Tracking Concept Drift at Feature Selection Stage in SpamHunting: An Anti-spam Instance-Based Reasoning System

J.R. Méndez[1], F. Fdez-Riverola[1], E.L. Iglesias[1], F. Díaz[2], 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 propose a novel feature selection method able to handle concept drift problems in spam filtering domain. The proposed technique is applied to a previous successful instance-based reasoning e-mail filtering system called SpamHunting. Our *achieved information* criterion is based on several ideas extracted from the well-known information measure introduced by Shannon. We show how results obtained by our previous system in combination with the improved feature selection method outperforms classical machine learning techniques and other well-known lazy learning approaches. In order to evaluate the performance of all the analysed models, we employ two different corpus and six well-known metrics in various scenarios.

1 Introduction and Motivation

Internet has introduced a revolutionary way for communication issues. Some daily activities such as news reading or message sending has been innovated and facilitated. Now, an Internet user can send an e-mail through thousands of kilometres with no cost. Unfortunately, some people and companies with doubtful reputation had quickly discovered how to take advantage of this new technology for advertising purposes. Since then, they are constantly sending a lot of advertisement messages known as spam e-mails. These messages are damaging the rights of Internet users because they are paying the transfer costs of the spam messages. Moreover, spam collapses networks, routers and information servers belonging to the Internet Service Providers (ISPs) generating high costs and damages.

Although some legal actions have been introduced for combating the delivery of spam messages, at the moment anti-spam filtering software seems to be the most viable solution to the spam problem. Spam filtering software is often classified as *collaborative* or *content-based* [1]. In the context of collaborative systems, the message filtering is carried out by using judgements made by other users [2]. Although there is no doubt

T.R. Roth-Berghofer et al. (Eds.): ECCBR 2006, LNAI 4106, pp. 504 – 518, 2006.

that collaborative techniques can be useful to spam filtering, systems able to analyse in detail the intrinsic properties of the message (subject, body contents, structure, etc.) have a better chance of detecting new spam messages [3]. These approaches are included within the content-based approach and are studied in this work.

The main types of content-based techniques are machine learning (ML) algorithms and memory and case-based reasoning approaches. In ML techniques an algorithm is used to 'learn' how to classify new messages from a set of training e-mails. On the other hand, memory and case-based reasoning techniques store all training instances in a memory structure and try to classify new messages by finding similar e-mails to it. Hence, the decision of how to solve a problem is deferred until the last moment. Although ML algorithms have been successfully applied in the text classification field, recent research work has shown that case-based reasoning (CBR) and instance-based reasoning (IBR) systems are more suitable for the spam filtering domain [4, 5, 6].

In this paper we propose and analyse an enhancement over a previous successful anti-spam IBR-based system called SpamHunting [6]. The main objective is to discuss and test a new improvement over knowledge representation in SpamHunting to show the importance of instance representation in CBR/IBR approaches. Results obtained by different well-known spam filtering models and those obtained by our new approach are shown for benchmarking purposes. The models selected to carry out the evaluation are Naïve Bayes [7], boosting trees [8], Support Vector Machines [9], and two case-based systems for spam filtering that can learn dynamically: a Cunningham *et al.* system which we call *Odds-Ratio CBR* [4] and its improved version named *ECUE* [5]. Experiments have been carried out using two different well-known public corpora of e-mails and taking into account several measures in order to represent different points of view.

We are also interested in achieving new findings about the role of feature selection process when using CBR/IBR approaches on the spam filtering domain. Specially, our aim is centred in handling the concept drift problem [4] (inherent in the spam filtering domain) at this early stage. In this work we are showing the dynamical adaptation capacities of SpamHunting when the environment changes. We also describe in detail the role of the feature selection preprocessing step in this kind of situations.

The rest of the paper is structured as follows: section 2 we outline machine learning and case-based e-mail filters mentioned above. In section 3 the SpamHunting IBR architecture is described in detail while in section 4 we present our improved feature selection method for our previous SpamHunting system. Section 5 contains a description of some relevant issues belonging to the experimental setup while section 6 is focused in showing the empirical results obtained by the different models. Finally, in section 7 we expose the main conclusions reached as well as the future lines of our research work.

2 Spam Filtering Techniques

This section contains a brief description of the popular Spam filtering techniques. The following subsections are structured as follows: Subsection 2.1 contains a short introduction to classical ML models that has been successfully applied to the spam filtering domain. Subsection 2.2 is focused in summarizing newest models proposed in the most recent research work.

