Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 33 (2007) 36–48
Expert Systems with Applications
www.elsevier.com/locate/eswa

Applying lazy learning algorithms to tackle concept drift in spam filtering ☆

F. Fdez-Riverola [a,*], E.L. Iglesias [a], F. Díaz [b], J.R. Méndez [a], J.M. Corchado [c]

[a] Dept. Informática, University of Vigo, Escuela Superior de Ingeniería Informática, Edificio Politécnico, Campus Universitario As Lagoas s/n, 32004 Ourense, Spain
[b] Dept. Informática, University of Valladolid, Escuela Universitaria de Informática, Plaza Santa Eulalia 9-11, 40005 Segovia, Spain
[c] Dept. Informática y Automática, University of Salamanca, Plaza de la Merced s/n, 37008, Salamanca, Spain

Abstract

A great amount of machine learning techniques have been applied to problems where data is collected over an extended period of time. However, the disadvantage with many real-world applications is that the distribution underlying the data is likely to change over time. In these situations, a problem that many global eager learners face is their inability to adapt to local concept drift. Concept drift in spam is particularly difficult as the spammers actively change the nature of their messages to elude spam filters. Algorithms that track concept drift must be able to identify a change in the target concept (spam or legitimate e-mails) without direct knowledge of the underlying shift in distribution. In this paper we show how a previously successful instance-based reasoning e-mail filtering model can be improved in order to better track concept drift in spam domain. Our proposal is based on the definition of two complementary techniques able to select both terms and e-mails representative of the current situation. The enhanced system is evaluated against other well-known successful lazy learning approaches in two scenarios, all within a cost-sensitive framework. The results obtained from the experiments carried out are very promising and back up the idea that instance-based reasoning systems can offer a number of advantages tackling concept drift in dynamic problems, as in the case of the anti-spam filtering domain.

Keywords: IBR system; Concept drift; Anti-spam filtering; Model evaluation

1. Introduction and motivation

The huge expansion of Internet usage in recent years has increased marketing opportunities. As a result, the problem of spam has grown astronomically, and the earlier techniques for keeping it under control no longer work. Unsolicited commercial communications now represent more than 50% of e-mail traffic in the European Union and around the world.

Spam is beginning to undermine the integrity of e-mail and even to discourage its use. The great majority of Internet users' mailboxes are swamped by unwanted messages over which they have no control. In large numbers, Internet users have reported that they trust e-mail less, and 29% of users even say they do not use e-mail as much as they used to because of spam (Fallows, 2004). Users worry that the growing volume of spam is getting in the way of their ability to safely send and receive e-mail.

In order to reduce the inconveniences continually imposed by spam, a number of advances are being made. The success of machine learning (ML) techniques in text categorization (Sebastiani, 2002) has led researchers to explore learning algorithms in anti-spam filtering. However, the spam domain has a further complication because it is a cost-sensitive problem: the cost of accidentally blocking a

☆ This work has been supported by the Spanish Council for Science and Technology (MEC) in projects TIC2003-07369-C02-02.
* Corresponding author. Tel.: +34 988387015; fax: +34 988387001.
E-mail addresses: riverola@uvigo.es (F. Fdez-Riverola), eva@uvigo.es (E.L. Iglesias), fdiaz@infor.uva.es (F. Díaz), moncho.mendez@uvigo.es (J.R. Méndez), corchado@usal.es (J.M. Corchado).

0957-4174/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.04.011

legitimate message can be higher than letting a spam message pass the filter, and this difference must be taken into account during both training and evaluation tasks (Androutsopoulos et al., 2000; Hidalgo, López, & Sanz, 2000).

Another important aspect of anti-spam filtering domain is the necessity to manage concept drift problem. While much of the research on machine learning has focused on static problems (Vapnik, 1999), a significant issue in many real-world domains is its changing environment (Kelly, Hand, & Adams, 1999). In those situations, the target concepts (spam or legitimate e-mails) may depend on some hidden context and usually this context is dynamic. Changes in the hidden context can induce changes in the target concept, which is generally known as concept drift (Widmer & Kubat, 1996). Concept drift in spam is particularly difficult as the spammers actively change the nature of their messages to elude spam filters.

Research on concept drift shows that lazy learning algorithms are among the most effective models (Widmer & Kubat, 1996). With lazy learning the decision of how to solve a problem is deferred until the last moment, making it possible to use knowledge about the non-available domain until that moment. On the other hand, eager learning systems determine their generalisation mechanism by building a model based on training data before considering any new unseen knowledge.

In this paper, we propose two new techniques for tracking concept drift in our novel SpamHunting model, a fully automated instance-based reasoning (IBR) system for spam labelling and filtering. We show how RTI (relevant term identification) and RMS (representative message selection) techniques can significantly augment the accuracy of our system while leaving the rest of advantages unchanged.

The rest of the paper is organized as follows: Section 2 introduces an overview of other work using machine learning as well as memory and case-based techniques for anti-spam filtering; Section 3 deals with the problem of concept drift, summarizing several approximations to detect and inform the proposed models; Section 4 discusses in detail RTI and RMS techniques for managing concept drift in our previous SpamHunting system; Section 5 introduces our experimental results, investigating separately the effect of several evaluation metrics with different models and corpus; Finally, Section 6 concludes and suggests new directions for further research.

2. Classical approaches in spam filtering

Because of the volume of spam e-mail and its evolving nature, many ML techniques have been applied in the domain of anti-spam filtering. The Naïve Bayes learner is the most widely used algorithm. Although its independence assumption is over-simplistic, studies in anti-spam filtering have found Naïve Bayes to be effective (Androutsopoulos et al., 2000; Androutsopoulos, Koutsias, Chandrinos, Paliouras, & Spyropoulos, 2000; Androutsopoulos, Paliouras, & Michelakis, 2004; Sahami, Dumais, Heckerman, & Horvitz, 1998). Additionally, its improved version called Flexible Bayes provides an alternative approach for continuous attributes (John & Langley, 1995).

Another accurate technique is given by the use of support vector machines (SVM) (Vapnik, 1999). SVMs can use all the terms of the available messages because their learning capacity does not degrade even if many characteristics exist (Drucker, Wu, & Vapnik, 1999). Boosting algorithms have also been used as weak learners. Examples of this technique can be seen in the work of Friedman, Hastie, and Tibshirani (2000), the well-known AdaBoost (Schapire & Singer, 2000) or boosting using C4.5 trees (Drucker et al., 1999).

Examples of systems that generate classification rules are Ripper (Cohen & Singer, 1999) and Rocchio (Joachims, 1997). Ripper implements a method for inducing classification rules from a set of examples. Unlike the previous commented algorithms it does not need a feature vector, instead of this, it forms *if–then* rules which are disjunctions of conjunctions. Rocchio uses normalized TF/IDF (term frequency–inverse/document frequency) representation of the training vectors. The advantage of Rocchio algorithm is its fast training and testing stages. Another approximations are the use of Latent Semantic Indexing (Gee, 2003) or the use of C4.5 with PART (Hidalgo et al., 2000).

Those techniques assign the same importance to all errors, which does not apply to the spam domain, where a false positive (FP) error is more serious than a false negative (FN) one. In order to create cost-sensitive learning methods, successful adaptations of existing algorithms have been made (Ting, 1998). Another alternative is the generation of a cost-sensitive classifier starting from a learning algorithm plus a training collection and a cost distribution (Gómez, Puertas, Carrero, & De Buenaga, 2003).

As part of the study of anti-spam filtering techniques, there has also been intensive research into the use of memory-based classifiers (Androutsopoulos et al., 2000; Androutsopoulos et al., 2004; Sakkis et al., 2003). In general, the use of memory-based anti-spam filters is leading to better results than ML algorithm-based approaches, mainly when the cost of the FP errors is high (Sakkis et al., 2003). TiMBL (Daelemans, Jakub, van der Sloot, & van den Bosch, 1999) provides an implementation of a basic memory-based classification algorithm with a variant of *k-nn*. One important difference from *k-nn* basic is in the definition of the *k*-neighbourhood. TiMBL considers all training instances at the *k* closest *distances* form the unseen instance.

Moreover, case-based approaches are suitable for spam classification because they offer a natural framework to unify learning and collaboration approaches and to continually learn in the presence of new knowledge (Cunningham, Nowlan, Delany, & Haahr, 2003). Case-based approaches outperform previous techniques in anti-spam filtering (Delany et al., 2004). This is because spam is a disjoint concept: spam about *porn* has little in common with

spam offering *rolexes*. Case-based classification works well for disjoint concepts whereas ML techniques try to learn a unified concept description. Another advantage of this approach is the ease with which it can be updated to catch the concept drift in spam. The work of Delany et al. (2004) presents a case-based system for anti-spam filtering called ECUE (e-mail classification using examples) that can learn dynamically. In ECUE each e-mail is a case represented as a vector of binary features. The system uses a similarity retrieval algorithm based on the utilization of case retrieval nets (CRN) (Lenz, Auriol, & Manago, 1998). CRN networks are equivalent to the *k-nn* algorithm but are computationally more efficient in domains where there is feature-value redundancy and missing features in cases, such as in spam. ECUE is a system evolved from a previously successful model (Cunningham et al., 2003) designed by the same authors.

