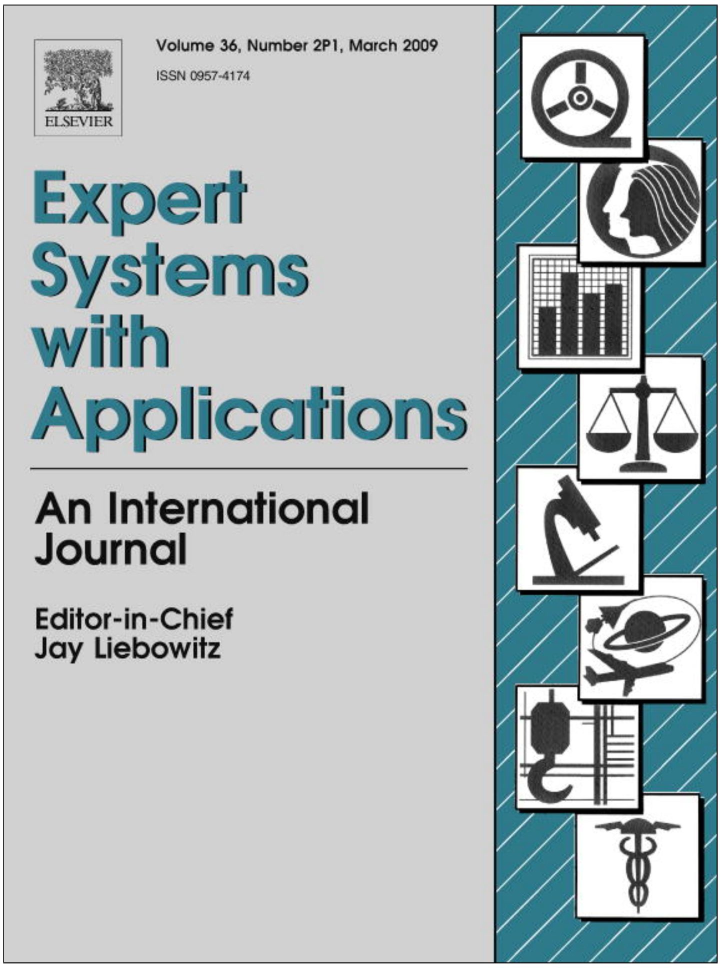

Volume 36, Number 2P1, March 2009
ISSN 0957-4174
ELSEVIER
Expert Systems with Applications
An International Journal
Editor-in-Chief
Jay Liebowitz

Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 36 (2009) 1601–1614

Expert Systems with Applications

www.elsevier.com/locate/eswa

Managing irrelevant knowledge in CBR models for unsolicited e-mail classification ☆

J.R. Méndez [a], D. Glez-Peña [a], F. Fdez-Riverola [a,*], F. Díaz [b], 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

The problem of unsolicited e-mail has been increasing during recent years. Fortunately, some advanced technologies have been successfully applied to spam filtering, achieving promising results. Recently, we have introduced SpamHunting, a successful spam filter able to address the concept drift problem by combining a relevant term identification technique with an evolving sliding window strategy.

Several successful spam filtering techniques use continuous learning strategies to achieve better adaptation capabilities and address concept drift issues. Nevertheless, due to the presence of concept drift and hidden changes in the environment, the presence of obsolete and irrelevant knowledge becomes a serious drawback. Soon after the launch of the filter, many decisions are made based on irrelevant and/or obsolete knowledge. Therefore, in such a situation, the use of forgetting strategies is as important as the implementation of continuous learning approaches.

In this paper we introduce a novel technique designed for identifying and removing the obsolete and irrelevant knowledge that has accumulated over to the passage of time. We have carried out several experiments to test for the suitability of our proposal showing the results obtained and its applicability.

Keywords: Anti-spam filtering; Irrelevant knowledge; Concept drift; EIRN viewer; CBR system

1. Introduction and motivation

One of the most influential advances in recent years has been the extraordinary evolution of the Internet and other communication technologies. During this period, *spammimg*, a new form of fraud, has become prevalent. In this context, any abuse of the communication technologies disturbing other users with bulk and/or undesired content is considered spamming. Nowadays, spam can be found in instant messaging software, newsgroup sites, web search engines, blog sites or mobile short message services.

The most common and well-known form of *spam* is spam e-mail delivery. The increase in the number of spam messages sent across Internet has become alarming. According to the information provided in Rhiolite Software (2006), the amount of spam messages traveling through the Internet is similar to the number of legitimate ones. The great amount of spam messages into the incoming mailboxes of Internet users is restricting the usability of the communication form. As this is the oldest and most problematic form of spam, our work has focused on this area.

In Fdez-Riverola, Iglesias, Díaz, Méndez, and Corchado (2007) we presented a successful spam filtering model called SpamHunting, a Case-Based Reasoning (CBR) system which implements a disjoint knowledge representation

☆ 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: moncho.mendez@uvigo.es (J.R. Méndez), dgpena@uvigo.es (D. Glez-Peña), riverola@uvigo.es (F. Fdez-Riverola), fdiaz@infor.uva.es (F. Díaz), corchado@usal.es (J.M. Corchado).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.11.037

engine able to address concept drift and disjoint category issues (Méndez, Fdez-Riverola, Díaz, Iglesias, & Corchado, 2006). To detect the changes in the significance of a term, we should bear in mind that concept drift takes place when a term characteristic of one class (spam or legitimate) becomes representative of the other Tsymbal et al. (2004). This kind of situations has become increasingly frequent in recent years. For example, some years ago, a message containing terms such as "watch" or "Rolex" would be classified as legitimate. Nevertheless, a short time later, the sales of "Swiss watch replicas" became popular and the Internet e-mail was the best infrastructure for advertising these products. The immediate consequence of this was the delivery of millions of spam messages containing the terms "watch" and "Rolex". Clearly, from the example, the cause of concept drift is often difficult (sometimes impossible) to be adequately handled by models. We know why "watch" and "Rolex" became usual terms belonging to spam messages but, spam filters are not able to detect this issue and successfully adapt their knowledge representation model to evolve with the environment.

