A Comparative Performance Study of Feature Selection Methods for the Anti-spam Filtering Domain

J.R. Méndez[1], F. Fdez-Riverola[1], F. Díaz[2], E.L. Iglesias[1], and J.M. Corchado[3]

[1] Dept. Informática, University of Vigo, Escuela Superior de Ingeniería Informática,
Edificio Politécnico, Campus Universitario As Lagoas s/n, 32004, Ourense, Spain
{moncho.mendez, riverola, eva}@uvigo.es
[2] Dept. Informática, University of Valladolid, Escuela Universitaria de Informática,
Plaza Santa Eulalia, 9-11, 40005, Segovia, Spain
fdiaz@infor.uva.es
[3] Dept. Informática y Automática, University of Salamanca,
Plaza de la Merced s/n, 37008, Salamanca, Spain
corchado@usal.es

Abstract. In this paper we analyse the strengths and weaknesses of the mainly used feature selection methods in text categorization when they are applied to the spam problem domain. Several experiments with different feature selection methods and content-based filtering techniques are carried out and discussed. Information Gain, χ^2-text, Mutual Information and Document Frequency feature selection methods have been analysed in conjunction with Naïve Bayes, boosting trees, Support Vector Machines and ECUE models in different scenarios. From the experiments carried out the underlying ideas behind feature selection methods are identified and applied for improving the feature selection process of SpamHunting, a novel anti-spam filtering software able to accurate classify suspicious e-mails.

1 Introduction and Motivation

Nowadays Internet mail service (e-mail) has become essential in the enterprise and personal productivity. The amount of messages flowing throw the e-mail servers has been increasing during last years. Everyday, Internet mail is used to send a great amount of documents with a wide variety of data and information. However, some of the contents sent across Internet using e-mail servers are useless and unwanted. Frequently they are advertising and/or fraudulent messages known as spam e-mails.

As the amount of spam messages is constantly increasing, a considerable money loss has been caused [1]. Recently, several legal and technical actions have been applied in order to combat spam e-mails. The former ones are based on adjusting the international laws including sanctions for spammers (senders of e-mail messages) whereas the use of anti-spam filtering software are the basis of the later ones. However, the effectiveness of both methods has been very limited.

At the moment anti-spam filtering software seems to be the most viable solution to spam problem. Spam filtering methods are often classified as *collaborative* or *content-based* [2]. The collaborative filtering entails the collecting of some identifying information about spam messages such as the subject, the sender or the result of com-

P. Perner (Ed.): ICDM 2006, LNAI 4065, pp. 106 – 120, 2006.

puting a hash function over the body of the message [3]. The collected data is shared with the community in the form of a digital fingerprint of each spam message. The community users can obtain the existing spam message fingerprints and use them for identifying spam e-mails which had been previously received and categorized by other users.

Despite there is no doubt that collaborative techniques help to spam filtering, they are very simplistic and unable to generalize over knowledge. Due to this fact, content-based approaches had become very popular during last years. Content-based techniques are based on analysing intrinsic properties extracted from the messages (e.g. message subject, body contents, structure, etc.) [4]. This kind of approaches is more effective than previous one because new spam messages can be correctly classified by using generalization methods over the extracted features from examined e-mails.

In the framework of content-based techniques, traditionally two main approaches have been adopted to the problem of spam filtering according to how the classification is generated by the system: *eager* vs. *lazy* learning models. The main types of content-based techniques are machine learning (ML) algorithms and case/instance-based (memory-based) reasoning approaches. ML approaches use an algorithm to 'learn' the classification from a set of training messages. On the other hand, memory-based and case-based reasoning techniques store all training instances in a memory structure and try to classify new messages finding similar e-mails on it. Hence, the decision of how to classify an incoming message is deferred until the last moment.

Regardless of the selected learning strategy, in order to train and test content-based filters it is necessary to build a large corpus with spam and legitimate e-mails or use a publicly available corpus. Anyway, e-mails have to be preprocessed to extract their words (*features*) belonging to the message subject, the body and/or the attachments. Also, since the number of features in a corpus can end up being very high, it is common to choose those features that better represent each message before carrying out the filter training to prevent the classifiers from over-fitting [5]. The effectiveness of content-based anti-spam filters relies on the appropriate choice of the features. If the features are chosen so that they may exist both in a spam and legitimate messages then, no matter how good learning algorithm is, it will make mistakes. Therefore, the preprocessing steps of e-mail features extraction and the later selection of the most representative are crucial for the performance of the filter.

For several years we have been working in the identification of techniques to completely automate the reasoning cycle of case based reasoning (CBR) systems [6, 7] and lately we have been applying all this knowledge in the development of an anti-spam filtering software called SpamHunting [8]. Our model implements an instance-based anti-spam system which successfully combine a dynamical *k-nn* strategy to retrieve the most similar messages and an innovative feature selection method based on identifying the most relevant features of each e-mail. As classical feature selection methods are not straightforwardly applicable within our model (due to its underground ideas), we are trying to reach the best conclusions behind them applied on the anti-spam filtering domain in order to improve our newly feature selection method.

In this paper, we analyse what are the strengths and the weaknesses of different feature selection methods employed in text categorization when they are applied to the spam problem domain. Therefore, we will show the results obtained by different well-known content-based techniques when the preprocessing of the training corpus changes. The selected models for the evaluation were Naïve Bayes [9], boosting trees [10], Support Vector Machines [11] and a case-based system for spam filtering named ECUE [12]. The feature selection methods we take into account were *Document Frequency* [13], *Information Gain* [14], *Mutual Information* [15, 16] and a χ^2*-test* [13].