3. An assessment of concept drift in the spam domain

Machine learning focuses on systems that can learn a symbolic description of a concept. The goals for this concept are that it should be accurate, simple, general and that readable by domain experts. When a classifier for a static concept is learned, it can be used to classify future instances indefinitely. However, for many learning tasks of real-world data, when it is collected over an extended period of time, its underlying distribution is likely to change because concepts are often not stable. This kind of phenomenon is known as concept drift.

A typical example is spam filtering, where both the legitimate and the spam e-mail issue can change over time. Such changes are usually referred to as concept drift (Schlimmer & Granger, 1986) and can be caused by a changed context. As an example, the type of messages for a given user may be different due to a new job, studies, hobbies, etc. Often these changes make the model built on old data inconsistent with the new data, and regular updating of the model is necessary. The cause of change is usually hidden, not known a priori or given explicitly in the form of predictive features, making the learning task more complicated.

A problem with many global eager learners is their inability to adapt to local concept drift (only particular types of spam may change with time, while others remain the same) (Tsymbal, 2004). In this situation, many global models are discarded simply because their accuracy on the current data falls, even if they are still good experts for stable parts of the data.

3.1. The problem of concept drift

As mentioned earlier, on-line learning in domains where the target concept depends on hidden context presents several difficulties. Two kinds of *actual* concept drift that may occur in the real world are normally distinguished in the literature (Standley, 2003): *sudden* and *gradual* concept drift. However, hidden changes in context may not only be a cause of a change of target concept, but may also cause a change of the underlying data distribution. The need for a change in the current model due to the change of data distribution is called *virtual concept drift* (Widmer & Kubat, 1996). Both *actual* and *virtual* concept drift may occur together or separately.

Theoretical results in handling concept drift have also been studied in computational learning theory. In particular, the work of Helmbold and Long (1994) establishes bounds on the *extent of drift* that can be tolerated assuming possibly permanent but very slow drift, whereas in Kuh, Petsche, and Rivest (1991) a maximal frequency of concept changes (rate of drift) that is acceptable by any learner is defined. However, it cannot usually be guaranteed that the application at hand obeys these restrictions. Hence more application-oriented approaches rely on intuitive heuristics that work well in their particular application domain, but their parameters usually require tuning and are not often transferable to other domains (Klinkenberg & Rüping, 2003).

Three approaches to handling concept drift can be distinguished in the available systems: (i) instance selection (ii) instance weighting and (iii) ensemble learning.

The most common concept drift handling technique is based on instance selection and consists in generalizing from a window that moves over recently arrived instances and uses the learnt concepts for prediction only in the immediate future. Some algorithms use a time window of fixed size (Kubat, 1989), while others use heuristics to adjust the window size to the current extent of concept drift (Klinkenberg, 2004; Widmer & Kubat, 1996). For windows of fixed size, the choice of an appropriate window size is a compromise between fast adaptation and good generalization in phases without concept change. The basic idea of adaptive window management is to adjust the window size to the current extent of concept drift. Many case-base editing strategies in case-based reasoning that delete noisy, irrelevant and redundant cases are also a form of instance selection (Cunningham et al., 2003).

Instance weighting uses the ability of some learning algorithms such as support vector machines to process weighted instances (Klinkenberg & Joachims, 2000; Syed, Liu, & Sung, 1999; Taylor, Nakhaeizadeh, & Lanquillon, 1997). Data or parts of the hypothesis are weighted according to their age and/or utility for the classification task. In Klinkenberg (2004), Klinkenberg shows that instance weighting techniques handle concept drift less effectively than analogous instance selection techniques, which is probably due to over fitting the data.

Finally, ensemble learning maintains a set of concept descriptions, the predictions of which are combined using voting or weighted voting, or the most relevant description is selected. Building on the analysis presented in Kuncheva (2004), the techniques for using ensemble to handle concept drift fall into two groups (Delany, Cuningham, & Tsymbal, 2005): (i) *dynamic combiners* where the base classifiers are trained in advance and the concept drift is tracked by

changing the combination rule and (ii) *incremental approaches* that use fresh data to update the ensemble and incorporate a "forgetting" mechanism to remove old or redundant data from the ensemble.

As mentioned by Widmer and Kubat (1996) effective learning in environments with hidden contexts and concept drift requires a learning algorithm that can detect context changes without being explicitly informed about them, and that can quickly recover from a context change and adjust its hypotheses to a new context, and make use of previous experience in situations where old contexts and corresponding concepts reappear.

3.2. Previous work handling concept drift

Perhaps the first systems capable of tracking concept drift in supervised learning were STAGGER (Schlimmer & Granger, 1986), FLORA (Kubat, 1989) and IB3 (Aha, Kibler, & Albert, 1991). Learning in time-varying environment has also been studied in the framework of genetic algorithms (Smith, 1987), neural networks (Narendra & Parthasarathy, 1990), classification trees C4.5 (Harries & Horn, 1995) and Support Vector Machines (Klinkenberg & Joachims, 2000; Syed et al., 1999). Computational learning theory has also investigated the problem (Hembold & Long, 1994; Kuh et al., 1991). In unsupervised learning, the system COBBIT (Kilander & Jansson, 1993) warrants some mention.

Among other approaches, lazy learning is able to adapt well to local concept drift due to its local nature. In the anti-spam filtering domain, the advantages of lazy learning algorithms for handling concept drift were discussed in Cunningham et al. (2003): (i) lazy learning performs well with disjoint concepts, such as spam, which consists of many different subtypes; (ii) case-bases in lazy learning are easy to update (e.g., when new types of spam appear); and (iii) lazy learning allows straightforward sharing of knowledge for particular types of problems, making it easier to maintain multiple potentially distributed case-bases. In the work of Delany, Cunningham, Tsymbal, and Coyle (2004) it is shown how concept drift can be managed in a CBR system simply by defining a set of editing techniques that use the competence characteristics of a case-base to remove noisy and redundant cases. Detailed information about these techniques can be found in Delany et al. (2004).

In the next section we present an improved version of our previous SpamHunting system (Fdez-Riverola, Iglesias, Díaz, Méndez, & Corchado, in press), where two complementary approximations are defined in order to select relevant terms (concepts) from significant (up-to-date) e-mails.

4. SpamHunting: a novel IBR technique to tackle concept drift

In this section we outline our SpamHunting system, a lazy learning hybrid model based on an IBR approach (Watson, 1997) to accurately solve the problem of spam labelling and filtering. This section summarizes the model architecture and explains in detail the improvements made to effectively track the concept drift problem: (i) capturing drift by selecting relevant updated terms (RTI technique) and (ii) representative message selection (RMS technique) by using an evolving sliding window.

4.1. Model operation overview

Whenever SpamHunting receives a new e-mail, the system executes a cycle that evolves through the four steps of a classical CBR system (Fdez-Riverola et al., in press). In order to classify each incoming e-mail correctly, SpamHunting creates a new message descriptor (lower part of Fig. 1). This message descriptor consists of a sequence of N features that better summarize the information contained in the e-mail. For this purpose, we store and index data from two main sources: (i) information obtained from the header of the e-mail and (ii) those terms that are more representative of the subject, body and attachments of the message.

The retrieval stage is carried out using our enhanced instance retrieval network (EIRN) model (upper part in Fig. 1). The EIRN model facilitates the indexation of instances and the selection of those that are most similar to the instance-message. The reuse of similar e-mails is carried out by means of the utilization of a weighted voting mechanism, which generates an initial solution by creating a model with the retrieved instances. The revision stage is only carried out in the case of spam messages. For this purpose, the system employs general knowledge in the form of meta-rules that are extracted from the e-mail headers. Finally, the retain (learning) stage is carried out whenever the system classifies an incoming e-mail, updating the knowledge structure of the whole system (e-mail base situated in the center of Fig. 1). The hybrid system also takes into account the feedback of the user when it receives an incorrectly classified e-mail.

In order to increase our knowledge about the concept drift problem and gain a deeper insight into the EIRN network operation, we have constructed a watching module that can be plugged into the IBR SpamHunting system. Fig. 2 shows a snapshot of this tool.