2.1 ML Classical Approaches

There is no doubt regarding the similarities of text categorization and the spam filtering domain. In fact, both research fields are included into the document automatic classification domain belonging to the Natural Language Processing (NLP) area. Both are based on distributing a collection of documents (or corpus) into several classes or categories. However, we should note that spam and legitimate classes are generally more imprecise, internally disjointed and user-dependant than text categories [1]. Moreover, there are some additional problems in the spam filtering domain such as noise level and concept drift [5, 10].

Due to the related similarity between text categorization and spam filtering domains, several commonly used models and techniques from the former have been successfully applied on the later. The traditional Bayesian method is a clear example of this issue. This kind of spam filters are based on computing the probability of a target message being spam taking into account the probability of finding its terms in Spam e-mails. If some words of the target message are often included in Spam messages but not in legitimate ones. Then it would be reasonable to assume that target e-mail is more likely to be Spam. Although there are several Bayesian approaches, it is Naïve Bayes that is widely used for Spam filtering [7].

Besides Bayesian models, Support Vector Machines (SVM) and boosting techniques are also well-known ML algorithms used in this domain [11]. SVMs [9] have become very popular in the machine learning and data mining communities due to its good generalization performance and its ability to handle high-dimensional data through the use of kernels. They are based on representing e-mails as points in an *n*-dimensional space and finding a hyperplane that generates the largest margin between the data points in the positive class and those in the negative class. Some implementations of SVM can be found in ML environments such as Waikato Environment for Knowledge Analysis[1] (WEKA) or Yet Another Learning Environment[2] (YALE). Particularly, WEKA includes the *Sequential Minimal Optimization* (SMO) algorithm that has demonstrated a good trade-off between accuracy and speed (see [12] for details).

Boosting techniques [8] classify a target e-mail by combining and weighting the outputs of several weak learners when they are applied over a new message. Weak learners are simple classification algorithms that can learn with an error rate slightly lower than 50%. Several boosting algorithms have been introduced for classification. Of these the AdaBoost algorithm [13] is commonly used.

2.2 Recent Trends in the Spam Filtering Domain

Recently, several new ML models have been introduced for e-mail classification such as Chung-Kwei [14], which is based on pattern-discovery. In this sense, recent research work are focused on improving or adapting current classification models used in spam filtering domain. In this sense, two improvements over Bayesian filtering are proposed in [15, 16] while in [17] Hovold presents an enhancement over SVM model enabling misclassification costs. Keeping in mind the continuous update of the knowledge and the concept drift problem, an incremental adaptive Bayesian learner is

[1] WEKA is available from http://www.cs.waikato.ac.nz/ml/weka/

[2] YALE is available from http://yale.sourceforge.net

presented in [18] while in [19, 20] an ensemble classifier able to track concept drift and a SVM enhancement for support this problem are proposed respectively. However, we highlight advances achieved by using CBR systems as they have started a revolution in Spam filtering applications.

Case-based approaches outperform classical machine learning techniques in anti-spam filtering because they work well for disjoint sub-concepts of the spam concept (spam about *porn* has little in common with spam offering *rolex*) whereas classical ML techniques try to learn an unified concept description [5]. Another important advantage of this approach is its ease of updating to tackle the concept drift problem in the anti-spam domain [21].

Cunningham *et al.* have proposed in [5] a successful case-based system for anti-spam filtering that can learn dynamically. The system (which we call *Odds-Ratio CBR*) uses a similarity retrieval algorithm based on Case Retrieval Nets (CRN) [22]. CRN networks are equivalent to the *k*-nearest neighbourhood algorithm but are computationally more efficient in domains where there is feature-value redundancy and missing features in cases, as spam. This classifier uses a unanimous voting technique to determine whether a new e-mail is spam or not. All the returned neighbours need to be classified as spam e-mails in order to classify the new e-mail as Spam.

In the work of Delany *et al.* [4], it is presented the ECUE system (*E-mail Classification Using Examples*) as an evolution from *Odds-Ratio CBR* preceding model. While the previous system uses an odds ratio method for feature selection, the ECUE model uses Information Gain (IG) [23].

Recently, a successful spam filtering IBR model called SpamHunting has been proposed [6]. The main characteristics and the model operation of this system are briefly outlined in the next section.

3 SpamHunting IBR System

The SpamHunting system is a lazy learning hybrid model based on an instance-based reasoning approach able to solve the problem of spam labelling and filtering [6]. This system incorporates an Enhanced Instance Retrieval Network (EIRN) model, which is able to index e-mails in an effective way for efficient retrieval.