Another relevant issue tackled in this paper is analysing how feature selection methods can be affected by noise data. Several significant terms such as 'viagra' or 'mortgage' are often obfuscated in spam messages ('v1agra', 'm0rtgage') in order to difficult class identification. Spammers are constantly innovating in word hiding tricks in order to decrease the anti-spam filtering software effectiveness. Finally, several attachments of e-mail messages can contain interesting features helpful for classifying it. We are interested in knowing if the results get better when attachments are processed. In this way, experiments have been carried out when incorporating features extracted from the attachments belonging to the messages and without it.

The rest of the paper is structured as follows: Section 2 summarizes previous work on machine learning techniques and case-based systems successfully applied to the anti-spam filtering domain. In Section 3, we describe the selected corpus for empirical model evaluation and discuss several issues related with message representation and feature selection. Then, Section 4 presents the experiments carried out and the results obtained discussing the major findings. Finally, Section 5 outlines the conclusions obtained from experimentation and presents further work.

2 Content-Based Techniques for Spam Filtering

This section introduces a brief description of the most referenced content-based techniques used in the anti-spam filtering domain. Subsection 2.1 presents a short description of classical machine learning models while Subsection 2.2 summarizes a small review of case-based and memory-based methods. Finally, Subsection 2.3 contains a detailed introduction to our Spam-Hunting model for spam labelling and filtering.

2.1 Machine Learning Approaches

Several machine learning algorithms used in text categorization [17] have also been applied to spam filtering due to the fact that classifying spam e-mails based on the textual content of the messages can be seen as a special case of categorization, with the categories being 'spam' and 'legitimate'. Regarding this subject, the most accurate techniques we should mention are Naïve Bayes, Support Vector Machines and Boosting methods because they have lead to successful research activities in the spam filtering domain.

The first research studies primarily focused on the problem of filtering spam were those of Sahami *et al.* [9] and Drucker *et al.* [18]. In [9], the authors trained a Naïve Bayesian (NB) classifier on manually categorized legitimate and spam messages, reporting impressive precision and recall on unseen e-mails. On the other hand, Drucker *et al.* verified the validity of SVMs' effectiveness in spam detection in [18].

SVMs are based on the *Structural Risk Minimization* principle [11] from computational learning theory. The idea behind structural risk minimization is to find a hypothesis for which one can guarantee the lowest true error. The true error is the probability that the hypothesis will make an error on an unseen and randomly selected test example (new e-mail message). The training of a SVM is usually slow but an optimized algorithm called *Sequential Minimal Optimization* (SMO) has demonstrated a good trade-off between accuracy and speed (see [19] for details).

Besides NB and SVM models, boosting methods are also well-known ML techniques used in this field. The purpose of boosting is to find a highly accurate classification rule by combining many *weak learners* (or weak hypotheses), each of which may be only moderately accurate [10]. The main idea of boosting is to combine the hypotheses to one final hypothesis, in order to achieve higher accuracy than the weak learner's hypothesis would have. From the several boosting algorithms that have been applied for classification tasks, we could highlight Adaboost [20].

An important aspect to take into account is that representative features on spam and legitimate messages can change with the course of time. So for example, the presence of the term 'rolex' has been indicative of legitimate e-mails several years ago but nowadays the existence of this feature is a clue for detecting spam e-mails. This kind of changes in the context of spam domain is the cause of the concept drift problem [21]. Recent work in ML techniques applied to spam detection is taking into account this situation in two different ways: (*i*) improving the performance over current ML models [22, 23, 24] and (*ii*) handling the concept drift problem [25, 26, 27].

2.2 Case-Based and Memory-Based Reasoning Approaches

Because of the changing nature of spam, a anti-spam filtering software using some machine learning approach will need to be dynamic. Several researches have suggested that a memory-based approach may work well [12]. Instance-based (or memory-based) methods are characterized by using a memory structure where each training instance is stored. The utilization of a memory structure should make the retrieval of similar instances easier and faster. In the operation mode, e-mails retrieved are used directly for classification purposes. The main advantages derived from the use of instance-based models are its capacity to continuous updating, manage disjoint concepts and handle concept drift.

In [28] a preliminary evaluation of using memory-based models in spam filtering domain is shown. In this work, TiMBL software [29] (which implements several memory-based learning techniques) is used for identifying the set of training instances in the *k* closest distances from the target problem. The solution is computed taking into account the retrieved instances by a voting strategy which gives priority to legitimate e-mails by a weighting process.

Later on, some CBR systems have been successfully adapted in this domain combining ideas from memory-based models and lazy learning. Case-based reasoning (CBR) is a *lazy* approach to machine learning where induction is delayed until run time. In [12] a case-based system for anti-spam filtering called ECUE (*E-mail Classification Using Examples*) is presented. ECUE can learn dynamically and each e-mail is a case represented as a vector of binary features. If the feature exists in the e-mail then the case assigns to the feature a value of *true*, otherwise the value of the feature is set to *false*.

The ECUE system uses a *k-nn* classifier to retrieve the *k* most similar cases to a target case. The similarity retrieval algorithm is based on Case Retrieval Nets (CRN)

[30], which is a memory structure that allows efficient and flexible retrieval of cases. ECUE classifier uses unanimous voting to determine whether a new e-mail is spam or not. In order to classify a new message as spam, all the returned neighbours need to be classified as spam e-mails.