With the EIRN viewer we can obtain a visual approach to the distribution of the terms built up by the EIRN model over a whole period of time. For each selected e-mail (left panel in Fig. 2) a graphical representation is built which helps in the visualization of the prediction power of the relevant terms selected by the network for the current message. The more terms are selected close to the axes, the easier it will be to classify the target e-mail. Those terms that are situated along the main diagonal do not provide valuable information because they have the same probability of belonging to a spam or a legitimate message. From this tool, we have also gained access to the overall statistics of our EIRN network.

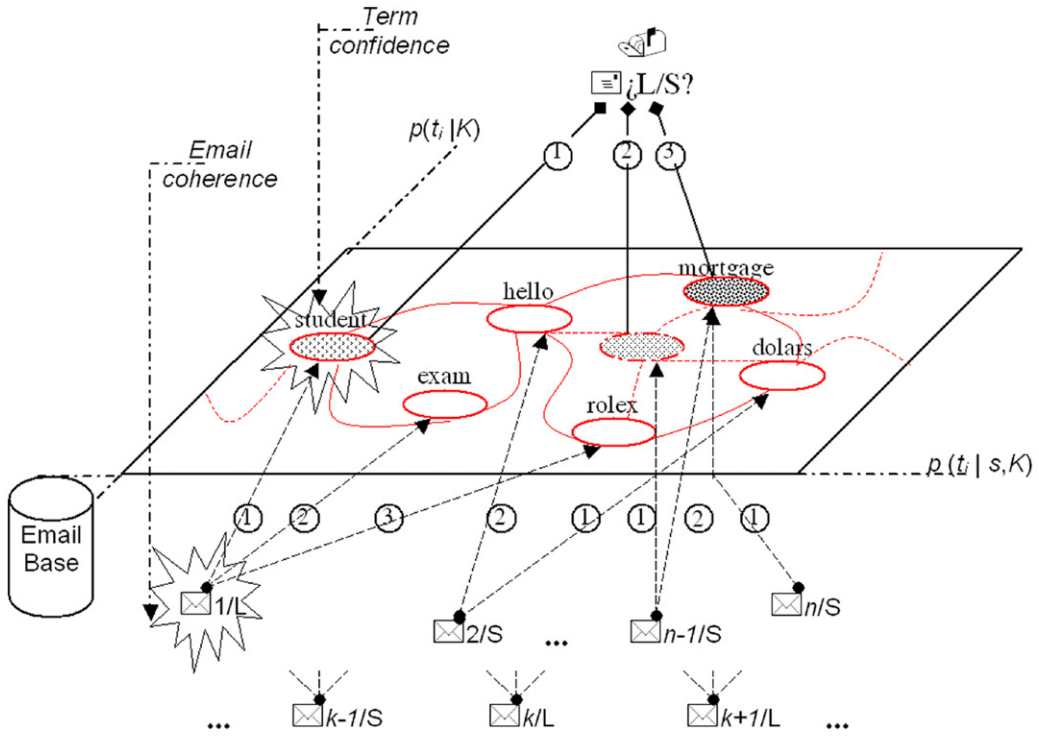

Fig. 1. SpamHunting instance representation and indexing structure.

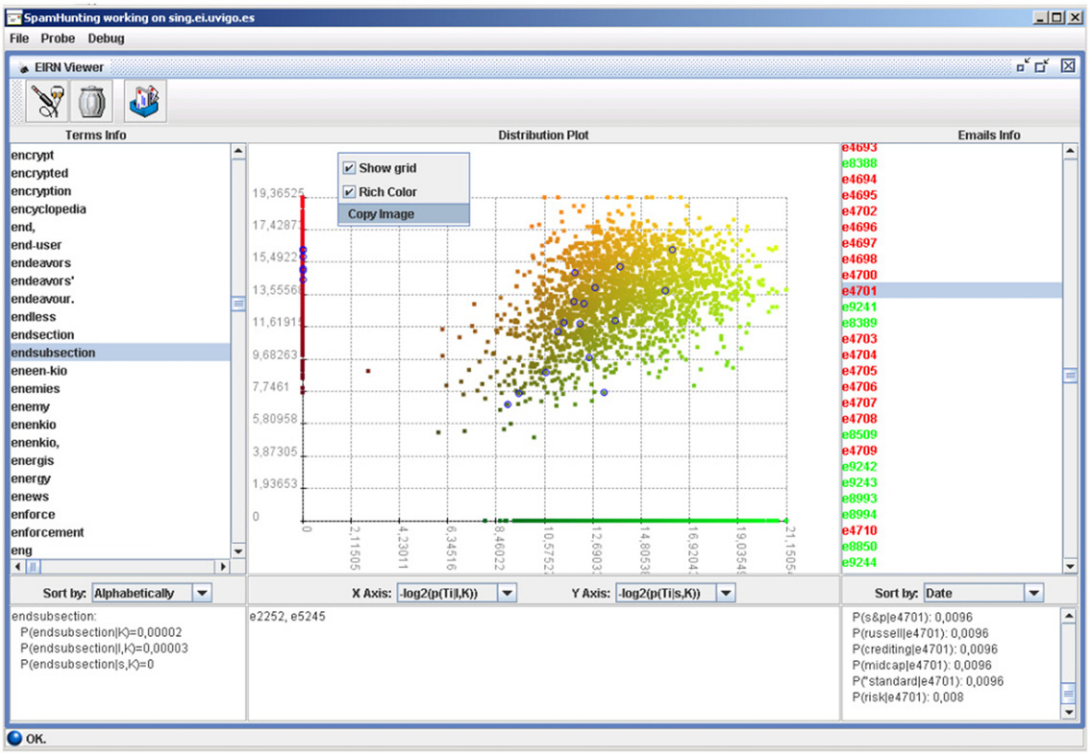

Fig. 2. EIRN viewer module of IBR SpamHunting system.

The application shown in Fig. 2 can be executed in two ways: (i) as a real-time watching window to follow the evolution of the system in operation or (ii) in edit mode, which allows the user to visualize and change the parameters that compose the EIRN model.

4.2. Capturing drift by relevant term identification

Starting from the list of words contained in a given message, we are interested in identifying and selecting the most relevant terms in this e-mail. Without any other

information, the only way to do this it is based on the frequency of each word in the message. But if we have available a set of e-mails (a whole corpus or a set of e-mails selected by a sliding window), we can use information about the underlying distribution of the e-mail set in relation to the target concept (spam or legitimate) to modulate the relevance of each word inside a specific message. Therefore, we are interested in defining a measure that can evaluate the relevance of a word in this way. The goal is to use this criterion to select the most relevant terms within the e-mail according to the target concept. Before defining the measure, we need to introduce some useful notation.

First, the set of available e-mails (whole corpus or sliding window) is denoted by $\boldsymbol{K}$, and the target concept by the set $\boldsymbol{C} = \{s, l\}$, where the symbols s and l stand for spam and legitimate status respectively. A classification of the messages of $\boldsymbol{K}$ is given by suprajective mapping between the sets $\boldsymbol{K}$ and $\boldsymbol{C}$. Namely, the classification class is given by

$$\begin{aligned} \text{class}: \quad & \boldsymbol{K} \to \boldsymbol{C} \\ & e_i \to \text{class}(e_i) \end{aligned} \tag{1}$$

From these classifications the set of messages can be partitioned into two classes: spam and legitimate e-mails. Any two e-mails e_i and e_j belong to the same class if class$(e_i) =$ class(e_j). Given an e-mail e_j, the set of features in the message after the preprocessing step is denoted by $\boldsymbol{T}|\{e_j\}$. So, given a corpus $\boldsymbol{K}$ with m^K e-mails ($m^K =$ card$(\boldsymbol{K})$), the set of terms belonging to the corpus, $\boldsymbol{T}|\boldsymbol{K}$, is given by Eq. (2):

$$\boldsymbol{T}|\boldsymbol{K} = \bigcup_{1 \leqslant j \leqslant m^k} \boldsymbol{T}|\{e_j\} \tag{2}$$

Moreover, given a corpus $\boldsymbol{K}$ with m^K e-mails, the absolute frequency of appearance of the term T_i in the spam messages of the corpus $\boldsymbol{K}$ will be denoted by n_{is}^K, with each term $T_i \in \boldsymbol{T}|\boldsymbol{K}$ $(i = 1, \ldots, f^K$, where $f^K =$ card$(T|\boldsymbol{K}))$. In the same way, the notation n_{il}^K stands for the absolute frequency of appearance of the term T_i in the legitimate messages of the corpus $\boldsymbol{K}$. Naturally, the sum of both frequencies, $n_{is}^K + n_{il}^K = n_i^K$, stands for the absolute frequency of appearance of the term T_i in the corpus $\boldsymbol{K}$. The total number of appearances of the terms in spam e-mails of the corpus $\boldsymbol{K}$, denoted by N_s^K, is given by the sum $\sum n_{is}^K$. In the same way, the total number of appearances of the terms in legitimate e-mails of the corpus $\boldsymbol{K}$ can be defined, $N_l^K = \sum n_{il}^K$, and in the whole corpus $\boldsymbol{K}$, $N^K = \sum n_i^K$. Once these absolute frequencies are determined, the probability distribution of the terms conditioned to the corpus $\boldsymbol{K}$, $p(T_i|\boldsymbol{K})$, can be estimated by its relative frequency using Eq. (3).

