

We give our thanks to VODAFONE for funding this research and providing us with the Spanish test data.

7. REFERENCES

[1] Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Paliouras, G., Spyropoulos, C.D. An Evaluation of Naive Bayesian Anti-spam Filtering. Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML 2000), pp. 9-17, 2000.

[2] Christine E. Drakeand Jonathan J. Oliver, Eugene J. Koontz. Anatomy of a Phishing Email. Proceedings of the First Conference on Email and Anti-spam (CEAS), 2004.

[3] Graham, Paul. Better Bayesian Filtering. Proceedings of the 2003 Spam Conference, January 2003.

[4] Gómez, J.M., Maña-López, M., Puertas, E. Combining Text and Heuristics for Cost-Sensitive spam Filtering. Proceedings of the Fourth Computational Natural Language Learning Workshop, CoNLL-2000, Association for Computational Linguistics, 2000.

[5] Domingos, P. 1999. Metacost: A general method for making classifiers cost-sensitive. Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining.

[6] Drucker, H, Vapnik, V., Wu, D. Support Vector Machines for spam Categorization. IEEE Transactions on Neural Networks, 10(5), pp. 1048-1054, 1999.

[7] Frank, E., I.H. Witten. 1998. Generating accurate rule sets without global optimization. Machine Learning: Proceedings of the Fifteenth International Conference.

[8] Gómez, J.M. 2002. Evaluating cost-sensitive unsolicited bulk email categorization. Proceedings of the ACM Symposium on Applied Computing.

[9] Joachims, T. 2001. A statistical learning model of text classification with support vector machines. En Proceedings of the 24th ACM International Conference on Research and Development in Information Retrieval. ACM Press.

[10] Lewis, D.D. 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. En Proceedings of the 10th European Conference on Machine Learning. Springer Verlag.

[11] Provost, F., T. Fawcett. 2001. Robust classification for imprecise environments. Machine Learning Journal, 42(3):203-231.

[12] Quinlan, J.R. 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann.

[13] Salton, G. 1989. Automatic text processing: the transformation, analysis and retrieval of information by computer. Addison-Wesley.

[14] Sebastiani, F., 2002. Machine learning in automated text categorization. ACM Computing Surveys, 34(1):1-47.

[15] Ting, K.M. 1998. Inducing cost-sensitive trees via instance weighting. En Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery, 139-147.

[16] Witten, I.H., E. Frank. 1999. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann.

[17] Xiang,Y., Chowdhury, M., Ali, S. Filtering Mobile spam by Support Vector Machine. Proceedings of CSITeA-04 , ISCA Press, December 27-29, 2004.

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

[19] Yang, Y., J.O. Pedersen. 1997. A comparative study on feature selection in text categorization. En Proceedings of the 14th International Conference on Machine Learning.

[20] Bratko, A, B. Filipic. Spam Filtering using Character-level Markov Models: Experiments for the TREC 2005 Spam Track. Proceedings of the 2005 Text Retrieval Conference, 2005.

[4] One of the strengths of the ROCCH method is that it is able to detect that a specific classifier for a given cost may me optimal for other cost distributions, given that the class distribution affects also classifiers learning and performance.

[21] Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.. A bayesian approach to filtering junk e-mail. In Learning for Text Categorization: Papers from the 1998 Workshop, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.

[22] Dwork, C., Goldberg A., Naor M.. On memory-bound functions for fighting spam. In Proceedings of the 23rd Annual International Cryptology Conference (CRYPTO 2003), August 2003.

[23] R.J. Hall. How to avoid unwanted email. Communications of the ACM, March 1998.

[24] Golbeck, J., Hendler, J. Reputation network analysis for email filtering. In Proceedings of the First Conference on Email and Anti-Spam (CEAS), 2004.

[25] Tompkins T., Handley D. Giving e-mail back to the users: Using digital signatures to solve the spam problem. First Monday, 8(9), September 2003.