Recently we have developed a novel instance-based anti-spam filtering system called SpamHunting [8]. Next subsection contains a summarized description of the model operation.

2.3 SpamHunting IBR System

Our SpamHunting system is a lazy learning hybrid model based on an Instance Based Reasoning approach able to solve the problem of spam labeling and filtering [8]. This system uses an Enhanced Instance Retrieval Network (EIRN) model that effectively indexes all e-mails in the instance base.

At the preprocessing stage, text is extracted from the body of each message. PDF, images and HTML documents attached to the e-mail are also processed and converted to text. Then, text is tokenized by using space, carriage return and tabulator chars as token separators. Finally, a stopword removal process is performed over identified tokens by using the stopword list given in [31].

The feature selection process is carried out in an independent way for each training and testing e-mail. Therefore, each message has its own relevant features. The main idea behind feature selection in SpamHunting is finding the best fitting features for each e-mail. Currently, the feature selection process is done by computing the set of the most frequent terms which frequency amount is over a given threshold. We had empirically found that best results can be reached by using a threshold of approximately 30% of the frequency amount.

The relevant terms selected from e-mails are represented in the EIRN network as nodes while the messages are interpreted as a collection of weighted associations with term-nodes. The instance retrieval is carried out by projecting the terms selected from the target problem over the network nodes [8]. The set of messages sharing the maximum number of features with the actual target e-mail are selected as the closest e-mails. Finally, these messages are sorted keeping in mind the frequencies of each shared term between the retrieved e-mails and the target message.

As we can see from Figure 1, when a new e-mail arrives, the EIRN network can quickly retrieve the most similar messages stored in the instance base. Then, in the reuse stage, a preliminary solution is generated by using a unanimous voting strategy with all the retrieved e-mails in the previous stage. Finally, meta-rules extracted from e-mail headers are used in order to complete the revise stage.

Our EIRN model can effectively tackle with concept drift problem by using a term confidence metric associated with each node (represented as a color between red and green in Figure 1).

We are currently working on improving the feature selection method used by our SpamHunting system in two ways: (*i*) taking into account the background ideas from the current feature selection methods which has been successfully used for the spam labeling and filtering domain and (*ii*) handling concept drift problem. In this work, we analyze how can the feature selection method of SpamHunting system be improved in order to achieve better results.

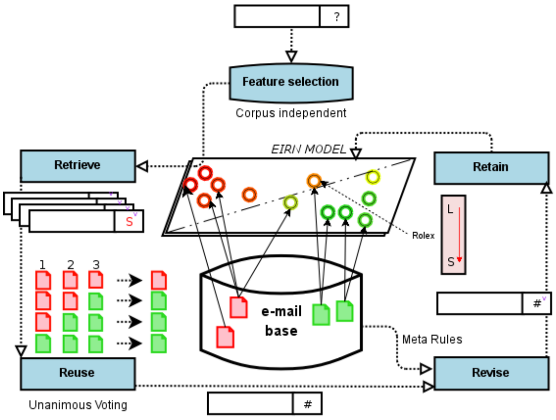


Fig. 1. SpamHunting model architecture

3 Preprocessing of Available Data and Message Representation

In this section we present several evaluation decisions related with feature selection, message representation and ready for use corpus. The following subsections are structured as follows: Subsection 3.1 contains relevant aspects relative to the corpus publicly available, Subsection 3.2 describes the main feature selection methods available in the context of spam filtering. Finally, Subsection 3.3 presents some message representation issues.

3.1 Benchmark Corpus

It is essential to provide content-based filters with an appropriate corpus of e-mails for training and testing purposes. The corpus should contain both spam and legitimate messages. Research on text categorization has been benefited significantly from the existence of publicly available, manually categorized document collections like the Reuters corpora [32], which has been used as standard benchmark. Producing similar corpora for anti-spam filtering is more complicated because of privacy issues. Publicizing spam messages does not pose a problem, since spam messages are distributed blindly to very large numbers of recipients and, hence, they are effectively already publicly available. Legitimate messages, however, in general cannot be released without violating the privacy of their recipients and senders. One way to bypass privacy problems is to experiment with legitimate messages collected from freely accessible news-groups, or mailing lists with public archives.

There are several publicly available corpora of e-mails just as LingSpam, JunkE-mail or PU. In our work, we use the SpamAssassin corpus that contains 2381 spam and 6951 legitimate messages. Legitimate e-mails have been collected from public fore or donated by users with the understanding that they may be made public. SpamAssassin has the disadvantage that its legitimate messages are not indicative of the legitimate messages that would arrive at the mailbox of a single user. Many of the legitimate messages that a user receives contain terminology reflecting his/her profession, interests, etc. that is rare in spam messages, and part of the success of personal learning-based filters is due to the fact that they learn to identify this user-specific terminology. In a concatenation of legitimate messages from different users this user-specific terminology becomes harder to identify. Hence, the performance of a learning-based filter on the SpamAssassin corpus may be an under-estimate of the performance that a personal filter can achieve.

3.2 Preprocessing and Feature Selection

An aspect that helps to improve the classification accuracy of filtering software are the preprocessing steps applied to the training and test corpus. Specially, a relevant issue in natural language processing problems is the tokenizing scheme. In our case, we consider certain punctuations like exclamation points as part of a term since spammers tend to use phrases like '*FREE!!!*'. In addition, all words were converted to lowercase in order to reduce the vocabulary and after that, those terms having smaller semantic contribution are eliminated by using the stopword list given by [21].