Figure 1 presents the SpamHunting model architecture. As it shows, an instance representation stage is needed in order to correctly classify an incoming e-mail. In this step a message descriptor should be generated. This message descriptor consists of a sequence of N features that better summarize the information contained in the e-mail. For this purpose, we use data from two main sources: (*i*) information obtained from the header of the e-mail (see Table 1) and (*ii*) those terms that are more representative of the subject, body and attachments of the message.

In order to gather additional information, the pdf files, images and HTML documents attached to the e-mail are processed and converted to text. This text and the e-mail body are tokenised together 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 [24].

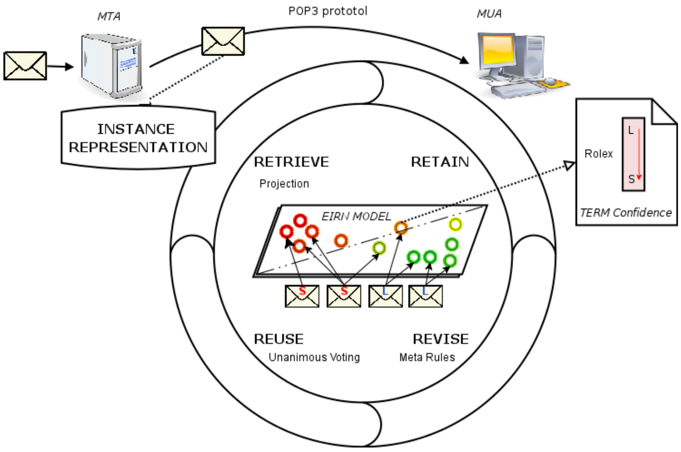

Fig. 1. SpamHunting model architecture

The selection of the best representative terms is carried out in an independent way for each training and testing e-mail. Therefore, each message has its own relevant features. The term selection process is done by computing the set of the most frequent terms which frequency amount is over a given threshold [6]. We have empirically found that best results are obtained by using a threshold of approximately 30% of the frequency amount.

Table 1. Representation of header features stored in the instance-descriptor of SpamHunting system

Variable	**Type**	**Description**
From	String	Source mailbox
Return Path	String	Indicates the address used for reply purposes
Date	Date	Delivery date
Language	String	Tongue of the language
Attached Files	Integer	Number of attached files
Content Type	String	MIME type

As Figure 1 shows, the relevant terms selected from the messages are represented in the EIRN network as term-nodes, while the instances are interpreted as a collection of weighted associations with term-nodes. The instance retrieval is carried out by projecting the selected terms from the target problem over the network nodes [6]. The set of messages sharing the maximum number of features with the actual target e-mail is 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.

The EIRN network is able to store some useful information about how words are affected by concept drift. In order to support this feature, a confidence measurement for each term-node is computed and saved. Expression (1) defines the confidence of a term w_i using the current knowledge $\boldsymbol{K}$ where $P(w_i \mid S, \boldsymbol{K})$ and $P(w_i \mid L, \boldsymbol{K})$ stands for the probability of finding the term w_i in spam and legitimate messages from $\boldsymbol{K}$ respectively.

$$c\left(w_i,\boldsymbol{K}\right)=\frac{\left|P\left(w_i \mid S,\boldsymbol{K}\right)-P\left(w_i \mid L,\boldsymbol{K}\right)\right|}{P\left(w_i \mid S,\boldsymbol{K}\right)+P\left(w_i \mid L,\boldsymbol{K}\right)} \tag{1}$$

In the reuse stage, using a unanimous voting strategy taking into account all the retrieved e-mails in the previous phase generates a preliminary solution. This approach has been previously used in other successful spam filtering CBR systems [4, 5].

The revise stage is only carried out when the assigned class is spam and it entails the utilisation of meta-rules extracted from e-mail headers. This re-evaluation is performed with the goal of guaranteeing the accuracy of the proposed solution.

Finally, when the classification process has been completed, a new instance message containing the instance-descriptor and the solution (assigned class) is constructed and stored in the instance base for future reuse. During this stage, confidence level of term-nodes affected by the message indexing should be recalculated in order to adequately track concept drift effects. A detailed description of the model operation can be found in [6].

4 SpamHunting Feature Selection Improvement

In this section, an improvement for the SpamHunting relevant terms selection algorithm can be found. A detailed explanation about the underground ideas behind our proposal and its main abilities are contained below.