The goal of the technique introduced in this work is the identification of terms that are no longer useful for spam filtering. The objective is to remove them from the system knowledge maintaining the quality of the solutions generated by the classifier. When a given term which is only found on messages of a certain class emerges in the opposite category, then the term (affected by concept drift) becomes unhelpful. In these situations, the interpretation of the differences between the probability of finding the affected words in spam and legitimate messages facilitates the identification of irrelevant knowledge that could be discarded.

The main goal of this work could easily be confused with the objective of case-based editing techniques (Delany & Cunningham, 2004; Smith et al., 1998; Smyth et al., 1999). These approaches have been designed to identify instances that have become irrelevant and removing them from system memory. However, our proposal is focused in modifying our message indexing structure when the effects of concept drift are detected. The use of both approaches is not incompatible because the goals and the model adaptations made by these techniques are both completely different and complementary.

The rest of the paper is organized as follows: Section 2 presents previous work on spam filtering domain in general, and particularizes the achievements of memory-based and long-life CBR models; Section 3 introduces our novel technique for irrelevant knowledge detection and removal as well as its integration with our previous SpamHunting system; Section 4 explains the benchmark protocol designed for validating our proposal and remarks the successful results obtained; Finally, Section 5 presents the conclusions and future work.

2. Previous work in spam filtering domain

Although several classical Machine Learning (ML) techniques have been widely used for fighting spam, their usage has so far been discouraging due to the introduction of successful models able to outperform previously achieved performance. This section summarizes previous work on spam filtering techniques ranging from classical ML approaches to modern memory-based systems.

Sahami, Dumais, Heckerman, and Horvitz (1998) is the first and most famous technique used for spam filtering. This kind of spam classifier is 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 from the target message are often included in spam e-mails, but not in legitimate ones, then it would be reasonable to assume that the target e-mail is more likely to be spam. Although there are several approaches for estimating the optimal combination of term probabilities, multinomial Naïve Bayes using boolean attributes seems to be the most effective for spam filtering (Metsis, Androutsopoulos, & Paliouras, 2006).

A common alternative for spam filtering is Adaptive Boosting (AdaBost) (Carreras et al., 2001), an algorithm for constructing an accurate classifier as a combination of weak learners.

SVMs have become very popular in the ML and data mining communities due to its good generalization performance and its ability to handle high-dimensional data through the use of kernels (Vapnik, 1999). They can be used for representing e-mails as points in an n-dimensional space and finding a plane that generates the largest margin between the data points in the positive class and those in the negative class (Druker & Vapmik, 1999).

The main drawback of these classifiers is that they have been mainly designed for a general purpose operation. Recently, some specific domain techniques have been developed including Chung-Kwei (Rigoutsos et al., 2004), a successful proposal based on pattern-discovery from the IBM Research Group.

Nowadays, lazy learning approaches implemented by memory-based systems are the most promising strategies to fight against spam delivery. The next subsection summarizes the most relevant work carried out in this domain.

2.1. Lazy learning approaches

Because of the changing nature of spam, the latest spam filtering tools have been endowed with continuous updating strategies (Fdez-Riverola, Iglesias, Díaz, Méndez, & Corchado, 2007). Following this method, several researchers have suggested that memory-based approaches should work well (Delany, Cunningham, & Coyle, 2004). Case and memory-based methods are characterized by using a knowledge base where each training instance (e-mail) is stored. The usage of indexing structures makes the retrieval of similar messages easier and faster. In operation mode, the retrieved e-mails are used directly for classification purposes. The main advantages derived from the use of instance-based models are its capacity to perform continuous updating, the possibility of managing disjoint concepts and their ability to handle concept drift (Delany et al., 2004; Fdez-Riverola et al., 2007; Méndez et al., 2006). Due to the relevance of these kinds of strategies on the target domain, a brief description of the state-of-the-art has been included in this subsection.

In Androutsopoulos et al. (2000) a preliminary evaluation of memory-based models in the spam filtering domain was shown. In this work, TiMBL software (Daelemans, Jakub, Sloot, & Bosh, 1997) (which implements several memory-based learning strategies) is used for identifying the set of training messages within the *k* closest distances from the target e-mail. Starting from the previous retrieved set of messages, the final class is computed using a voting strategy which gives more priority to legitimate e-mails by means of a weighting process.

Later on, some CBR systems were successfully adapted to spam filtering combining ideas from memory-based models and lazy learning. Case-based reasoning represents a lazy approach to machine learning where induction is delayed until a new problem needs to be solved. In Delany et al. (2004) a CBR system for spam filtering called ECUE (*E-mail Classification Using Examples*) is presented. ECUE can learn dynamically and each message is represented as a case containing a vector of binary features (terms). If the feature exists in the message then the system assigns a value of *true*, otherwise the value of the attribute is set to *false*.

ECUE system uses a *k-nn* classifier to retrieve the *k* most similar cases to a given e-mail. The similarity retrieval algorithm is based on Case Retrieval Nets (CRN) (Lenz, Auriol, & Manago, 1998), which implements a memory structure that allows an efficient and flexible retrieval of stored messages. ECUE classifier uses a unanimous voting strategy able to determine whether a new e-mail is spam or not. To consider a new message as spam, all the returned neighbours need to be classified as spam e-mails. ECUE takes advantage of two case-based editing techniques that make it possible to achieve a long term operation and good performance.

SpamHunting is a lazy learning hybrid model for spam filtering which implements an Instance-Based Reasoning (IBR) lifecycle. In the proposed approach, we use an efficient indexing structure called EIRN (*Enhanced Instance Retrieval Network*) (Fdez-Riverola et al., 2007) to manage the system knowledge. The design of this structure facilitates the indexation of new e-mails taking into account their most relevant terms with different weights, as well as the retrieval of messages containing a given list of relevant terms.

In Fdez-Riverola et al. (2007) we showed some instance representation details of our previous SpamHunting system. In our model, an instance comprises general information extracted from the target e-mail headers (message sender, reply address, date, language and number of attachments) plus a list of pairs $\langle$*term*, *relevancy*$\rangle$. These attributes are computed as described in Méndez et al. (2006) and they have been introduced to include information about the features (words) that best summarize the message content.