Typically, terms are strings of characters where stemming could be applied. Stemming lowers the size of the feature vector but it may be the case that certain forms of a word (such as the active tense) may be important in classification, so we do not stem words here. If the message representation scheme includes all the identified features in the training corpus, then very high-dimensional feature spaces would be generated. Several authors have noted the need for feature selection in order to make possible the use of conventional ML techniques to improve generalization accuracy and to avoid over-fitting the models [13].

The most widely used filter consists on calculating the *Information Gain* (IG) [14] of each term *t*. IG measures the number of bits of information obtained for category prediction (legitimate and spam) by knowing the presence or absence of a feature in a message. Subsequently, those terms whose value of IG overcomes a certain threshold are selected. Another mechanisms that allow approximating the ideal number of terms can be employed. It is the case of the *Document Frequency* (DF) [13], M*utual Information* (MI) [15, 16] or the χ*2-test* (CHI2) [13].

DF stands for the number of e-mails in which a feature occurs. We can compute the DF for each unique term in the training corpus and remove from the feature space those terms whose DF is less than some predetermined threshold. The basic assumption is that rare terms are either non-informative for category prediction, or not influential in global performance. Although DF is the simplest technique for vocabulary reduction, it is usually considered an *ad hoc* approach to improve efficiency, not a principled criterion for selecting predictive features [13].

CHI2 measures the lack of independence between a term *t* and a category *c*. Just like MI, we can compute for each category the χ^2 *statistic* using a two-way contingency table. IG, CHI2 and DF metrics are the most effective aggressive feature removal methods in the context of text categorization while MI has lower performance due to a bias favouring rare terms and a strong sensitivity to probability estimation errors [13]. Therefore, DF could be used to replace IG and CHI2 when the computation (quadratic) of these measures is too expensive.

3.3 Message Representation

Another relevant issue is the internal structure of the messages used by the different models during training and classification stages. In learning algorithms, training messages are usually represented as a vector of weighted terms like the vector space model in information retrieval [33].

Once carried out the feature extraction process over the whole corpus, the weight of terms in each message need to be calculated. The measure of the weight can be (*i*) binary (1 if the term occurs in the message, 0 otherwise), (*ii*) the *term frequency* (TF) representing the number of times the term occurs in the message, or (*iii*) TF.IDF where IDF means *Inverse Document Frequency* denoting those terms that are common across the messages of the training collection [33]. It is more normal in text classification for lexical features to carry frequency information, but previous evaluations showed that a binary representation works better in this domain [9, 18, 33, 34].

Term frequency is used in order to carry out our experiments with all models except ECUE because, as its author defines it, this model represents each message in a memory structure, which is only able to apply a binary scheme.

4 Experimental Results

The first goal of our experiments is a comparative study of the above feature selection methods (IG, DF, MI and CHI2) when they are applied to the anti-spam filtering domain. Moreover, experimental results will also be useful to improve the feature selection method of our SpamHunting IBR model and they will allow us to know the behaviour of the system in presence of noise data. Finally, these experiments will be useful for giving us an advice about the convenience of parsing e-mail attachments. The experiments have been carried out using implementations of Naïve Bayes, Adaboost, SVM and the ECUE CBR system.

The tests have been carried out for two different scenarios. In the first one, we only consider the features extracted from the subject and the body of the e-mails. Later, we append the features from the attachments. In order to handle the diverse formats of the attached files, we use different techniques for each case, taking into account the 'content-type' header information. So, HTML code was translated into text/plain using HTMLParser tool, images were processed using the Asprise OCR software and the text inside pdf documents was extracted using the PDFBox package.

Six well-known metrics proposed by Androutsopoulos *et al.* [4] have been used in order to evaluate the performance of all the analyzed models: percentage of correct classifications (%OK), percentage of False Positives (%FP), percentage of False Negatives (%FN), spam *recall*, spam *precision* and Total Cost Ratio (TCR). All the experiments have been carried out using a 10-fold stratified cross-validation [35] in order to increase the confidence level of the results obtained.

4.1 Benchmark of the Different Configurations

In this subsection, results from evaluation of the different scenarios using %OK, %FP, %FN, recall and precision metrics are showed and discussed. In this sense, Tables 1 and 2 summarize the results obtained for the evaluated models with and without attachments (w-Att and w/o-Att respectively). In square brackets it is indicated the number of selected features for each tested technique. For this experiment we have selected the best performance model of each approach varying between 100 and 2000 representing features.