A relevant issue related to the context of Artificial Intelligence is the need for adequately knowledge representation. Problem solving gets easier when a suitable knowledge representation is chosen. We think that modern and classical classifier models are not sufficient to achieve accurate classification results in spam-filtering domain. In other words, if the knowledge is not perfectly represented, the classifier will not achieve accurate results [25].

Our successful SpamHunting IBR system is based on an EIRN network which has been combined with: (*i*) an original method for selecting the most relevant features in order to improve the representation of the instances and (*ii*) some mechanisms designed to adequately handle concept drift during the instance representation stage. Using SpamHunting architecture, we had achieved better results than other current classifier models and other non-improved k-nearest neighbourhood approaches [6].

Shannon has introduced the use of probabilities for measuring the information amount provided by knowing a concrete feature [26]. Keeping in mind this approach, if we are trying to identify somebody, knowing the name is more useful than having knowledge about sex. This happens because the probability of finding somebody knowing the sex is lower than the probability of finding someone when name has been provided. In this context, Expression (2) is used to compute the amount of information achieved by knowing a feature X, where $P(x_i)$ stands for the probability of each event x_i between the n possible values.

$$H(X)=-\sum_{i=1}^{n}P\left(x_i\right)\log P\left(x_i\right) \tag{2}$$

From the above discussion, we can deduce the following ideas: (*i*) the word (term) length is a relevant issue for categorization and filtering because largest words are unusual in documents (the probability of finding a document knowing that it contains a long word is higher) and (*ii*) we should introduce a measurement able to estimate the usefulness of knowing whether or not a keyword *w* is present.

Afore mentioned ideas are important and should be applied to improve the selection of relevant features and consequently the instance representation. The main target goal is to maximise the information contained in an instance.

In this sense, Expression (3) defines the Achieved Information (AI) measure when a term *w* is found in a message *e* having the current knowledge ***K***. $P(w \mid e)$ represents the frequency of appearance of a word *w* in the considered message *e*, $P(w \mid S, \boldsymbol{K})$ and $P(w \mid L, \boldsymbol{K})$ are the frequencies of finding the word *w* in the current spam and legitimate stored instances (***K***) respectively, and finally, *length*(*w*) measures the number of characters of the word *w*.

$$AI(w,e \mid \boldsymbol{K}) = P(w \mid e) \cdot \left[1 - \frac{1}{length(w)}\right] \cdot \left[\frac{\left|P(w \mid S,\boldsymbol{K}) - P(w \mid L,\boldsymbol{K})\right|}{P(w \mid S,\boldsymbol{K}) + P(w \mid L,\boldsymbol{K})}\right] \tag{3}$$

We highlight the importance of including variable ***K*** designed for addressing concept drift. In the presence of concept drift, some terms affected by the passage of time can loose its capacity of correctly classifying messages. Therefore, the measurement of this capacity for each word should not be previously calculated using only the training corpus. It must be computed when the target message arrives using all available knowledge at this time.

When a word *w* is not present in any instance stored in the SpamHunting instance base (***K***), the second part in square brackets belonging to Expression (3) will be replaced with 1. Therefore, when no information has been compiled about a term, we assume that it will be fully predictive. This decision prevents to stop discovering new predictive words and represents an important advance included in our SpamHunting system to handle concept drift.

The underlying idea is that the concept drift problem must be addressed at the instance representation stage. Using techniques designed for handling concept drift at this early stage can boost the accuracy of the models. As static feature selection methods (calculated before the training stage) are not able to handle concept drift in this way, we use a dynamical feature selection process.

The method proposed for selecting the most relevant terms is made by following two steps: (*i*) computing the AI measure for all words included in the target message and (*ii*) select the most helpful terms from the message having an AI amount greater than a percentage of the total AI of all terms belonging to the target e-mail.

In our forthcoming experimentation we have tested different percentage configurations varying between 20% and 65% with the aim of finding the best threshold. Finally, we have chosen 60% as it produced the best results on the related preliminary experimentation.

5 Experimental Setup

In this section we discuss several relevant decision related to the configuration of the experiments. Firstly, Subsection 5.1 contains a description of the available spam corpora for benchmarking purposes. Then, Subsection 5.2 is focused in message tokenising, preprocessing and representation issues.

Evaluation has been done by a comparative performance study of several classical ML models (Naïve Bayes, AdaBoost and SVM), two case-based reasoning approaches proposed by Cunningham (ECUE and *Odds-Ratio CBR*) and our previous successful SpamHunting IBR system (with and without applying our proposed feature selection improvement).