The retrieval stage of SpamHunting has been widely described in Fdez-Riverola et al. (2007). It was designed as a projection of the relevant features belonging to the target message over the EIRN network to determine the set of instances sharing at least one relevant feature with the target e-mail. From the retrieved instances, only those with the greatest amount of shared features are selected to accomplish the revise and reuse stages.

The reuse stage of the SpamHunting system has been designed as a unanimous voting procedure due to the performance achieved in previous research work (Delany & Cunningham, 2004). Therefore, a target message will be considered spam only if all the retrieved instances belong to this class. This kind of strategy represents an effective method for reducing the false positive error rate (Delany & Cunningham, 2004; Delany et al., 2004).

The revise stage has been addressed in Méndez et al. (2007). It was designed as a way to compute the quality of the information recovered during the retrieval stage, bearing in mind the main goal of correctly classifying the target message. To implement and test this idea, we have developed a measurement for the compatibility between the target message and the retrieved instances which is based on their common terms. Results achieved from the benchmark test carried out in Méndez et al. (2007) clearly show the good performance of this measure.

The usage of continuous updating strategies takes advantage of the large amount of knowledge accumulated during the system operation. Nevertheless, with the passage of the time, the stored knowledge becomes obsolete and imprecise. Therefore, existing models should use strategies to detect and remove such knowledge. The next subsection presents a summary of previous work on this topic.

2.2. Improving the quality of existing knowledge in CBR systems

A key difference between the editing process and the knowledge adaptation carried out in a CBR system is the granularity of the information managed by each approach. The former is focused on examining cases or instances to remove those that are redundant, contradictory or ambiguous. The later is able to analyze features to drop attributes and/or instances that will not be useful in a given context. In this subsection previous work on case-base maintenance is summarized.

In Smyth et al. (1995) a preliminary work on case-base editing has been introduced. This work presents a case deletion strategy able to reduce the amount of stored data keeping all the available knowledge. This means that the competence of the CBR system (the range of problems it can solve) is not damaged. This work introduces the definition of Coverage and Reachability sets for a case *c*. The former is defined as the set of target problems (cases stored in the system memory) that can be solved using *c* while the later contains all target cases that are helpful to solve *c*. It also identifies four different types of cases: (i) pivotal, (ii) auxiliary, (iii) spanning, and (iv) support cases.

In the work of Smyth et al. (1996) the *utility problem* (present in long-life systems due to over-fitting) was identified as the cause of efficiency reduction and the achievement of poor quality solutions. This work motivates the usage of case-base maintenance techniques such as those introduced in Smyth et al. (1995). Later on, in Smyth (1998) the basis of case-base maintenance was established and the need of modeling the competence of case bases was introduced. In Smith et al. (1998) a model for the competence of case-based systems was introduced. The competence or coverage concepts represent a measurement of the range of target problems that a given system can solve. Finally, in McKenna and Smith (2000a, 2000b) a large collection of editing techniques was introduced and compared.

ECUE was the first model to use case-base maintenance techniques to track concept drift (Delany & Cunningham, 2004; Delany, Cunningham, Tsymbal, & Coyle, 2004). It includes two case-base editing techniques used to remove duplicate knowledge and inconsistent information from the system memory (Delany & Cunningham, 2004). The proposed algorithms are based on combining the coverage, liability and reachability sets of the stored cases. As explained in Delany and Cunningham (2004), by combining the knowledge sets mentioned, the superfluous instances can be identified and removed preserving the system performance (CRR, *Conservative Redundancy Removal*). Moreover, instances that introduce noise in the classification process are also recognized and dropped (BBNR, *Blame-Based Noise Removal*).

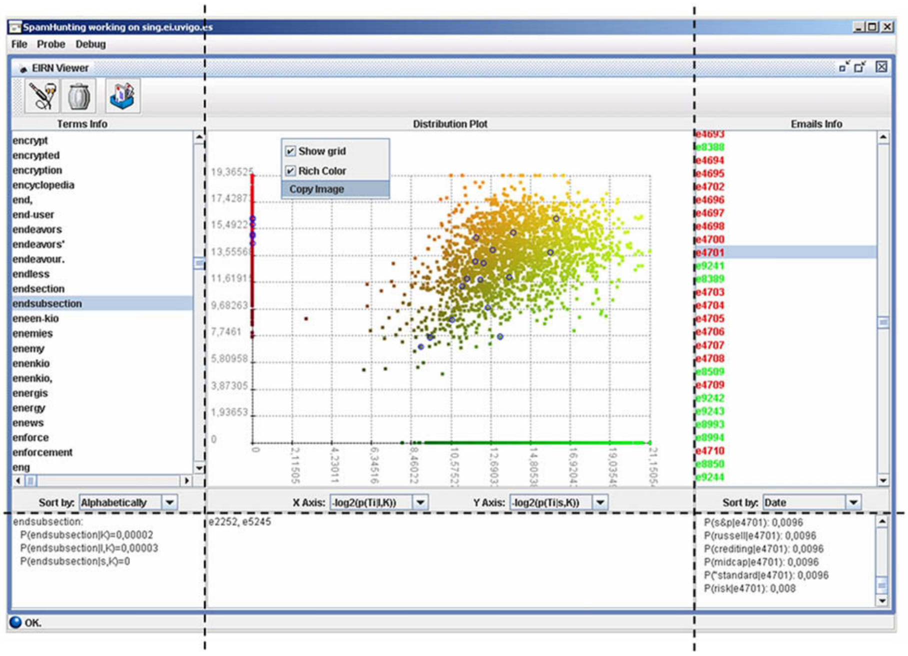
Although we have several years of experience working with CBR in the spam filtering domain, we have not yet addressed how we can solve the removal of irrelevant knowledge produced by the use of continuous updating strategies. This important question is discussed in Section 3, including a deeper description of our irrelevant knowledge identification and removal technique proposed in this study.