Table 1. Mean value of correct classifications, FPs and FNs with 10 fold-cross validation

	NB [1000]		**AB [700]**		**SVM [2000]**		**ECUE [700]**	
	w/o-Att.	w-Att.	w/o-Att.	w-Att.	w/o-Att.	w-Att.	w/o-Att.	w-Att.
%OK								
IG	849,6	842,8	885,1	885,3	919,1	921,2	893,0	897,4
DF	849,0	842,7	882,9	878,3	920,7	921,0	863,6	873,9
MI	694,6	695,2	695,1	695,1	--	--	696,6	696,0
CHI2	846,8	839,3	887,8	884,5	918,9	919,9	888,7	894,4
%FP								
IG	48,6	60,9	13,4	12,3	8,7	5,7	6,1	7,9
DF	48,8	57,9	13,0	12,3	7,0	6,9	5,6	7
MI	1,8	2,1	0,0	0,0	--	--	46,9	40,3
CHI2	52,2	64,3	11,8	12,4	8,5	5,8	10,9	12,2
%FN								
IG	35,0	29,5	34,7	35,6	5,4	6,3	34,1	27,9
DF	35,4	32,6	37,3	42,6	5,5	5,3	64,0	52,3
MI	236,8	235,9	238,1	238,1	--	--	189,7	196,9
CHI2	34,2	29,6	33,6	36,3	5,8	7,5	33,6	26,6

Results in Tables 1 and 2 show that, despite IG has the smallest fail amount, IG, DF and CHI2 have similar effects on the performance of the evaluated classifiers. So, the mean value of correct classifications, FPs (legitimate messages classified as spam) and FNs (spam messages classified as legitimate) are practically the same with any above selection feature measures. These results support previous studies carried out with another classification models when they are applied in text categorization [13].

On the other hand, although MI method achieve the smallest amount of FPs, its fail amount is about 7 times greater than other methods. Therefore, MI method is clearly the worst feature selection method because it is very noise sensitive. If MI method is excluded, then DF achieves the smallest number of FPs.

As it can be seen from Table 1 and 2, SVM model was unable to transform the input space into a new and linearly separable one when using MI method (marked as '--'). Therefore, using MI for spam filtering and labelling is generally a bad idea.

Table 2. Averaged recall and precision scores over 10 fold-cross validation

	NB [1000]		**AB [700]**		**SVM [2000]**		**ECUE [700]**	
	w/o-Att.	w-Att.	w/o-Att.	w-Att.	w/o-Att.	w-Att.	w/o-Att.	w-Att.
Recall								
IG	0,8530	0,8761	0,8543	0,8505	0,9773	0,9735	0,8568	0,8828
DF	0,8513	0,8631	0,8433	0,8211	0,9769	0,9777	0,7312	0,7803
MI	0,0055	0,0092	0,0000	0,0000	--	--	0,2033	0,1730
CHI2	0,8564	0,8757	0,8589	0,8475	0,9756	0,9685	0,8589	0,8883
Precision								
IG	0,8071	0,7745	0,9385	0,9431	0,9643	0,9762	0,9711	0,9639
DF	0,8063	0,7807	0,9394	0,9412	0,9710	0,9713	0,9689	0,9640
MI	0,0000	0,0000	0,0000	0,0000	--	--	0,5085	0,5059
CHI2	0,7965	0,7648	0,9457	0,9423	0,9650	0,9756	0,9495	0,9456

4.2 Statistical Analysis of Benchmarking Results

After showing empirical results and several preliminary comments about them, we are going to analyse the generated outcome from a statistical point of view in order to check for the importance of the findings. In this sense, for each analysed algorithm the Cochran Q test shows differences between the proportions of failure or correct decision for all the variants, that is to say, depending on the criterion for selecting relevant terms (CHI2, IG, MI and DF) and the decision of taking or not into account e-mails attachments (w-Att and w/o-Att, respectively).

Table 3. Kappa coefficients of agreement beetween different configurations of the selected methods

		TC	w/o-Att			w-Att		
			AB-χ^2	AB-IG	AB-DF	AB-χ^2	AB-IG	AB-DF
	TC	**1**	**0,87**	**0,86**	**0,85**	**0,86**	**0,86**	**0,84**
w/o -Att	AB-χ^2		1	0,98	0,97	0,91	0,91	0,89
	AB-IG			1	0,97	0,90	0,90	0,89
	AB-DF				1	0,89	0,90	0,89
w- Att	AB-χ^2					1	0,98	0,95
	AB-IG						1	0,95
	AB-DF							1

		TC	w/o-Att			w-Att		
			SVM-χ^2	SVM-IG	SVM-DF	SVM-χ^2	SVM-IG	SVM-DF
	TC	**1**	**0,96**	**0,96**	**0,96**	**0,96**	**0,97**	**0,97**
w/o -Att	SVM-χ^2		1	0,99	0,97	0,97	0,97	0,96
	SVM-IG			1	0,97	0,97	0,97	0,97
	SVM-DF				1	0,97	0,97	0,98
w- Att	SVM-χ^2					1	0,98	0,97
	SVM-IG						1	0,97
	SVM-DF							1

		TC	w/o-Att			w-Att		
			NB-χ^2	NB-IG	NB-DF	NB-χ^2	NB-IG	NB-DF
	TC	**1**	**0,76**	**0,77**	**0,77**	**0,75**	**0,76**	**0,75**
w/o-Att	NB-χ^2		1	0,98	0,97	0,89	0,89	0,89
	NB-IG			1	0,97	0,87	0,88	0,87
	NB-DF				1	0,88	0,88	0,88
w-Att	NB-χ^2					1	0,99	0,97
	NB-IG						1	0,97
	NB-DF							1