$$p(T_i|K) = \frac{n_i^K}{N^K} \tag{3}$$

Similarly, the conditional probability distribution of the terms given the spam or legitimate status of the e-mails in the corpus $\boldsymbol{K}$, $p(T_i|s, \boldsymbol{K})$ and $p(T_i|l, \boldsymbol{K})$, respectively, can be estimated by their relative frequencies:

$$p(T_i|s, K) = \frac{n_{is}^K}{N_s^K} \quad \text{and} \quad p(T_i|l, K) = \frac{n_{il}^K}{N_l^K} \tag{4}$$

Once the notation has been introduced, we are interested in defining criteria for the relevance of each term, T_i, which appears in a specific e-mail, e, of a corpus $\boldsymbol{K}$. In order to define this measurement, the following reasoning is carried out. First, the probability that the e-mail e is a spam message can be expressed as

$$p(s|e) = \sum_{t_i \in e} p(s|T_i, e) p(T_i|e) \tag{5}$$

The expression $p(T_i|e)$ is known, given the e-mail e. Although the expression $p(s|T_i, e)$ is unknown, it can be estimated by the probability $p(s|T_i, \boldsymbol{K})$. That is, it can be approximated by the probability that an e-mail in the corpus $\boldsymbol{K}$ is a spam message if the term T_i is present in that e-mail. Therefore, expression (5) can be approximated by

$$p(s|e) \approx \sum_{T_i \in e} p(s|T_i, \boldsymbol{K}) p(T_i|e) \tag{6}$$

After this, and applying the Bayes' rule, the probability $p(s|e)$ can be expressed as

$$\begin{aligned} p(s|e) &\approx \sum_{T_i \in e} \frac{p(T_i|s, \boldsymbol{K}) p(s|\boldsymbol{K})}{p(T_i|\boldsymbol{K})} p(T_i|e) \\ &= p(s|\boldsymbol{K}) \sum_{T_i \in e} \frac{p(T_i|s, \boldsymbol{K}) p(T_i|e)}{p(T_i|\boldsymbol{K})} \end{aligned} \tag{7}$$

Secondly, the probability that an e-mail is legitimate given the e-mail, $p(l|e)$, can be determined in a similar way:

$$\begin{aligned} p(l|e) &\approx \sum_{T_i \in e} \frac{p(T_i|l, \boldsymbol{K}) p(l|\boldsymbol{K})}{p(T_i|\boldsymbol{K})} p(T_i|l) \\ &= p(l|\boldsymbol{K}) \sum_{T_i \in e} \frac{p(T_i|l, \boldsymbol{K}) p(T_i|e)}{p(T_i|\boldsymbol{K})} \end{aligned} \tag{8}$$

Moreover, we are interested in truly discriminating between spam terms and legitimate terms (those that are situated near the axes in Fig. 2). Therefore, the relevance measure of a term would be able to stress one term which is probably only in spam messages or only in legitimate messages, but is not equally probable in both kinds of e-mail, simultaneously. This fact can be modelled by means of the difference between the expressions (7) and (8), and each term of the sum can be interpreted as a measure of the contribution of each term in the final result, namely, a measure of the relevance of each term. Moreover, if we are not interested in the sign of the contribution (positive if the term helps to classify an e-mail as spam or negative if it helps to classify it as legitimate), the relevance of each term of the e-mail can be defined as follows:

$$r(T_i,e)=\left\{\frac{|p(s|\boldsymbol{K})p(T_i|s,\boldsymbol{K})-p(l|\boldsymbol{K})p(T_i|\boldsymbol{K})|}{p(T_i|\boldsymbol{K})}\right\}p(T_i|e) \tag{9}$$

The first factor in $r(T_i,e)$ depends on the corpus $\boldsymbol{K}$ and can be computed with the statistics given by expressions (3) and (4). This factor modulates the relevance of the term T_i inside the e-mail e, given by the second factor, $p(T_i|e)$. This formulation can be used to select the most relevant terms in two ways: (i) a fixed number of terms ordered with respect to $p(T_i|e)$ or (ii) a variable number of terms depending on the percentage of the whole sum of individual relevancies. The latter approach was used in the experiments carried out in this paper.

4.3. Representative message selection by using an evolving sliding window

The relevance metric given by expression (9) depends on the underlying probability distributions of the terms within the given corpus $\boldsymbol{K}$. As previously mentioned, in many real-world domains the target concepts (spam or legitimate e-mails) may depend on some hidden context and usually this context is dynamic. If the target concept can change, we can assume that the relevance of the terms can also change over time. Therefore it is desirable to have a mechanism which envisages this fact in order to measure the relevance of a term.

In this case, we propose a sliding window as a suitable mechanism to track and compute efficiently the underlying probability distributions used in expression (9), namely, the marginal distribution $p(T_i|\boldsymbol{K})$ and the conditional distributions $p(T_i|s,\boldsymbol{K})$ and $p(T_i|l,\boldsymbol{K})$. Assuming that the available e-mails (past e-mails, current e-mails and future e-mails) can be arranged in a specific order, $\sigma=\{\sigma(i)\}_{i\in\mathbb{N}}$, (e.g., a temporal order), a window $\boldsymbol{K}_t$ of size W at an epoch τ ($\tau\in\mathbb{N}$) can be defined as the subset of e-mails $\boldsymbol{K}_\tau=\{e_{\sigma(i)}:\tau\cdot W\leqslant i<(\tau+1)\cdot W\}$, according to the order $\boldsymbol{\sigma}$. From this subset, the probability distributions $p(T_i|\boldsymbol{K}_\tau)$, $p(T_i|s,K_\tau)$ and $p(T_i|l,\boldsymbol{K}_\tau)$ can be computed according to the expressions (3) and (4).

Now, we are interested in defining the movement of a window to the next epoch. As Fig. 3 shows, if the current epoch is τ, the window at next epoch can be viewed as the result of the following operation with sets, $\boldsymbol{K}_{\tau+1}=\boldsymbol{K}_\tau\cup\boldsymbol{I}_\tau-O_\tau$, where $\boldsymbol{K}_\tau$ is the current window, I_τ is the set of e-mails which are included in the new window, and $\boldsymbol{O}_\tau$ is the set of e-mails which leave the current window. If the size of the window is constant, that is $W=f(\tau)=k$, necessarily, the size of sets $\boldsymbol{I}_\tau$ and $\boldsymbol{O}_\tau$ must be also constant and equal, namely, $\Delta W=g(\tau)=k'$.

The probabilities of interest for the corpus at current epoch, K_τ, can be incrementally updated for the next epoch $\tau+1$. Given the counts for each term T_i in current corpus K_τ, namely, $n_i^{K_\tau}$, $n_{is}^{K_\tau}$, and $n_{il}^{K_\tau}$, and the counts for each term T_i in the O_τ subset, $n_i^{O_\tau}$, $n_{is}^{O_\tau}$, and $n_{il}^{O_\tau}$, we can also count the occurrences of each term T_i as new e-mails are gathered and included in the set $\boldsymbol{I}_\tau$, namely, the counts $n_i^{I_\tau}$, $n_{is}^{I_\tau}$, and $n_{il}^{I_\tau}$. When the current window at epoch τ must be shifted to the next epoch $\tau+1$, we can update incrementally the probability distributions from this counters as follows:

$$\begin{aligned}
p(T_i|K_{\tau+1})&=\frac{n_i^{K_\tau}+n_i^{I_\tau}-n_i^{O_\tau}}{N^{K_\tau}+N^{I_\tau}-N^{O_\tau}}\\
p(T_i|s,K_{\tau+1})&=\frac{n_{is}^{K_\tau}+n_{is}^{I_\tau}-n_{is}^{O_\tau}}{N_s^{K_\tau}+N_s^{I_\tau}-N_s^{O_\tau}}\\
p(T_i|l,K_{\tau+1})&=\frac{n_{il}^{K_\tau}+n_{il}^{I_\tau}-n_{il}^{O_\tau}}{N_l^{K_\tau}+N_l^{I_\tau}-N_l^{O_\tau}}
\end{aligned} \tag{10}$$