3. Detecting irrelevant knowledge in SpamHunting

As we previously mentioned, one of the most significant problems in spam filtering is the presence of concept drift (Tsymbal et al., 2004; Widmer & Kubat, 2001). In this section we introduce a novel technique able to identify and discard irrelevant and spurious knowledge from the memory of our previous successful SpamHunting system. Our proposed approach is based on a practical point of view of concept drift and its utilization is compatible with previous case-base editing techniques. After the introductory explanation, a new formal technique is suggested (Section 3.2) based on the practical ideas previously exposed (Section 3.1) and taking into consideration the inner characteristics of our SpamHunting system.

One of the most relevant features of our EIRN structure is the possibility of generating different views of the underlying indexed knowledge. In Fdez-Riverola et al. (2007) we presented this functionality although its benefits have not yet been tested. Fig. 1 shows a screenshot of the software developed for taking advantage of this capacity.

As we can observe from Fig. 1, the application frame is structured into six panels. First, on the top of the left column, we can find all the relevant terms that the EIRN network uses for indexing existing instances. Under this panel, a section is located that summarizes some statistics



encrypt
encrypted
encryption
encyclopedia
end,
end-user
endeavors
endeavors'
endeavour.
endless
endsection
endsubsection
eneen-kio
enemies
enemy
enenkio
enenkio,
energis
energy
enews
enforce
enforcement
eng

Sort by: Alphabetically

X Axis: -log2(p(Ti|l,K))

Y Axis: -log2(p(Ti|s,K))

e4693
e8388
e4694
e4695
e4702
e4696
e4697
e4698
e4700
e4701
e9241
e8389
e4703
e4704
e4705
e4706
e4707
e4708
e8509
e4709
e9242
e9243
e8993
e8994
e4710
e8850
e9244

Sort by: Date

endsubsection:
P(endsubsection|K)=0,00002
P(endsubsection|l,K)=0,00003
P(endsubsection|s,K)=0

e2252, e5245

P(s&p|e4701): 0,0096
P(russell|e4701): 0,0096
P(crediting|e4701): 0,0096
P(midcap|e4701): 0,0096
P("standard|e4701): 0,0096
P(risk|e4701): 0,008

Fig. 1. EIRN viewer plugin of our SpamHunting system.

referring to the selected terms including: (i) probability of finding the term in the case base, (ii) frequency of finding the term in spam e-mails and (iii) probability of finding the term in legitimate messages.

At the top of the right column, we can find all the instances stored in the system memory. Under this panel, an instance statistics panel can be found where all the relevant terms belonging to the selected messages as well as their frequencies are showed.

At the top of the central column, we have placed a plot representing the relevant terms indexed in our EIRN model. The plot panel has been developed with built-in clipboard copy support, "save as image" capability and drag and drop compatibility for use with any image application or word processor. Both selected terms belonging to the left panel or part of a message from the right panel are always highlighted in our graphic representation.

The EIRN viewer module represents each term as a two-dimension point on the plot. The coordinates of each point are computed according to the function selected in the combo boxes placed under the plot. The following measurements are available for each coordinate: (i) the probability of finding the term t in spam e-mails stored in the system memory, $p(t|s,K)$, (ii) the logarithmic form of the previous value, $-\log_2(p(t|s,K))$, (iii) the probability of finding the term t in legitimate messages, $p(t|l,K)$, (iv) the logarithmic form of the previous value, $-\log_2(p(t|l,K))$, and (v) the probability of finding the term t in the system memory, $p(t|K)$.

To select the terms which are considered relevant for each message, SpamHunting uses a measurement called AI (*Achieved Information*) (Méndez et al., 2006). This measurement has been designed for estimating the relevancy of each term identified in a message to compute its classification. It combines information concerning the importance of each term belonging to the given message and its ability to distinguish between spam and legitimate classes. Expression (1) shows this measure.

$$AI(w,e|t) = P(w \wedge e) \cdot \left[1 - \frac{1}{\text{length}(w)}\right] \cdot \left[\frac{|P(w \wedge S|t) - P(w \wedge L|t)|}{P(w \wedge S|t) + P(w \wedge L|t)}\right] \qquad (1)$$

where $P(w \wedge e)$ represents the frequency of term w in the message e, length(w) stands for the length of the word w and finally, $P(w \wedge S|t)$ and $P(w \wedge L|t)$ are, in that order, the probability of finding the word w in spam and legitimate messages stored in the system memory.

One of the most relevant issues about the AI measurement is the presence of the variable t that represents the passage of the time. The AI value for a term w can change over time due to the effects of concept drift. Therefore, the variable t represented in Expression (1) stands for the exact moment in which a given message arrives.

When a new e-mail is presented to our SpamHunting system, its terms are tokenized by using blank chars. Then, each identified word is rated using the AI measurement and the set of most relevant terms for the target e-mail is computed as follows. From the words identified in the target message, we select the smallest set from the highest rated ones whose quantity of AI exceeds of a certain percentage p of the total AI (achieved by all the words belonging to the e-mail). Starting from previous empirical evaluations, we have observed that values near 60% are good choices for p (Méndez et al., 2006).

Once the available visualization software has been presented, we can introduce a visual procedure for identifying at a glance the presence of concept drift and irrelevant knowledge in the EIRN plot. The next subsection contains a detailed description of the process as well as some practical examples.

3.1. Tracking down unworthy terms through a quick survey of the EIRN plot

The utility of the EIRN viewer plugin is the identification of spurious knowledge derived from concept drift effects through a quick look over the graphic representation. In this subsection we show how to visually identify irrelevant knowledge that has been affected by concept drift.

Concept drift can be explained as a change in the environment of the problem caused by hidden issues of the domain (Widmer & Kubat, 2001). These changes often alter the underlying distribution probabilities of some features considered by the models. Therefore, those features affected by concept drift should be detected and discarded. In the spam filtering domain, the effects of concept drift cause a term representative of a certain class (spam or legitimate) to appear in the opposite class. As a consequence, each feature used for indexing existing messages should be carefully examined when the probabilities of finding it in spam and legitimate classes change over the time.

As we have previously explained, the difference between the probability of finding a term in spam messages and legitimate ones can be easily monitored with our EIRN viewer module. Moreover, we have contemplated the possibility of displaying terms as points using their base 2 logarithm representations to generate a scattered plot. These properties of the EIRN viewer plugin will be used to identify irrelevant information stored by the model.