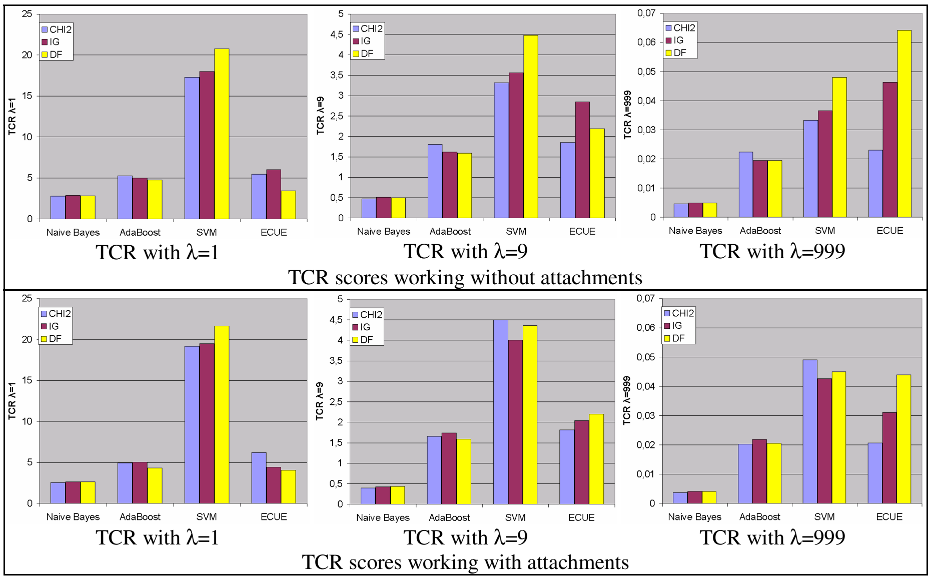
		TC	w/o-Att			w-Att		
			ECUE-χ^2	ECUE-IG	ECUE-DF	ECUE-χ^2	ECUE-IG	ECUE-DF
	TC	**1**	**0,88**	**0,87**	**0,78**	**0,88**	**0,89**	**0,81**
w/o-Att	ECUE-χ^2		1	0,94	0,80	0,88	0,87	0,81
	ECUE-IG			1	0,83	0,89	0,90	0,84
	ECUE-DF				1	0,79	0,79	0,86
w-Att	ECUE-χ^2					1	0,97	0,84
	ECUE-IG						1	0,86
	ECUE-DF							1

Given that the MI criterion gives the worst results for all the algorithms, we are interested in the analysis of the rest of factors. We consider each variant as a method, which gives us its opinion about the classification of a message. Now we are interested in knowing the degree of agreement between these methods. In order to do this, we use the Kappa test. Formally, the Kappa test compares the outcome between two methods when the observations are measured on a categorical scale. Both methods must rate the same cases using the same categorical scale to be considered as equivalent. Table 3 shows the Kappa coefficient of agreement between all the possibilities ({CHI2, IG and DF}×{w/o-Att, w-Att}) and the *true class* method (which always gives the true class of the e-mail). The comparison of the Kappa coefficient agreement of each variant with the true class algorithm gives us an idea of the accuracy of each method (bold values in Table 3). The statistical analysis of these values ranks the SVM as the best algorithm, positioning the AB and ECUE methods in second place without significant differences and the NB algorithm in last position.

In order to analyse the impact of each factor for several configurations of the algorithms, the Kappa coefficients of agreement has been statistically analysed by means of a comparison of these indexes by each one of the considered factors. The non-normality of data determines the use of a Kruskal-Wallis test in order to test the null hypothesis (no difference in the Kappa coefficients depending of each factor).

4.3 Analysis of the TCR Scores

In this subsection, TCR scores are calculated and discussed in order to confirm findings reflected in previous subsections. TCR provides a measurement of the spam filter effectiveness keeping in mind the fact that FP errors are more serious that FN. The λ parameter indicates how is the cost of an FP error in relation to an FN.



TCR with λ=1 TCR with λ=9 TCR with λ=999

TCR scores working without attachments

TCR with λ=1 TCR with λ=9 TCR with λ=999

TCR scores working with attachments

Fig. 2. TCR benchmark considering MI, CHI2 and DF feature selection methods

Figure 2 shows a benchmark of the feature selection methods within different scenarios and models using the TCR score. From results, we can realize that models handling a better capacity of discarding irrelevant features (SVM and AB to a lesser extent) get the best score when attachments are used. Other models, such as NB and AB are not able to avoid or weight down irrelevant terms. Therefore, the SVM input space conversion really complements the use of any feature selection method.

Results from Figure 2 confirm the fact that the processing of the attachment can add irrelevant or confused features to the model input space, making the learning process more difficult. However, if the selected model performs a deeper feature analysis, it can obtain better results by taking advantage of this fact.

4.4 Improving SpamHunting Feature Selection Model

Keeping in mind the background ideas from previous experiments and the existing SpamHunting feature selection model, we had concluded that a feature weight reflecting the classifying capability of a term could be used to improve global performance in the model. Expression (1) defines our proposed measure of a term t for labelling an e-mail e, where $P(t \mid e)$ represents the frequency of the term t in the message e, $P(t \mid s, \mathbf{K})$

and *P*(*t* | *l*, **K**) stand for the document frequency of *t* in spam and legitimate messages respectively, *P*(*s*| ***K***) and *P*(*l*, ***K***) are the frequency of the spam and legitimate categories and *P*(*t*| ***K***) represents the document frequency of a term *t*. In Expression (1), the relevance of a term *t* within a document *e* (measured as *P*(*t* | *e*)) is weighted by a factor that quantifies the classifying capability of the term *t* computed by the expression inside square brackets. The proposed classifying capability metric will be near to 1 for the most representative terms of each class.

$$TC(t,e) = P(t \mid e) \cdot \left[\frac{\left| P(s,\boldsymbol{K}) \cdot P(t \mid s,\boldsymbol{K}) - P(l,\boldsymbol{K}) \cdot P(t \mid l,\boldsymbol{K}) \right|}{P(t \mid \boldsymbol{K})} \right] \tag{1}$$

Despite this metric seems to be very noise sensitive, the feature selection over each document prevents the selection of noise terms in all messages belonging the training corpus. The feature selection for each message *e* is computed by selecting the smallest set of terms from *e* having the largest TC.