Finally, a variable size of the sliding window with epochs τ can also be considered. To model how the size of the sliding window changes with τ, it is assumed that the set of new e-mails, which are newly included in the next window, the size of $\boldsymbol{I}_\tau$, is always a fixed number ΔW. At the same time, the number of e-mails which are discarded from the current window varies. During the initial epochs the number of discarded e-mails are less than the number of e-mails which are removed at a later epoch. This can be interpreted as the ability of the system to "forget" older e-mails over the course of the epochs. Therefore, the number of discarded e-mails at epoch τ, the size of the set $\boldsymbol{O}_\tau$, can be considered as an increasing function with τ according to the following law:

$$\Delta W\left(1-\mathrm{e}^{-\frac{\tau}{T^2}}\right) \tag{11}$$

where T is the parameter that controls the ability of the system to forget, and is referred to as *memory rate*. If the memory rate T grows, the system forgets e-mails slower than if the memory rate T is less. Once the size of the sets $\boldsymbol{I}_\tau$ and $\boldsymbol{O}_\tau$ are established, the size of the next window $\boldsymbol{K}_{\tau+1}$, $W(\tau+1)$ can be computed from the size of current window $W(\tau)$ as

$$W(\tau+1)=\begin{cases}W(\tau)+\Delta W\cdot\mathrm{e}^{-\frac{\tau}{T^2}} & \text{if } \tau>0\\ W_0 & \text{if } \tau=0\end{cases} \tag{12}$$

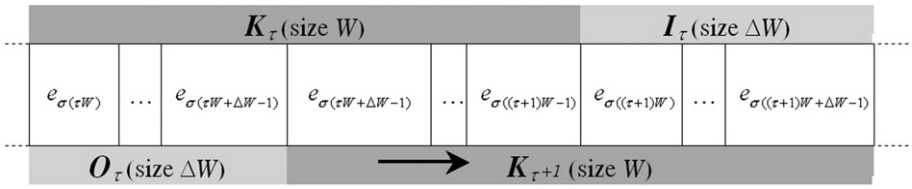

Fig. 3. Movement of the sliding window.

5. Model evaluation

From our previous work (Fdez-Riverola et al., in press), we demonstrated that the most accurate spam filtering model was our earlier version of SpamHunting and the ECUE system (Delany et al., 2004). Apart from these models, we also tested Naïve Bayes, support vector machines and AdaBoost algorithms obtaining least effective performance from a classical static point of view, where new messages are simply classified and no update of the model occurs.

This section introduces our evaluation of the proposed RTI and RMS techniques within the SpamHunting system for effectively tracking concept drift. The results are presented in two scenarios. Firstly, we carry out an evaluation of the performance of the system from a classical dynamic point of view, where new messages are simply classified and the case-base is updated with the predicted message. Secondly, we test our system following a user-dependent schema, where the model is trained with one corpus and tested with another different one. The experiments carried out were offline evaluations using e-mails collected over an extended period of time.

The selected models for the actual experimentation were our previous SpamHunting system, now implementing RTI and RMS techniques, and an improved version of the ECUE system (Delany et al., 2004). The final goal was to compare the performance of both approximations in a dynamic real environment.

5.1. Experimental setup

The key objective was to evaluate the performance of the improved SpamHunting systems over other well-known approaches in two different experiments. For this reason, we used 10-fold stratified cross-validation (Kohavi, 1995) in the first experiment, a technique that increases the confidence of experimental findings when using small datasets. Therefore, the available messages were partitioned into 10 parts, with each part maintaining the same ratio of legitimate and spam messages as the entire corpus. Each experiment was repeated 10 times, each time reserving a different part as the testing corpus and using the remaining 9 parts as the training corpus. Selected performance scores were then averaged over the 10 iterations.

Despite privacy issues regarding the content of a message, there are several publicly available corpuses on spam. In our work, we use the SpamAssassin corpus[1] (for both first and second experiments) and the Ling-Spam corpus[2] (for the second experiment only). The SpamAssassin corpus contains 9332 different messages received from January 2002 up to and including December 2003 distributed as Table 1 shows.

Table 1
Description of the SpamAssassin corpora of e-mails

Year	Legitimate messages	Spam messages	L:S ratio	Total messages
2002	2801 (84.9%)	498 (15.1%)	5.62	3299
2003	4150 (68.8%)	1883 (31.2%)	2.20	6033

Table 2
Description of the Ling-Spam corpora of e-mails

Year	Legitimate messages	Spam messages	L:S ratio	Total messages
–	2412 (83.4%)	481 (16.6%)	5.02	2893

The Ling-Spam corpus contains 2893 e-mails where the spam messages were donated by one author and the legitimate messages were retrieved from the archives of a moderated, and hence spam-free, list about linguistics. Table 2 shows how these messages are distributed given their class.

From Table 2 it can be observed that the legitimate-to-spam ratio in the Ling-Spam corpus (5.02) is very close to those messages belonging to the year 2002 of the SpamAssassin corpus (5.62) (see Table 1). Although Ling-Spam has the disadvantage that its legitimate messages are more topic-specific than the legitimate messages most users receive, it will be of great help to test the adaptability of the analyzed systems.

5.2. Dynamic evaluation: gradual concept drift

The objective of this evaluation was to examine at a detailed level the performance of both systems (ECUE and SpamHunting) with continuous updating of the case-base through the storage of each new classified e-mail. In this experiment, our SpamHunting system was tested with different configurations varying the number of selected terms for each e-mail (as explained in Section 4.2) and taking into account only the messages belonging to the sliding window.

Fig. 4 shows the percentage of correct classifications (%OK), percentage of false positives (%FP) and percentage of false negatives (%FN) belonging to the analyzed models. From Fig. 4 we can surmise that the model producing a higher percentage of correct answers and a lesser number of FP errors is the SpamHunting system with a 60% of relevant terms captured for each e-mail.

In order to obtain a deeper insight into the operation of the different models analyzed, we calculate the recall (Fig. 5a) and precision (Fig. 5b) scores for the ECUE system and the five variants of the SpamHunting system. From Fig. 5a (filter effectiveness) it can be seen that the best model is the ECUE system. The variants of the SpamHunting systems are within the same interval of correctly classified spam messages, following the ECUE closely

[1] The SpamAssassin corpus was created by Justin Mason of Network Associates, and is publicly available for download at http://www.spamassassin.org/publiccorpus/.

[2] The Ling-Spam corpus is publicly available for download at http://www.iit.demokritos.gr/skel/i-config/.

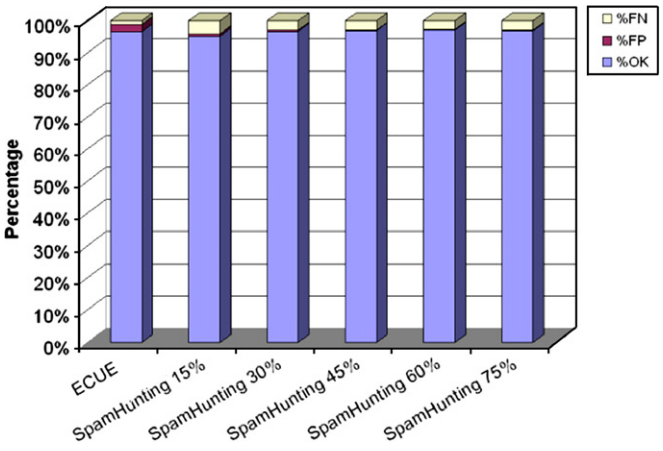

Fig. 4. Percentage of correct classifications, FP errors and FN errors from validation over the SpamAssassin corpus.

and with a high value of spam recall. From Fig. 5b (filter safety) it can be seen that the technique that best classifies spam messages is SpamHunting [60%]. In this case, the model with worst precision is ECUE.

In general, a high spam recall value indicates low FN error rate, and a high spam precision value implies low FP error rate. These two metrics are straightforward to understand, but do not reflect differential treatment of the two types of errors. The TCR score is introduced for this reason (Androutsopoulos et al., 2004), where higher TCR values indicate better performance of the models.

Now, let us assume that FP errors are λ times more costly than FN errors, where λ depends on the usage scenario. Three different usage scenarios are used in our experiments. In the first one, the filter flags messages when it suspects them to be spam, without removing them. In this case, $\lambda = 1$. The second scenario assumes that messages classified as spam are returned to the sender. In this scenario $\lambda = 9$ is considered, that is, mistakenly blocking a legitimate message was taken to be as bad as letting nine spam messages pass the filter. In the third scenario messages classified as spam are deleted automatically without further processing. Now $\lambda = 999$ is used. Fig. 6 shows the results taking into account the TCR score and varying the λ parameter as commented above.

From Fig. 6 we can see that when FP errors are assigned the same importance as FN errors (a non-realistic point of view for a final user) there are differences between the models but they are not very significant variations. As soon as one increases the importance of classifying legitimate e-mail correctly (considering an FP error to be more costly than an FN error) the situation changes drastically and the SpamHunting system with 60% of relevant terms selected produces much better results. This circumstance is supported by the high precision score obtained by the SpamHunting system shown in Fig. 5b and the better ratio between FP and FN errors demonstrated in Fig. 4. Fig. 6d clearly shows how the SpamHunting [60%] system outperforms the rest of the analyzed models in all the tree different usage scenarios.