5.1 Available Corpus

A significant issue about experimental configuration is choosing a corpora of e-mails for benchmarking purposes. Despite privacy issues, a large number of corpus like SpamAssassin[3], Ling-Spam[4], DivMod[5], SpamBase[6] or JunkEmail[7] can be downloaded from Internet. Table 2 shows a short description of the related corpus focussing on the spam and legitimate ratio and the distribution form.

Table 2. Comparative study of the most well known corpus

Corpus	Legitimate%	Spam%	Format	Preprocessing steps applied
Ling-Spam	83.3	16.6	Tokens	Tokenised
PU1	56.2	43.8	Token Ids	Tokenised ID representation for each token
PU2	80	20	Token Ids	Tokenised ID representation for each token
PU3	51	49	Token Ids	Tokenised ID representation for each token
PUA	50	50	Token Ids	Tokenised ID representation for each token
SpamAssassin	84.9	15.1	RFC 822	Not preprocessed
Spambase	39.4	60.6	Feature Vectors	Tokenised Feature selection
Junk-Email	0	100	XML	Not preprocessed
Bruce Guenter	0	100	RFC 822	Not preprocessed
DivMod	0	100	RFC 822	Not preprocessed

In this work, we are using the SpamAssassin and Ling-Spam corpus. The former comprises 9332 messages from January 2002 up to and including December 2003. The later contains 2412 previously tokenised messages without any date information. Although these corpuses seem old, the Spam problem remains the same. We have used them since they are the most widely used public corpora in spam filtering domain.

[3] Available at http://www.spamassassin.org/publiccorpus/

[4] Available at http://www.iit.demokritos.gr/

[5] Available at http://www.divmod.org/cvs/corpus/spam/

[6] Available at http://www.ics.uci.edu/~mlearn/MLRepository.html

[7] Available at http://clg.wlv.ac.uk/projects/junk-e-mail/

5.2 Message Representation Issues

A relevant question in models applied to spam filtering is the internal structure of the messages used during the training and the classification stages. The knowledge representation is different in classical ML techniques and CBR/IBR models.

In the context of the classical spam filtering ML models, messages are usually represented as a vector $\vec{t} = \left\langle t_1, t_2, ..., t_p \right\rangle$ containing numerical values that represent certain message features. When we use this form of model, messages must be represented with the same features. The selected features are often representing the presence or absence of a term in the message. This idea has been inherited from the vector space model in information retrieval [24, 27].

CBR/IBR systems use a memory structure able to store all messages in the form of cases or instances. This structure is optimised to quickly carry out the retrieval stage (given a target problem, recover cases from memory that are relevant to solving it). As with SpamHunting, this kind of systems is able to work when messages are represented with distinct feature measurements.

A significant topic for message representation is feature extraction (identifying all possible features contained in a message). Feature identification can be performed by using a variety of generic lexical tools, generally by tokenising the text extracted from e-mails into words. At first glance, it seems to be a simple tokenising task guided by several characters as word separators. However, at least the following particular cases have to be considered with care: hyphens, punctuation marks and the case of the letters (lower and upper case) [25]. In the spam domain, punctuation marks and hyphenated words are among the best discriminating attributes in a corpus, because they are more common in spam messages than legitimate ones.

In our experimentation, text for tokenising was extracted from e-mail body and attachments. In order to process diverse formats of the attached files, we use different techniques in each case taking into account the “content-type” header information. So, HTML code was translated into text/plain using the HTMLParser[8] tool, images were processed using the Asprise OCR[9] software and the text inside pdf documents was extracted using the PDFBox[10] package. We tokenised the text extracted from e-mails using only blank spaces in order to preserve the original aspect of the words belonging to each message and finally, all identified words were converted to lower case.

When the tokenising step has been completed, stopword removal (which drop articles, connectives and other words without semantic content) and/or a stemming (which reduces distinct words to their common grammatical root) can be applied to identified tokens [24]. In our experiments we have used only stopword removal as it has shown to be the best choice for the majority of systems [25].