Table 4 shows a comparative performance study of the SpamHunting model when applying both the original feature selection method and the improved one. Results are obtained with and without processing attachments. From Table 4 we can realize that it is possible to obtain better results by applying a feature weighting or finding a feature selection method more accurate than the one currently used in our SpamHunting system. Moreover, we can deduce that our model is able to handle noise data and that it is possible to improve results by processing e-mail attachments. In fact, a similar behaviour with the SVM model is observed when the scenario is changed.

Table 4. Averaged recall and precision scores over 10 fold-cross validation

	Original feature selection		**Improved feature selection**			**Original feature selection**		**Improved feature selection**	
metric	w/o-Att.	w-Att.	w/o-Att.	w-Att.	**Metric**	w/o-Att.	w-Att.	w/o-Att.	w-Att.
%OK	96,53%	96,30%	96,67%	96,67%	Precision	0,99	0,99	0,99	0,99
%FP	0,19%	0,17%	0,04%	0,19%	TCR λ=1	7,50	7,15	7,79	7,93
%FN	3,28%	3,52%	3,29%	3,14%	TCR λ=9	5,33	5,52	7,19	5,46
Recall	0,87	0,86	0,87	0,88	TCR λ=999	0,87	1,62	5,49	2,12

5 Conclusions and Further Work

In this article, we perform a deeper analysis of feature selection methods in the context of spam filtering using different models and preprocessing scenarios. Finally, findings are used to improve the feature selection method of our SpamHunting filtering software.

The reasons for good or bad accuracy of the evaluated feature selection methods are originated by the criteria they use to choose terms. Those methods with a good performance share the same bias. Therefore, IG, DF and CHI2 score in favour of common terms over rare terms while MI prioritises low frequency terms. As demonstrated in [13], results indicate that common terms are indeed informative for text categorization tasks. Moreover, the result obtained by DF evidences that the use of category information for feature selection does not seem to be crucial for improving accuracy.

MI method is clearly disadvantaged when it is applied to spam filtering. This is motivated because spammers obfuscate terms introducing noise into messages. They spread to include in their messages multiple sequences of special characters and to change characters inside the words for punctuation symbols to make more difficult the detention of their e-mail. This noise can be seen in the quantity of rare features (with very low frequency) included in spam e-mails. Therefore, a way of improving the results obtained by this method is to eliminate those characteristics whose appearance frequency is very low before carrying out the feature selection, just as in [4, 9, 34]. The use of MI with the SVM model is not advisable because the model will probably not be able to complete the training process.

With respect to DF, it is the simplest feature selection method, however, experimental results have shown its good performance when it is applied in spam filtering. Results achieve the smallest amount of false positives probably because a previous stopword removal has been carried out. Without this preprocessing step, the feature selection made will probably contain a lot of semantically empty words.

Talking about IG and CHI2 feature selection methods, results show that generally IG achieves a little bit better precision (security) while CHI2 is slightly superior in recall measure (effectiveness). The experiments carried out have probed that there is no significant differences between this methods regardless of the selected model.

The use of message attachments is useful when it is combined with models, which are able to discard irrelevant or confused features by performing a deeper analysis of the input space in order to detect and remove fake features. Strict methods that use all features from input space achieve poor results because several detected features in the attachment processing are confused or irrelevant. In this sense, SpamHunting feature selection method should be constructed by combining term frequency with a metric able to measure the classification capability of each term. Talking about noise data, as SpamHunting retrieval model is able to distinguish noise data and skip it, this problem can be avoided in our system. The classification capability of each term should be dynamically calculated by using all messages stored in the instance base.

Finally, in order to handle the concept drift problem, a relevant issue is how to give priority to recent e-mails detecting the change of the words representing legitimate and spam classes. Building models and techniques with the capacity of tackling with this problem is the most recent trend in spam research. In this way, several authors have explained the main ideas behind the models, which are able to manage this characteristic of spam filtering problem [21].

Talking about the future work, a new research line in spam filtering has been opened since SpamHunting model introduced the feature selection over each message (and not over the whole training corpus). This way of representing the messages is more effective than others because each document has its own relevant features, allowing continuous updating of the model knowledge and being more suitable for working with disjoint concepts (such as spam). In a first stage we will try to apply the ideas exposed in this paper to improve the feature selection method and later, we will work over the SpamHunting model in order to achieve better results.