To detect modifications on the underlying probabilities of existing terms using our EIRN viewer plugin, they must be represented as points in a 2D plane where their coordinates are computed using Expression (2).

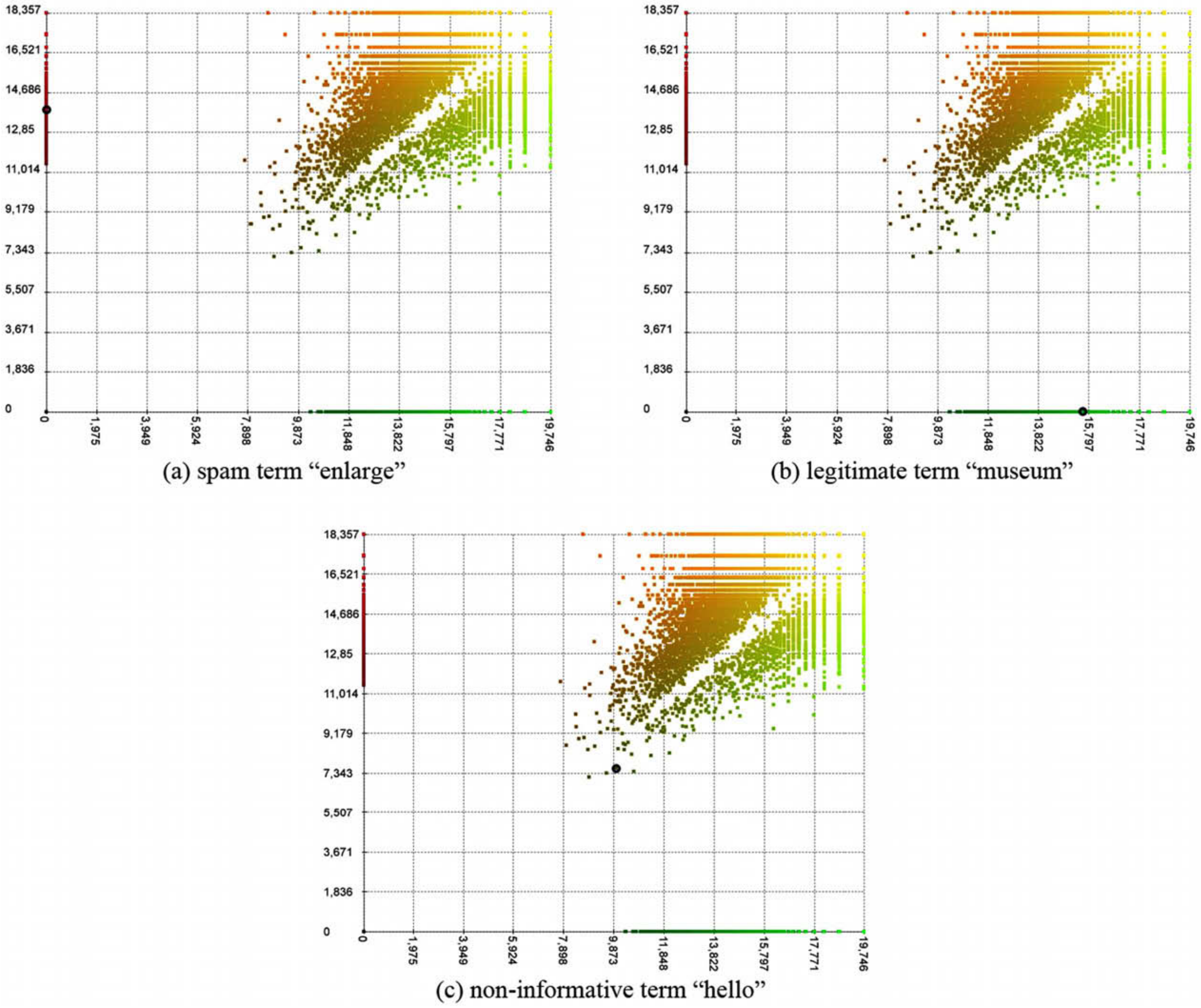
$$R(t_i) = [-\log_2(p(t_i|s,K)), -\log_2(p(t_i|l,K))] \qquad (2)$$

where $p(t_i|s,K)$ and $p(t_i|l,K)$ are in that order, the probability of finding the term t_i in spam and legitimate messages, and $R(t_i)$ stands for the coordinates of the term t_i. As Expression (2) shows, the y coordinate of each term is indicative of the probability of finding it in spam messages, whereas the x coordinate is indicative of the probability of finding it in legitimate e-mails.

As an example, the term "enlarge" is currently included in thousands of spam messages received by Internet users. Therefore, the probability of finding it in spam messages is very high while the probability of finding it in legitimate ones is very low. Consequently, the representation of the term"enlarge" is closer to the y axis than the x axis. Moreover, considering the representation of the term "museum" that appears in lots of legitimate messages in academic environments, we can state that it will be represented closer to the x axis than the y axis. Both "enlarge" and "museum" terms are useful for spam filtering because they are very indicative of a certain class. Finally, an irrelevant term for spam filtering such as the word "hello" will appear with a similar frequency in spam and legitimate messages. Therefore, the distance of the point representing the term "hello" to the x and y axis is going to be similar. This means that the term "hello" will be plotted near to the bisector of the angle comprised between the x and y axis. Fig. 2 shows a graphic representation of these ideas.

From these findings, it can be seen that those terms near the bisector are unhelpful for spam classification and can be safely removed. The representation of each term is unceasingly changing and needs to be taken into consideration by continuous updating strategies implemented in future spam filtering systems. These movements are indicative of the constantly changing environment of the spam filtering domain. In this work, we introduce the study of the position of a term as a technique for identifying the presence of concept drift. Fig. 3 shows some interesting screenshots of a real EIRN network trained with messages received during years 2004 and 2005. Each screenshot has been captured highlighting a different term using a big circle.