Once carried out the lexical analysis over the training corpus, a large number of features would probably have been identified. In our experimentation we use a feature selection method to select the most predictive ones. *Information Gain* (IG), *Mutual Information* (MI) and the χ^2 statistic are well-known methods used for aggressive feature removal in text categorization domain [23]. From them, we had chosen the IG

[8] HTMLParser is available for download at http://htmlparser.sourceforge.net/

[9] Asprise OCR can be downloaded at http://asprise.com/product/ocr/

[10] PDFBox is available for download at http://www.pdfbox.org/

method to select the most predictive features as it has been successfully used for feature removal in several spam filtering research works [3, 4]. This method is based on computing the IG measure for each identified feature by using the equation given in Expression (4) and selecting those terms having the highest computed value.

$$IG(t) = \sum_{c \in \langle l,s \rangle} P(t \wedge c) \cdot \log \frac{P(t \wedge c)}{P(t) \cdot P(c)} \tag{4}$$

We have kept the original feature selection method used by the *Odds-Ratio CBR* model based on computing an odds ratio measurement. Moreover, the number of selected features for each message needs to be decided. For our comparisons, we have selected the best performance configuration of each classical ML technique varying between 100 and 2000 features. In order to test the *Odds-Ratio CBR*, and ECUE models we have maintained their original feature selection configurations. The first one uses 30 words for representing spam class and 30 words describing legitimate category while an IG selection of 700 features has been recommended by the authors for using ECUE CBR system.

Finally, for testing classical ML models the weight of terms in each message *e*, 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 calculated by Expression (5) or (*iii*) the *inverse document frequency* (IDF) given by Expression (6) denoting those terms that are common across the messages of the training collection.

$$t_i(e) = \frac{n_i(e)}{N(e)} \tag{5}$$

$$t_i(e) = \frac{n_i(e)}{N(e)} \log_2 \frac{m}{df(T_i)} \tag{6}$$

In Equations (5) and (6), $n_i(e)$ stands for the number of occurrences of term T_i in *e*, *N*(*e*) represents the recount of terms in *e*, *m* is the number of training messages and $df(T_i)$ stands for the number of training messages where the term T_i occurs.

A binary representation has been used for testing ML classical models. ECUE and *Odds-Ratio CBR* are also using a binary feature representation for organizing the case base by using Information Entity Nodes in a CRN Structure [4].

6 System Evaluation

Information about selected metrics and several minor details concerning the use of the different corpus for evaluation purposes are described in this section. Experimental results are also contained in Subsection 6.1.

Six well-known metrics [3] have been used in order to evaluate the performance of all the analysed models: *total cost ratio* (TCR) with three different scenarios, spam *recall*, spam *precision*, percentage of correct classifications (%OK), percentage of False Positives (%FP) and percentage of False Negatives (%FN).

Firstly, we had used the SpamAssassin corpus for analysing the improved version of the SpamHunting IBR system in action. Then, we have used the Ling-Spam corpus to demonstrate the significance of the achieved results. All experiments have been carried out using a 10-fold stratified cross-validation [28] in order to increase the confidence level of results obtained.

Finally, some details about classical ML models configuration is described. Decision Stumps [29] have been used as weak learners for AdaBoost classifier with 150 boost iterations and SVM has been tested by using a polynomial kernel.

6.1 Experimental Results

Initially, the performance of the analysed models was measured from a cost-sensitive point of view. For this purpose we compute the TCR metric in the above mentioned different situations. TCR assumes that FP errors are λ times more costly than FN errors, where λ depends on the usage scenario (see [3] for more details). 1, 9 and 999 values for the λ parameter have been used over the experiments.

Table 3 shows a TCR comparative of the analysed models when using the SpamAssassin corpus. The number of selected features used for each model is placed in square brackets. Results show that the classifications obtained by using the improved version of the SpamHunting IBR system is extremely safe and good (TCR λ=999). Moreover, the original version of SpamHunting, ECUE and *Odds-Ratio CBR* are also safer than classical ML approaches. From a different point of view, Table 3 also shows that only SVM model is able to go beyond the improved SpamHunting system in amount of correctly classified messages (TCR λ=1).

Table 3. TCR scores over 10 stratified fold-cross validation using SpamAssassin

	Model						
Metric	*Naïve Bayes* [1000]	*AdaBoost* [700]	*SVM* [2000]	*Odds-Ratio CBR* [60]	*ECUE* [700]	*Spam Hunting* [-]	*Improved Spam Hunting* [-]
TCR λ=1	2.647	5.011	22.852	1.382	6.792	7.498	12.255
TCR λ=9	0.416	1.688	5.225	1.345	2.658	5.331	9.293
TCR λ=999	0.004	0.020	0.057	0.990	0.036	0.874	6.573