References

1. Spam statistics. http://www.theregister.co.uk/security/spam/
2. Oard, D.W.: The state of the art in text filtering. User Modeling and User-Adapted Interaction, Vol.7, (1997) 141–178
3. Wittel, G.L., Wu, S.F.: On Attacking Statistical Spam Filters. Proc. of the First Conference on E-mail and Anti-Spam CEAS, (2004)
4. Androutsopoulos, I., Paliouras, G., Michelakis, E.: Learning to Filter Unsolicited Commercial E-Mail. Technical Report 2004/2, NCSR "Demokritos", (2004)
5. Méndez J.R., Iglesias E.L., Fdez-Riverola, F., Díaz F., Corchado, J.M.: Analyzing the Impact of Corpus Preprocessing on Anti-Spam Filtering Software. Research on Computing Science, To appear. (2005)
6. Corchado, J.M., Corchado, E.S., Aiken, J., Fyfe, C., Fdez-Riverola, F., Glez-Bedia, M.: Maximum Likelihood Hebbian Learning Based Retrieval Method for CBR Systems. Proc. of the 5th International Conference on Case-Based Reasoning, (2003) 107–121
7. Corchado, J.M., Aiken, J., Corchado, E., Lefevre, N., Smyth, T.: Quantifying the Ocean's CO2 Budget with a CoHeL-IBR System. Proc. of the 7th European Conference on Case-based Reasoning, (2004) 533–546
8. Fdez-Riverola, F., Lorenzo, E.L., Díaz, F., Méndez, J. R., Corchado, J.M.: SpamHunting: An Instance-Based Reasoning System for Spam Labelling and Filtering. Decision Support Systems, To Appear (2006)
9. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian approach to filtering junk e-mail. In Learning for Text Categorization – Papers from the AAAI Workshop, Technical Report WS-98-05, (1998) 55–62
10. Carreras, X., Màrquez, L.: Boosting trees for anti-spam e-mail filtering. Proc. of the 4th International Conference on Recent Advances in Natural Language Processing, (2001) 58–64
11. Vapnik, V.: The Nature of Statistical Learning Theory. 2nd Ed. Statistics for Engineering and Information Science, (1999)
12. Delany, S.J., Cunningham P., Coyle L.: An Assessment of Case-base Reasoning for Spam Filtering. Proc. of Fifteenth Irish Conference on Artificial Intelligence and Cognitive Science: AICS-04, (2004) 9–18
13. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. Proc. of the Fourteenth International Conference on Machine Learning: ICML-97, (1997) 412–420
14. Mitchell, T.: Machine Learning. Mc Graw Hill, (1996)
15. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. Proc. of the 7th International Conference on Information and Knowledge Management, (1998) 229–237
16. Church, K.W., Hanks, P.: Word association norms, mutual information and lexicography. Proc. of the ACL, Vol.27, (1989) 76–83
17. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys, Vol. 34 (1). (2002) 1–47
18. Drucker, H.D., Wu, D., Vapnik, V.: Support Vector Machines for spam categorization. IEEE Transactions on Neural Networks, Vol. 10 (5). (1999) 1048–1054
19. Platt, J.: Fast training of Support Vector Machines using Sequential Minimal Optimization. In Sholkopf, B., Burges, C., Smola, A. (eds.). Advances in Kernel Methods – Support Vector Learning, (1999) 185–208

20. Schapire, R.E., Singer, Y.: BoosTexter: a boosting-based system for text categorization. Machine Learning, Vol. 39 (2/3), (2000) 135–168
21. Tsymbal, A.: The problem of concept drift: definitions and related work, Available at http://www.cs.tcd.ie
22. Graham, P.: Better Bayesian filtering. Proc. of the MIT Spam Conference, 2003
23. Kolcz A., Alspector, J.: SVM-based filtering of e-mail spam with content specific misclassification costs. Proc. of the ICDM Workshop on Text Mining, (2001)
24. Hovold, J.: Naïve Bayes Spam Filtering Using Word-Position-Based Attributes. Proc. of the Second Conference on Email and Anti-Spam CEAS-2005. http://www.ceas.cc/papers-2005/144.pdf
25. Gama, J., Castillo, G.: Adaptive Bayes. Proc. of the 8th Ibero-American Conference on AI: IBERAMIA-02, (2002) 765–774
26. Scholz, M., Klinkenberg, R.: An Ensemble Classifier for Drifting Concepts. Proc. of the Second International Workshop on Knowledge Discovery from Data Streams, (2005) 53–64
27. Syed, N. A., Liu H., Sung. K.K.: Handling Concept Drifts in Incremental Learning with Support Vector Machines. Proc. of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, (1999) 317–321
28. Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C.D., Stamatopoulos, P.: Learning to Filter Spam E-Mail: A Comparison of a Naïve Bayesian and a Memory-Based Approach. In Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD) (2000) 1–13
29. Daelemans, W., Jakub, Z., Sloot, K., Bosh, A.: TiMBL. Tilburg Memory Based Learning, version 5.1, Reference Guide. ILK, Computational Linguistics, Tilburg University. http://ilk.uvt.nl/software.html#timbl
30. Lenz, M., Auriol, E., Manago, M.: Diagnosis and Decision Support. Case-Based Reasoning Technology. Lecture Notes in Artificial Intelligence, Vol. 1400, (1998) 51–90
31. Frakes, B., Baeza-Yates, R.: Information Retrieval: Data Structures & Algorithms. Prentice-Hall, (2000)
32. NIST: National Institute of Science and Technology. Reuters corpora. (2004), http://trec.nist.gov/data/reuters/reuters.html
33. Salton, G., McGill, M.: Introduction to modern information retrieval, McGraw-Hill, (1983)
34. Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C., Stamatopoulos, P. A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists. Information Retrieval, Vol. 6 (1). (2003) 49–73
35. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. Proc. of the 14th International Joint Conference on Artificial Intelligence: IJCAI-95, (1995) 1137–1143