As we can see from Fig. 3, terms that are unhelpful for classifying e-mails can be found using this system. An illustrative example of this situation is given by the terms "replica" and "watch" that were representative of legitimate messages until a new fraudulent product was advertised through Internet: the "Swiss watches replicas". Starting from that moment, the above mentioned terms moved

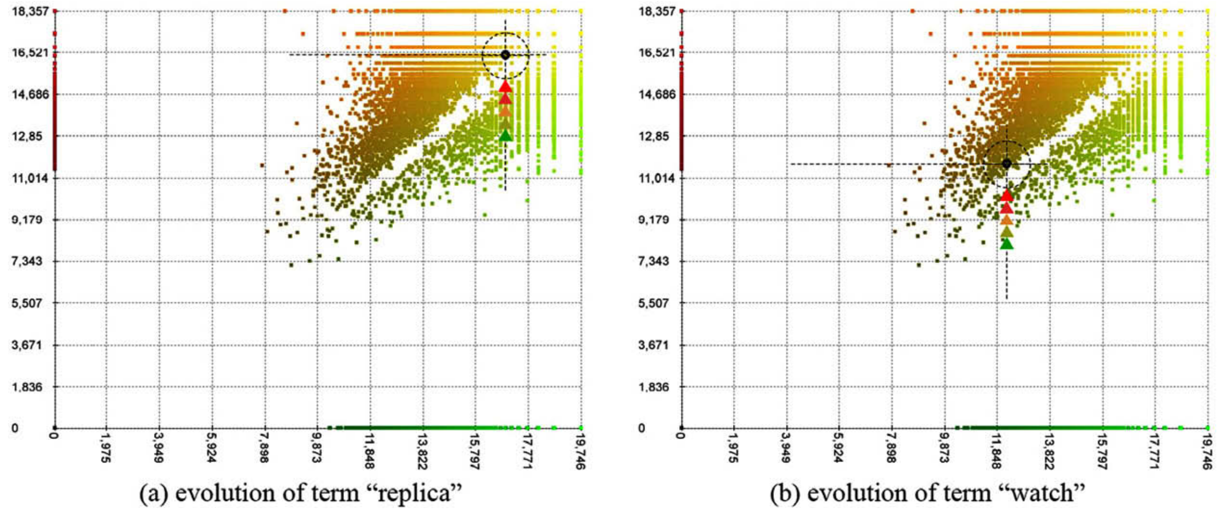


(a) spam term "enlarge"

(b) legitimate term "museum"

(c) non-informative term "hello"

Fig. 2. Screenshots of the EIRN viewer plugin for terms representation.



(a) evolution of term "replica"

(b) evolution of term "watch"

Fig. 3. Tracking concept drift by detecting sliding points in the EIRN viewer pluging.

from an initial position over the x axis to a new location near the bisector of the angle between the coordinate axes. This event, that took place during the years 2004 and 2005, clearly shows the adverse effects of concept drift over the information managed by spam filtering models. As it can be sensed, when a term falls in this kind of situation it is completely unhelpful (and sometimes confusing) for spam classification.

Obviously, there is a strong relationship between the appearance of concept drift, irrelevant terms, and the environment of the final user. First, concept drift is the cause of irrelevant knowledge, while the amount of irrelevant knowledge is dependent on the user environment. For instance, the term "Viagra" should not be used for spam filtering in a drug-company. We think that filtering these kinds of messages will be a difficult task within these environments, although this situation should not be an unsolvable problem. If all messages containing the word are successfully classified in other environments, this kind of spam messages should disappear because drug-companies are not a good choice for selling illegal drugs.

In this study, we suggest the elimination of terms situated near the bisector of the angle between the coordinate axes because they are considered unhelpful and irrelevant for the task of e-mail classification. These terms will be different depending on the execution context of the filter and its users. As we can observe from Fig. 3 plots, the proposed representation of the EIRN data is helpful for identifying concept drift and unhelpful terms. Based on the ideas reported previously, a formal method for addressing the automatic removal of unhelpful and irrelevant terms is introduced in next subsection.

3.2. Formal garbage term identification and removal algorithm

In this section a novel methodological approach for irrelevant knowledge identification is presented and formalized. As we can ascertain from the ideas mentioned above, the difference between the frequencies of finding a term in spam and legitimate messages can be successfully used to detect irrelevant terms. A small difference between these frequencies shows poor relevance while large differences are indicative of suitable terms for spam classification.

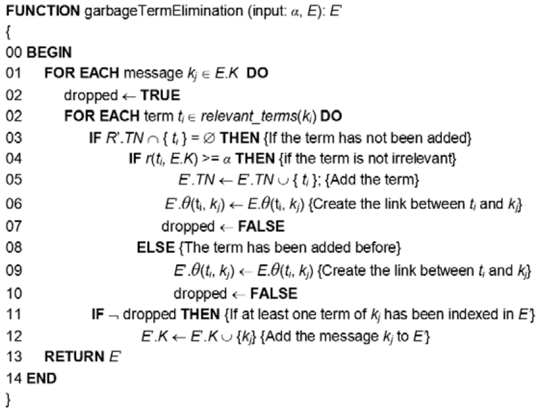
To prevent the elimination of terms that are present in a low number of messages, studies into spam and legitimate frequencies for a given term must be normalized using the global term frequency. Expression (3) formalizes these ideas proposing a new measurement $r(t_i)$ that represents the relevancy of the term t_i.

$$r(t_i) = \frac{|p(t_i|l,K) - p(t_i|s,K)|}{p(t_i|l,K) + p(t_i|s,K)} \tag{3}$$

where $p(t_i|l,K)$ and $p(t_i|s,K)$ are in that order, the frequency of the term t_i in legitimate and spam messages. The relevance of a term is a value between 0 and 1 that can be explained as follows: lower relevance values (near 0) indicate irrelevance while terms with higher relevance values are the most suitable for spam classification. From another point of view, terms with a relevance value of 0 are over the bisector of the angle between the coordinate axes, while those with maximum relevancy are placed over the coordinate axes.

The closeness of terms to the coordinate axes bisector can be used as the main criterion for detecting irrelevant knowledge. Therefore, we suggest a cut factor rate α that works as a minimum relevance threshold. As an example, if a 30% value is selected for α parameter, all terms t_i having a relevance value $r(t_i)$ lower than 0.3 will be removed.