Once we have compared both efficiency and efficacy of our SpamHunting system against ECUE, it is interesting to show how our EIRN model stores the information that it captures from the training messages. Columns in Table 3 represent the mean value of message and model terms for several configurations of the EIRN network.

The first row of Table 3 shows for each EIRN configuration, the mean value for the number of selected terms for a randomly chosen e-mail using the RTI technique. The second row of Table 3 indicates the number of different terms indexed by the corresponding EIRN network. For the experiments carried out in this paper, the best performance of the EIRN network was achieved storing the 60% of the total terms of frequency of each e-mail. This led to indexing 85,668 terms of the whole corpus and to representing each e-mail using an average of 41 terms.

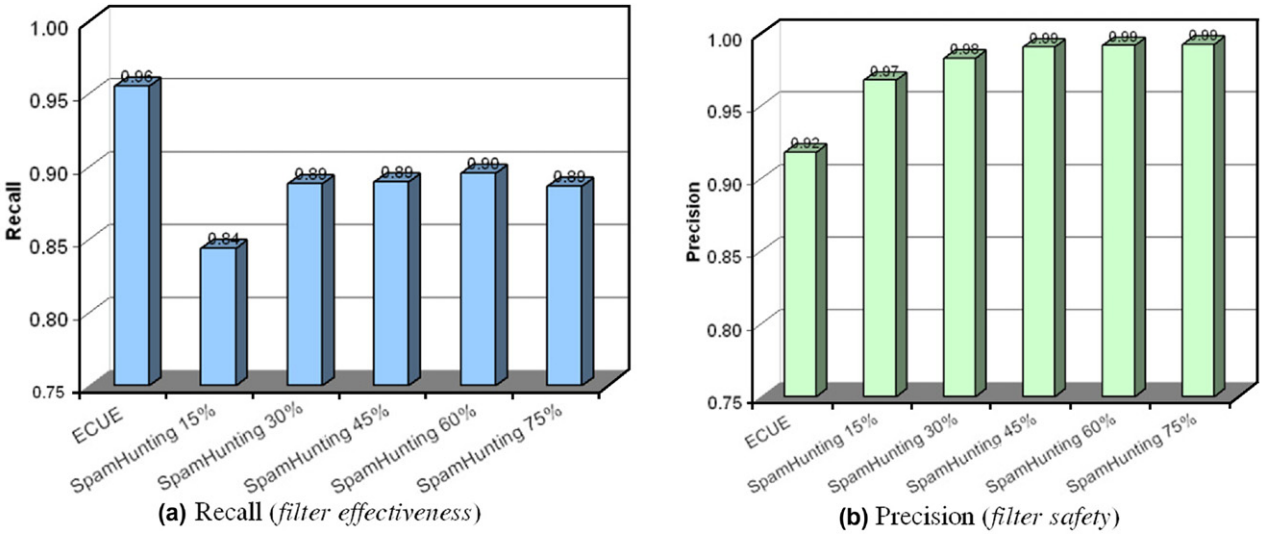

Fig. 5. Recall and precision values for the analyzed models.

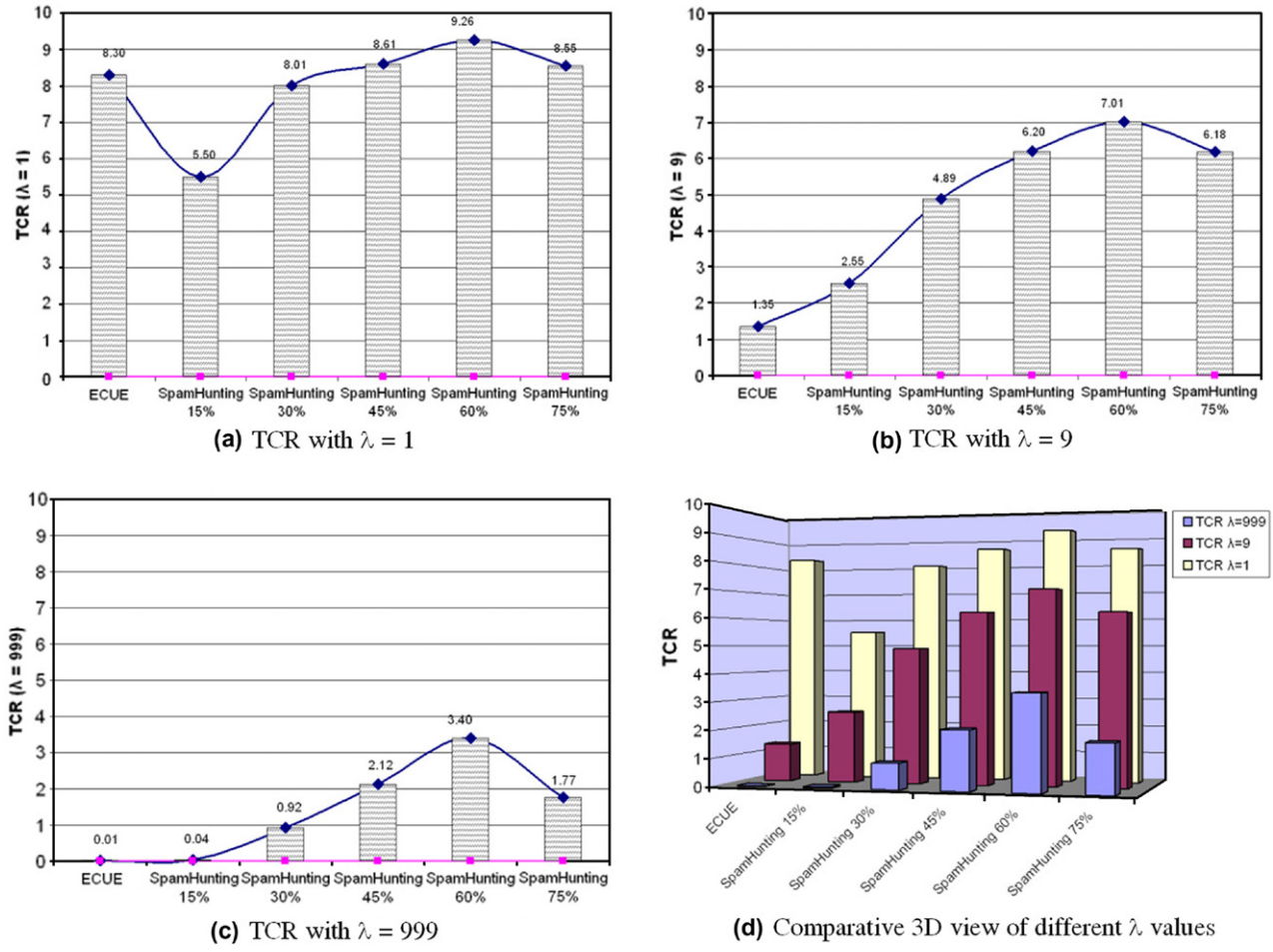

Fig. 6. TCR values for the analyzed models varying the *l* parameter over the SpamAssassin corpus.

Table 3
Mean value of message and model terms for several configurations of the EIRN network built by the SpamHunting system

	SpamHunting				
	15%	30%	45%	60%	75%
Message terms	4	13	26	41	58
Model terms	10,772	28,792	55,618	85,668	106,655

The parameters for the correct configuration of the sliding window were $\Delta W = 1400$ and a memory rate $T = 250$.

5.3. User-dependent evaluation: sudden concept drift

The goal of this experiment is to demonstrate the adaptability of a filter when it is trained with data coming from the mailbox of one user and tested with data belonging to another different user. Such a situation is especially interesting for the construction of filters that can operate on an enterprise level (e.g., with the filter running in a multi-user environment inside an ISP mail server). Moreover, this is a good approximation where sudden concept drift can occur because the system will capture messages containing different mailboxes.

In this experiment we will compare the best configuration of our enhanced SpamHunting system against ECUE. Moreover, we also show the performance obtained when classical ML approaches are used. We test Naïve Bayes, SVM and AdaBoost algorithms without any model rebuild during the test stage. All these models except our SpamHunting system use Information Gain to select the most predictive features since it has been shown to be an effective technique in aggressive feature removal in text classification. For our comparisons, we have selected the best performance model of each technique varying between 100 and 2000 features.

All the selected models were trained with the whole SpamAssassin corpus (9332 different messages, see Table 1) and were tested with the whole Ling-Spam corpus (2893 e-mails, see Table 2). SpamHunting and ECUE case-bases were updated as new e-mails arrived.