Table 4. TCR scores over 10 stratified fold-cross validation using Ling-Spam

	Model						
Metric	*Naïve Bayes* [1000]	*AdaBoost* [700]	*SVM* [2000]	*Odds-Ratio CBR* [60]	*ECUE* [700]	*Spam Hunting* [-]	*Improved Spam Hunting* [-]
TCR λ=1	7.769	22.871	27.385	2.152	13.070	1.211	6.014
TCR λ=9	3.798	9.016	8.672	2.152	1.811	1.122	5.250
TCR λ=999	1.524	6.471	5.788	2.152	0.017	0.757	4.415

In order to contrast and validate the obtained results with a different corpus, Table 4 shows analysed models in action when using the Ling-Spam corpus. SVM, AdaBoost and the improved version of SpamHunting get the highest score for the relation between security (lower FP amount) and hits (correctly classified messages) (TCR λ=999). From this fact, we can realize that the improved SpamHunting system gets a higher security level independently of the selected corpus.

From a different point of view, Table 5 shows the recall and precision scores obtained for each considered experimental corpus. Analysing recall scores and keeping in mind the idea of maximizing the highest correctly classified amount, we can realize that sometimes classical models can slightly get better than the improved version of SpamHunting. However, precision scores clearly show that the improved SpamHunting IBR system always gets the best balance between correctly classified amount and security scores. The precision score achieved by using ECUE system and *Odds-Ratio CBR* model should be highlighted, as they are extremely good.

Table 5. Recall and precision scores using Ling-Spam and SpamAssassin

	SpamAssassin		**Ling-Spam**	
Measure	Recall	Precision	Recall	Precision
Naïve Bayes [1000]	0.876	0.774	0.884	0.975
AdaBoost [700]	0.850	0.943	0.954	0.977
SVM [2000]	0.974	0.976	0.971	0.973
Odds-Ratio CBR [60]	0.276	0.992	0.526	1
ECUE [700]	0.883	0.964	0.985	0.928
Spam Hunting [-]	0.862	0.992	0.177	0.942
Improved Spam Hunting [-]	0.921	0.994	0.831	0.993

Taking into consideration other measures, Table 6 shows the percentage of correct classifications, false positives and false negatives belonging to the experimental work with the seven analysed models over the defined experimental configuration and corpus. Analysing Table 6 we can see that SVM and AdaBoost algorithms usually achieve the greatest percentage of correct classifications.

Table 6. Percentage of correct classifications, FPs and FNs

	SpamAssassin			**LingSpam**		
Measure	%OK	%FP	%FN	%OK	%FP	%FN
Naïve Bayes [1000]	90.3	6.5	3.2	97.7	0.4	1.9
AdaBoost [700]	94.9	1.3	3.8	98.9	0.4	0.7
SVM [2000]	98.7	0.6	0.7	99.1	0.4	0.5
Odds-Ratio CBR [60]	81.5	0.1	18.4	92.1	0	7.9
ECUE [700]	96.2	0.8	3.0	98.5	1.3	0.2
Spam Hunting [-]	96.3	0.2	3.5	86.1	0.2	13.7
Improved Spam Hunting [-]	97.9	0.1	2.0	97.1	0.1	2.8

From a different point of view, Table 6 shows that *Odds-Ratio CBR* and all versions of SpamHunting model achieve the lowest FP error. Other models (like SVM or AdaBoost) are able to slightly increment the correctly classified messages amount but they achieve a greater number of FP errors. Finally, It is needed to highlight the FP ratio obtained using the *Odds-Ratio CBR* model over the LingSpam corpus. This fact supports the suitability of the CBR/IBR approaches to spam filtering.

7 Conclusions and Future Work

In this paper we have introduced an improvement to our previous successful SpamHunting IBR system. We have carried out a deep analysis by choosing a representative set of spam filtering models (including Naïve Bayes, AdaBoost, SVM, and two case-based systems) in order to benchmark their performance while corpus is changed.

The original and improved versions of the SpamHunting IBR system had shown to be the safest spam filtering models by obtaining a convenient ratio between the FP error and correctly classified rates. Moreover, the improved version of SpamHunting is the first model able to adequately handle concept drift at the early instance representation stage.

We highlight results obtained in both versions of SpamHunting IBR system. Improvements in the relevant term selection stage have allowed a significant enhancement over the obtained results. Moreover, concept drift should be kept in mind while the most relevant terms are being selected because some features can indicate its presence (and consequently they should not be removed).