To apply the proposed technique to our SpamHunting system, we need to define our EIRN network. Expression (4) characterizes our indexing structure as a set of node terms (NT) which are related to the system memory (K) by means of a collection of relationships (θ).



```
FUNCTION garbageTermElimination (input: $\alpha$, $E$): $E'$
{
BEGIN
  FOR EACH message $k_j \in E.K$ DO
    dropped ← TRUE
    FOR EACH term $t_i \in relevant\_terms(k_i)$ DO
      IF $R'.TN \cap \{ t_i \} = \varnothing$ THEN {If the term has not been added}
        IF $r(t_i, E.K) >= \alpha$ THEN {if the term is not irrelevant}
          $E'.TN \leftarrow E'.TN \cup \{ t_i \}$; {Add the term}
          $E'.\theta(t_i, k_j) \leftarrow E.\theta(t_i, k_j)$ {Create the link between $t_i$ and $k_j$}
          dropped ← FALSE
        ELSE {The term has been added before}
          $E'.\theta(t_i, k_j) \leftarrow E.\theta(t_i, k_j)$ {Create the link between $t_i$ and $k_j$}
          dropped ← FALSE
    IF $\neg$ dropped THEN {If at least one term of $k_j$ has been indexed in $E'$}
      $E'.K \leftarrow E'.K \cup \{k_j\}$ {Add the message $k_j$ to $E'$}
  RETURN $E'$
END
}
```

Fig. 4. Garbage term identification and removal algorithm.

$$E(\mathrm{NT}, K, \theta)\begin{cases} K = \{k_1, k_2, \ldots, k_n\} \\ \mathrm{NT} = \{nt_i \mid \exists k_j \in K \mid \mathrm{ord}(t_i, k_j) > 0\} \\ \forall nt_i \in \mathrm{NT},\ k_j \in K, \theta(nt_i, k_j) = \mathrm{ord}(t_i, k_j) \end{cases} \tag{4}$$

where $\mathrm{ord}(t_i, k_j)$ is the position of the term t_i in the term list of the instance k_j ordered by relevance and nt_i is the node that represents the term t_i. This structure can be described as a 3-value list that includes the following information: (i) a set of message instances K, (ii) a set of terms, NT, that are relevant at least in one instance and (iii) a weighted relationship, θ, between each relevant term and the instances which represent. The value of the relationship $\theta(nt_i, k_j)$ between a term nt_i and a instance k_j represents the importance of t_i in the subject matter of the instance k_j.

Considering the EIRN structure, Fig. 4 shows the algorithm used to create a modified EIRN structure, E', as a result of the elimination of irrelevant knowledge from a source EIRN, E, using an α cut factor rate.

As illustrated in Fig. 4, the elimination of irrelevant knowledge is a light process. We suggest using this strategy to improve the performance of our SpamHunting system working in continuous updating mode. To test the suitability of our proposal, we have designed and executed a benchmark procedure. The next section comprises a deep description of the whole evaluation process as well as its execution results.

4. System evaluation

To test the convenience of the introduced approach, some relevant decisions must be taken. First, we need to select some datasets, as well as pertinent measurements and different models to compare them. Then, the experimental protocol needs to be established in order to select the steps and tools used during each phase. This section contains a description of the benchmark procedure used to test the proposed technique. Moreover, some relevant parameters and datasets (corpus) are also discussed.

The evaluation process has been divided into two sequential steps: (i) parameter optimization and (ii) benchmark procedure. During the first of these we have carried out several experiments to determine the best value for α parameter. The next step focused on carrying out a comparison between two well-known memory-based models: ECUE and SpamHunting systems.

This section has been structured as follows: Section 4.1 contains information about the datasets, measurements, models and their configuration details. Section 4.2 introduces the optimization stage and its outcomes and finally, Section 4.3 presents the benchmark stage, as well as the results achieved during its execution.

4.1. Corpus selection and miscellaneous configuration issues

The availability of public and representative datasets is a problem in the vast majority of research domains. This subsection introduces the main obstacles for the construction of spam filtering databases, as well as the difficulties in ensuring public accountability of the results obtained. Moreover, we present some publicly available corpus and discuss their main features. Finally, we justify the corpus selection for carrying out our experiments.

In general, spam messages can be considered as public because no personal information is included in them. Nevertheless, several privacy issues should be taken into consideration for the public sharing of legitimate messages. A fraudulent use of them can infringe the privacy rights of their owners. This fact may explain the great amount of public spam collections, as well as the reduced number of corpus containing legitimate messages.

To overcome these difficulties, some researchers distribute their messages as a list of feature identifiers instead of the original terms (Androutsopoulos, Paliouras, & Michelakis, 2004). The main disadvantage of this method is the impossibility of executing some pre-processing steps such as text extraction or tokenizing schemes.

Moreover, when a new technique is introduced, the experimental protocol should be clearly described to guarantee its reproduction by any user. Therefore, only publicly available datasets should be used in most of the research experiments. Fortunately, this kind of issue can easily be addressed by the use of some publicly available corpora such as the one constructed by the SpamAssassin team.[1] This initiative shares an extensive corpus containing more than nine thousands messages in EML format (RFC-822 (Crocker, 1982)). Another excellent alternative is the usage of the Ling-Spam corpus[2] containing more than two thousands messages shared as a collection of plain text in tokenized files. Although these corpuses do not contain message headers, they can be used for the execution and testing of anti-spam filters.

The main limitation of these public datasets is the low number of messages. A great amount of available e-mails

[1] Available at http://apache.spamassassin.org/.

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

facilitates the observation of the effects caused by the presence of irrelevant knowledge. Moreover, we should bear in mind the source of the Ling-Spam dataset, which has been extracted from a public linguistics e-mail distribution list. As a consequence, the subject matter of all legitimate messages belonging to this corpus is relative to Linguist issues.

To carry out the optimization of the cut-off threshold α, we will use a set of messages compiled since the year 2004 from several students and teachers belonging to our University (the SING corpus). This dataset contains a great amount of spam and legitimate e-mails. Finally, due to its large amount of messages, we found SpamAssassin as a good choice for the comparison of different spam filtering models. Table 1 summarizes the main characteristics of the above mentioned corpuses.