As in the previous experiment, Fig. 7 shows the percentage of correct classifications (%OK), percentage of false positives (%FP) and percentage of false negatives (%FN) belonging to the five analyzed models. From Fig. 7 we can surmise that the model producing a higher percentage of correct answers is our SpamHunting system, followed by SVM and Adaboost models. In this case, the rest of the

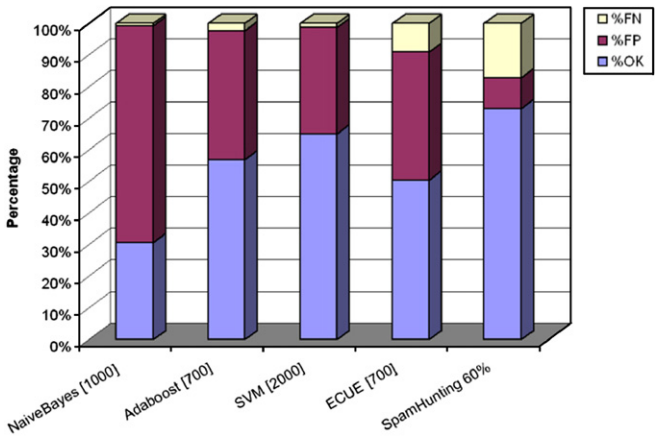

Fig. 7. Percentage of correct classifications, FP errors and FN errors from validation over the Ling-Spam corpus.

analyzed models (ECUE and NaiveBayes) achieve significantly worse results.

Now, if we focus our attention on the number of FP errors produced by each model analyzed in Fig. 7, we can appreciate that the best results are again achieved by the SpamHunting system, significantly reducing the FP rate. Surprisingly, the model with least FN errors is the Naïve Bayes network, but at the expense of a higher number of FP errors.

As in the previous experiment, let us assume that FP errors are λ times more costly than FN errors, where λ is assigned to 1, 9 and 999 in three different scenarios. From Fig. 8 one can surmise that in all situations the best model is our SpamHunting system. This new data backs up the results of the experiment carried out previously.

From this experiment, we can also conclude that the best model handling sudden concept drift is our SpamHunting approximation implementing RTI and RMS proposed techniques. This affirmation is reinforced by the intrinsic lazy learner that our model incorporates in its life-cycle and it encourages us to carry on with further research in this field. Moreover, it is worthwhile highlighting the results obtained by the SVM algorithm that demonstrates a performance able to improve on the ECUE system in several scenarios.

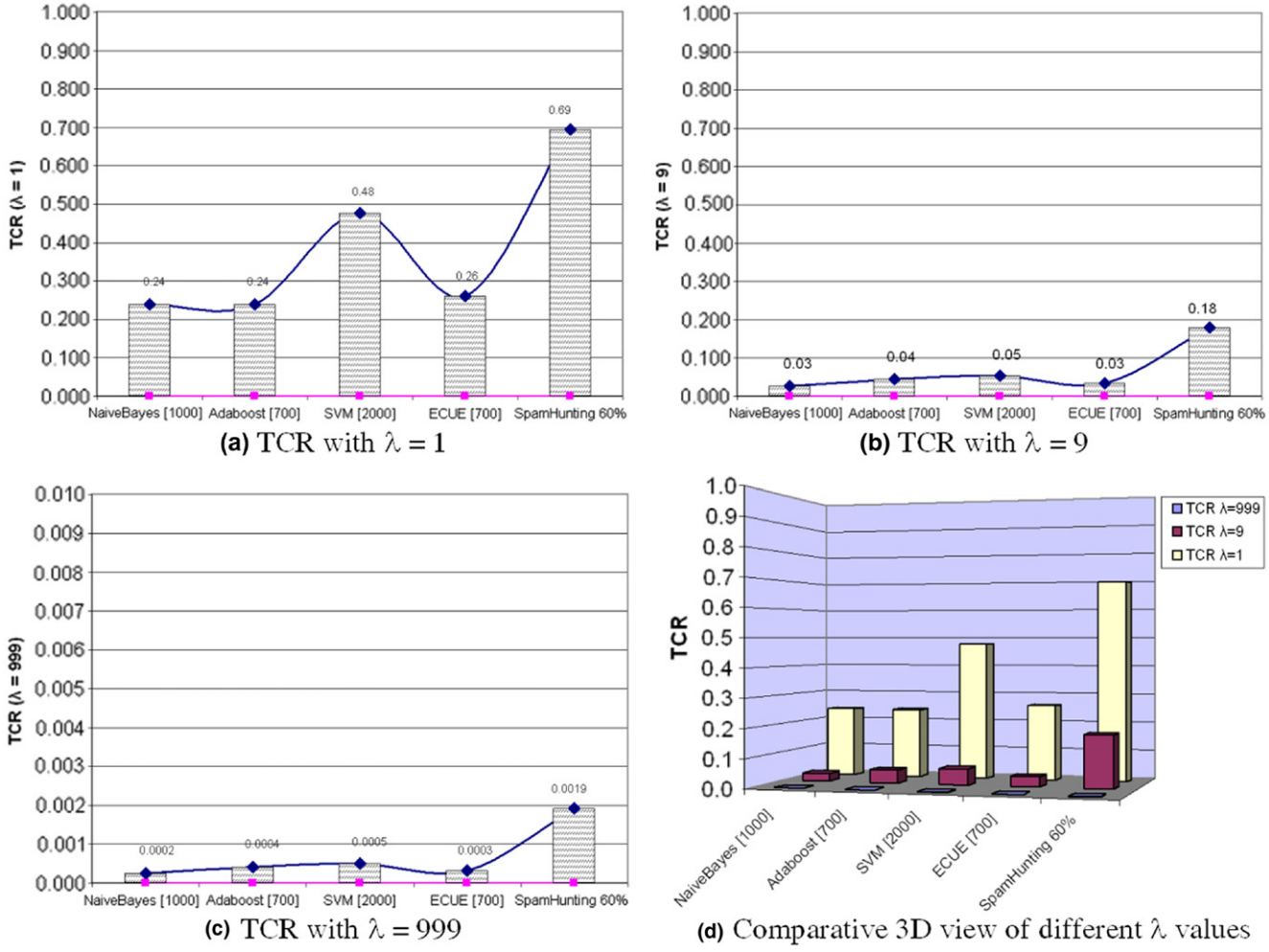

Fig. 8. TCR values for the analyzed models varying the *l* parameter over the Ling-Spam corpus.

6. Conclusions and further work

In this paper we have presented two novel techniques for effectively tracking concept drift in spam domain. On one hand, the RTI (relevant term identification) technique performs a selection of representative terms based solely on the information contained in each e-mail, but weighted with respect to the actual window size. On the other hand, the RMS (representative message selection) technique predicts those e-mails more applicable given the actual context in order to propose an accurate classification. We also show how these techniques can effectively be incorporated into our previous SpamHunting model, a successful implementation of an IBR system for anti-spam filtering.

There are important key benefits of such an approach to spam filtering. Firstly, we have shown that the use of lazy learner algorithms can handle the concept drift inherent in e-mail spam data, allowing for easy updating as new types of spam arrive. Secondly, the instance-based approach to spam filtering allows for the sharing of instances and thus a sharing of the effort of labelling e-mail as spam. Thirdly, an IBR approach facilitates the incorporation of new techniques when they are available without any complex model rebuilding.

We examined various performance aspects of several well-known ML techniques and documented successful anti-spam filters in our thorough investigation. For this purpose we studied and used a variety of measurements in our experiments to report performance. In this sense, the preliminary results obtained from a dynamic evaluation of the analyzed models showed how our SpamHunting system obtains a better ratio between FP and FN errors as well as in the precision score.

Another issue in anti-spam filtering is the cost of different misclassification errors in normal operation. In this sense we tested the performance of our model against other well-known classifiers in three different cost scenarios. Again, the SpamHunting system obtained significantly better results. Our experiments also showed that the real-life computational cost of running a SpamHunting system is always lower than other approaches.

Given the importance of concept drift in spam domain, we implemented the EIRN viewer, an application that allows us to gain a deeper insight related to how our EIRN model stores the information over time. The experiments carried out in this sense showed how our network attains better performance with 60% of the frequency of each message.

In order to simulate sudden concept drift, we also examined the effect of training the models with one corpus and testing their accuracy using another different corpus. Our corpus experiment confirmed that our SpamHunting system outperforms the rest of the analyzed models. We also concluded that lazy learning algorithms perform better than other techniques for anti-spam filtering.

The initial idea that instance-based reasoning systems can offer a number of advantages in the spam filtering domain is backed up by the results obtained from the experiments carried out. Spam is a disjoint concept and IBR classification works well in this domain. In addition IBR systems can learn over time simply by updating their memory with new instances of spam or legitimate e-mail. Moreover, it provides seamless learning capabilities without the need for a separate learning process and facilitates extending the learning process over different levels of learning.