The application of the Achieved Information (AI) measure has been suitable for selecting representative features in an e-mail. It has been designed for handling concept drift problem when the instance representation is computed. If instances are represented without taking care of concept drift, following stages of the CBR/IBR system will not be able to adequately support it.

Finally, as experimental results from this paper have shown, SVM and AdaBoost models get a great amount of correctly classified messages. We should note that these models are heavily focused in the feature selection issues. SVM model supports a second feature selection stage while the feature space is transformed into a new linearly separable space. In this process irrelevant features are discarded. In the other hand, AdaBoost constructs some weak classifiers by using subsets from all features and weights them according to its classification ability. When a weak classifier is assembled from inappropriate features it gets irrelevant because its weight will be very low.

Keeping in mind the previous related issues, future work should be focused in the relevant term selection process. Newer and original methods should be studied and probed with different e-mail corpus and preprocessing scenarios.

CBR/IBR systems have greatly contributed to the Spam filtering domain. As experimental results have shown, SpamHunting, ECUE, and *Odds-Ratio CBR* models are the most reliable choice for spam filtering. Therefore, we are aware of its probed capabilities for handling concept drift and manage disjoint concepts.

Acknowledgments

This work has been supported by the University of Vigo project SAEICS: *sistema adaptativo con etiquetado inteligente para correo spam / Adaptive system with intelligent labeling for spam e-mails*. The authors want to thank the considerable effort made by the reviewers in order to improve the quality of the paper.

References

1. Oard, D.W.: The state of the art in text filtering. User Modeling and User-Adapted Interaction, Vol. 7, (1997) 141–178
2. Wittel, G.L., Wu, S.F.: On Attacking Statistical Spam Filters. Proc. of the First Conference on E-mail and Anti-Spam CEAS, (2004)
3. Androutsopoulos, I., Paliouras, G., Michelakis, E.: Learning to Filter Unsolicited Commercial E-Mail. Technical Report 2004/2, NCSR "Demokritos", (2004)
4. 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
5. Cunningham, P., Nowlan, N., Delany, S.J., Haahr, M.: A Case-Based Approach to Spam Filtering that Can Track Concept Drift. Proc. of the ICCBR'03 Workshop on Long-Lived CBR Systems, (2003)
6. Fdez-Riverola, F., Iglesias, E.L., Díaz, F., Méndez, J.R., Corchado, J.M.: SpamHunting: An Instance-Based Reasoning System for Spam Labelling and Filtering. Decision Support Systems, (2006), *to appear*
7. 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
8. 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
9. Vapnik, V.: The Nature of Statistical Learning Theory. 2nd Ed. Statistics for Engineering and Information Science, (1999)
10. Lee, H., Ng, A.Y.: Spam Deobfuscation using a Hidden Markov Model. Proc. of the Second Conference on E-mail and Anti-Spam CEAS, (2005)
11. Druker, H., Vapmik, V.: Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks. Vol. 10 (5). (1999) 1048–1054
12. 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, MIT Press, (1999) 185–208
13. Schapire, R.E., Singer, Y.: BoosTexter: a boosting-based system for text categorization. Machine Learning, Vol. 39 (2/3). (2000) 135–168
14. Rigoutsos, I., Huynh, T.: Chung-Kwei: a Pattern-discovery-based System for the Automatic Identification of Unsolicited E-mail Messages (SPAM). Proc. of the First Conference on E-mail and Anti-Spam CEAS, (2004)
15. Graham, P.: Better Bayesian filtering. Proc. of the MIT Spam Conference, (2003)
16. 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
17. 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)
18. Gama, J., Castillo, G.: Adaptive Bayes. Proc. of the 8th Ibero-American Conference on AI: IBERAMIA-02, (2002) 765–774
19. 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
20. 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

21. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning, Vol. 23 (1). (1996) 69–101
22. Lenz, M., Auriol, E., Manago, M.: Diagnosis and Decision Support. Case-Based Reasoning Technology. Lecture Notes in Artificial Intelligence, Vol. 1400, (1998) 51–90
23. 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
24. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley, (1999)
25. 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, (2005) 17:129–138
26. Shannon, C.E.: The mathematical theory of communication. Bell Syst. Tech. J. Vol. 27. (1997) 379–423 & 623–656
27. Salton, G., McGill, M.: Introduction to modern information retrieval, McGraw-Hill, (1983)
28. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence IJCAI-95, (1995) 1137–1143
29. Oliver, J.J., Hand, D.J.: Averaging over decision stumps. Proc. of the European Conference on Machine Learning ECML-94, (1994) 231–241