Since SpamHunting is a long-life IBR spam filtering software, a key challenge for us in order to improve our obtained successful results is the development of a policy for instance-base maintenance as in the ECUE system. In this sense, instance editing techniques involve reducing an instance-base or training set to a smaller number of instances while endeavouring to maintain or even improve the generalization accuracy of the system. Moreover, we are working on the definition of a method to be applied in the revise stage of our SpamHunting system in order to maintain various concept definitions. Further work in this area will also include the comparison of our SpamHunting system with the more common ensemble approach to handling concept drift.

References

Aha, D., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning, 6*, 37–66.

Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C. D., & Stamatopoulos, P. (2000). Learning to filter spam e-mail: a comparison of a Naïve Bayesian and a memory-based approach. In *Proceedings of the workshop on machine learning and textual information access, 4th European conference on principles and practice of knowledge discovery in databases*, Lyon, France (pp. 1–13).

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., Paliouras, G. & Spyropoulos, C. (2000). An evaluation of Naïve Bayesian anti-spam filtering. In *Proceedings of the workshop on machine learning in the new information age, in 11th European conference on machine learning*, Barcelona, Spain (pp. 9–17).

Androutsopoulos, I., Paliouras, G., & Michelakis, E. (2004). Learning to filter unsolicited commercial e-mail. Technical Report 2004/2, NCSR "Demokritos". Available from http://www.iit.demokritos.gr/skel/i-config/publications/.

Cohen, W., & Singer, Y. (1999). Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems, 17*(2), 141–173.

Cunningham, P., Nowlan, N., Delany, S. J., & Haahr, M. (2003). A case-based approach to spam filtering that can track concept drift. In *Proceedings of the ICCBR'03 workshop on long-lived CBR systems*, Trondheim, Norway (pp. 115–123).

Daelemans, W., Jakub, Z., van der Sloot, K., & van den Bosch, A. (1999). TiMBL: tilburg memory based learner, version 2.0, Reference Guide. ILK, Computational Linguistics, Tilburg University. Available from http://ilk.kub.nl/~ilk/papers/ilk9901.ps.gz.

Delany, S. J., Cunningham P., & Coyle, L. (2004). An assessment of case-based reasoning for spam filtering. In *Proceedings of fifteenth Irish conference on artificial intelligence and cognitive science*, Castlebar Town (pp. 9–18).

Delany, S. J., Cunningham, P., Tsymbal, A., & Coyle, L. (2004). A case-based technique for tracking concept drift in spam filtering. In *Proceedings of the 24th SGAI international conference on innovative techniques and applications of artificial intelligence*, Cambridge, UK (pp. 3–16).

Delany, S. J., & Cuningham, P. (2004). An analysis of case-base editing in a spam filtering system. In *Proceedings of the 7th European conference on case-based reasoning*, Madrid, Spain (pp. 128–141).

Delany, S. J., Cuningham, P., & Tsymbal, A. (2005). A comparison of ensemble and case-base maintenance techniques for handling concept drift in spam filtering. Technical Report TCD-CS-2005-19, Computer Science Department, Trinity College Dublin.

Drucker, H. D., Wu, D., & Vapnik, V. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks, 10*(5), 1048–1054.

Fallows, D. (2004). Internet users and spam: what the attitudes and behavior of Internet users can tell us about fighting spam. In *Proceedings of the first conference on email and anti-Spam (CEAS)*. Mountain View, CA.

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, in press.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. *Annals of Statistics, 28*(2), 337–374.

Gee, K. R. (2003). Using latent semantic indexing to filter spam. In *Proceedings of the 2003 ACM symposium on applied computing*, Melbourne, FL, USA (pp. 460–464).

Gómez, J. M., Puertas, E., Carrero, F., & De Buenaga, M. (2003). Categorización de Texto Sensible al Coste para el Filtrado de Contenidos Inapropiados en Internet. *Procesamiento del Lenguaje Natural, 31*, 13–20.

Harries, M., & Horn, K. (1995). Detecting concept drift in financial time series prediction using symbolic machine learning. In *Proceedings of the eighth Australian joint conference on artificial intelligence*, Canberra, Australia (pp. 91–98).

Helmbold, D. P., & Long, P. M. (1994). Tracking drifting concepts by minimizing disagreements. *Machine Learning, 14*(1), 27–45.

Hidalgo, J. G., López, M. M., & Sanz, E. P. (2000). Combining text and heuristics for cost-sensitive spam filtering. In *Proceedings of the 4th computational natural language learning workshop*, Lisbon, Portugal (pp. 99–102).

Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of the 14th international conference on machine learning*, Nashville, Tennessee, USA (pp. 143–151).

John, G., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the 11th conference on uncertainty in artificial intelligence*, Montreal, Quebec, Canada (pp. 338–345).

Kelly, M. G., Hand, D. J., & Adams, N. M. (1999). The impact of changing populations on classifier performance. In *Proceedings of the 5th international conference on knowledge discovery and data mining*, New York (pp. 367–371).

Kilander, F., & Jansson, C. G. (1993). COBBIT – a control procedure for COBWEB in the presence of concept drift. In *Proceedings of the European conference on machine learning*, Vienna (pp. 244–261).

Klinkenberg, R. (2004). Learning drifting concepts: example selection vs. example weighting. *Intelligent Data Analysis, special issue on Incremental Learning System Capable of Dealing with Concept Drift, 8*(3), 281–300.

Klinkenberg, R., & Joachims, T. (2000). Detecting concept drift with support vector machines. In *Proceedings of the seventeenth international conference on machine learning* (pp. 487–494). Stanford University.

Klinkenberg, R., & Rüping, S. (2003). Concept drift and the importance of examples. Text mining, theoretical aspects and applications. Physica-Verlag, pp. 55–78.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on artificial intelligence*, Montreal, Quebec, Canada (pp. 1137–1143).

Kubat, M. (1989). Floating approximation in time-varying knowledge bases. *Pattern Recognition Letters, 10*, 223–227.

Kuh, A., Petsche, T., & Rivest, R. L. (1991). Learning time-varying concepts. *Advances in Neural Information Processing Systems, 3*, 183–189.

Kuncheva, L. I. (2004). Classifier ensembles for changing environments. In *Proceedings of the 5th international workshop on multiple classifier systems*, Cagliari, Italy (pp. 1–15).

Lenz, M., Auriol, E., & Manago, M. (1998). Diagnosis and decision support. Case-based reasoning technology. *Lecture Notes in Artificial Intelligence, 1400*, 51–90.

Narendra, K. S., & Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks, 1*, 4–27.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. In Learning for text categorization – papers from the AAAI workshop, Technical Report WS-98-05 AAAI, Madison, Wisconsin (pp. 55–62).

Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C., & Stamatopoulos, P. (2003). A memory-based approach to anti-spam filtering for mailing lists. *Information Retrieval, 6*(1), 49–73.

Schapire, R. E., & Singer, Y. (2000). BoosTexter: a boosting-based system for text categorization. *Machine Learning, 39*(2/3), 135–168.

Schlimmer, J. C., & Granger, R. H. (1986). Incremental learning from noisy data. *Machine Learning, 1*, 317–354.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47.

Smith, R. E. (1987). Diploid genetic algorithms for search in time varying environments. In *Proceedings of the international conference on genetic algorithms and their applications*, Tullahoma, Tennessee, US (pp. 202–206).

Standley, K. O. (2003). Learning concept drift with a committee of decision trees. Technical Report UT-AI-TR-03-302, Computer Sciences Department, University of Texas.

Syed, N. A., Liu, H., & Sung, K. K. (1999). Handling concept drifts in incremental learning with support vector machines. In *Proceedings of the fifth international conference on knowledge discovery and data mining*, San Diego (pp. 317–321).

Taylor, C., Nakhaeizadeh, G., & Lanquillon, C. (1997). Structural change and classification. In *Workshop notes of the ECML-97 workshop on dynamically changing domains: Theory revision and context dependence issues*, Prague, Czech Republic (pp. 67–78).

Ting, K. M. (1998). Inducing cost-sensitive trees via instance weighting. In *Proceedings of the 2nd European symposium on principles of data mining and knowledge discovery*, Nantes, France (pp. 139–147).

Tsymbal, A. (2004). The problem of concept drift: definitions and related work. Technical Report TCD-CS-2004-15, Computer Science Department, Trinity College, Dublin.

Vapnik, V. (1999). *The nature of statistical learning theory* (2nd ed.). *Statistics for engineering and information science*. NJ: Springer.

Watson, I. (1997). *Applying case-based reasoning: Techniques for enterprise systems*. San Mateo, CA: Morgan Kaufman.

Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning, 23*(1), 69–101.