As seen in Table 1, the SpamAssassin corpus is an old dataset while our corpus has been mainly constructed using recent e-mails. In addition to this, the SING corpus contains a large amount of messages. The current disadvantage of our corpus is the presence of private messages that should be carefully handled to preserve the identity of their senders/receivers. Therefore, we think that SING corpus is a good choice only for optimizing purposes while SpamAssassin should be used for model analysis.

Seven well-known measurements have been used for comparison purposes including: (i) percentage of correct classifications, false positive and false negative error rates, (ii) batting average (Graham-Cumming, 2004), (iii) precision and recall, (iv) *F*-score (Rijsbergen, 1979), (v) balanced *F*-score (Shaw, Burgin, & Howell, 1997; Yang, 1999), (vi) Total Cost Ratio (TCR) (Androutsopoulos, Koustias, Chandrinos, Paliouras, & Spyropoulos, 2000) and (vii) Receiver Operating Characteristic (ROC) curves (Egan, 1975).

To guarantee the quality of our experimental results, we have used a 10-fold stratified cross-validation for all experiments (Kohavi, 1995). The results reported are the means of the 10-folds. With the goal of demonstrating the suitability of our SpamHunting system working with the proposed technique in dynamic environments, we have compared its performance level with the results achieved by the ECUE model (Delany et al., 2004), a well-known lazy learning spam filter.

Finally, talking about the SpamHunting configuration, the percentage of AI selected from each message has been set to 60%, as it has proved to be a reasonable value (Méndez et al., 2006). In addition to this, Section 4.2contains information about the benchmark protocol used for optimizing α parameter as well as the results obtained.

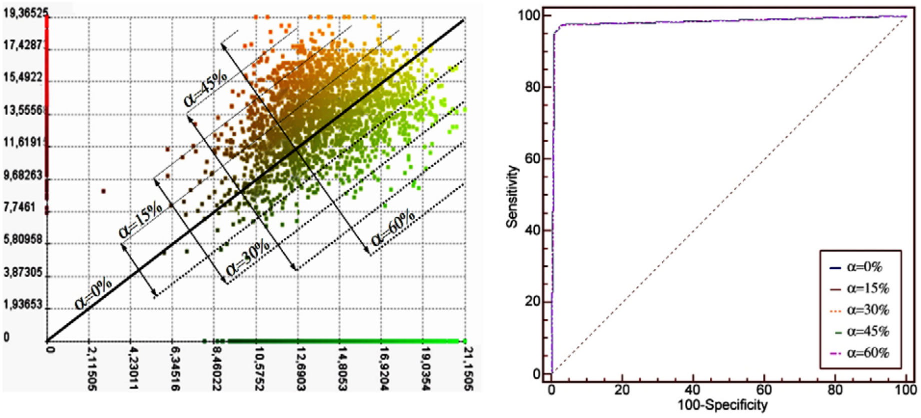
4.2. Optimization benchmark using the SING corpus

At the first stage of the experimental phase we are going to execute some tests to optimize the parameters of the technique introduced in this paper. This subsection presents the benchmark protocol designed for the α parameter optimization and the results achieved during the process.

We have tested different values for α threshold starting in 0 (no garbage removal) using steps of 0.15 up to 0.60 (see Fig. 5a). We have applied these SpamHunting configurations over our SING corpus containing 20,130 messages and have analyzed the following measurements: (i) ROC

Table 1
Comparison of some available datasets for spam filtering domain

	SpamAssassin	Ling-spam	SING
Spam messages amount	6951 (74.5%)	481 (16.6%)	7903 (39.3%)
Legitimate messages amount	2381 (25.5%)	2412 (83.3 %)	12,227 (69.7%)
Time interval	2001–2002	No available info about dates	2004–2006
Distribution form	EML (RFC 822 Compliant)	Text plain files	EML (RFC 822 Compliant)/XML



(a) different values for the α parameter

(b) ROC curves plot for analyzing the behavior of SpamHunting over the SING corpus

Fig. 5. EIRN viewer aspect and SpamHunting behavior using different values for the α parameter.

curves, (ii) percentages of false positive, false negative and correct classifications, (iii) recall and precision, and (iv) TCR for λ values of 1, 9 and 999. The main goal of this experiment is the identification of the best value for α parameter. Moreover, we have also analyzed the effects of garbage collection over the generated solutions.

Fig. 5b shows the ROC curve plot for the analyzed configurations. Since the differences between the ROC curves are unappreciable on the plot, we have executed a statistical test assuming the equality of the area under them as the null hypothesis. Table 2 shows the *p*-values achieved by the statistical test executed to compare the ROC curves for the different values of α parameter.

As we can see from Table 2, there are no statistical significant differences between discarding some irrelevant terms (specially for $\alpha = 0.15$ and $\alpha = 0.30$) or maintaining all the information existing in the CBR memory ($\alpha = 0$). This fact supports the idea that the proposed technique notably contributes to reduce the memory size of the case base while maintaining the classification accuracy of the system.

To obtain more detailed information about the analyzed configuration and to determine the best value for α parameter, we have used the percentage of correct classification and error rates, recall, precision and TCR measures. The results are summarized in Table 3.

As we can see from Table 3, there are no meaningful differences between the achieved results. From the outcomes of classical measurements and the ROC analysis, we believe that values smaller than 0.30 should be considered as the best for α parameter.

Once we have established the best interval for α, we have compared two well-known continuous updating CBR systems. The next subsection shows the benchmark protocol for analyzing ECUE and SpamHunting systems using a public well-known corpus as input dataset.

Table 2
Statistical analysis of the ROC curves for different values of α parameter

α	0	0.15	0.30	0.45	0.60
0		0.952	0.905	0.277	0.175
0.15			0.917	0.263	0.170
0.30				0.184	0.140
0.45					0.416
0.60					

Table 3
Percentage of correct classifications, FP and FN errors, recall, precision and TCR values from validation over the SING corpus

α	% OK	% FP	% FN	Recall	Precision	TCR		
						$\lambda = 1$	$X = 9$	$\lambda = 999$
0	95.936	0.482	3.582	0.909	0.986	9.773	5.177	0.092
0.15	95.936	0.482	3.582	0.909	0.987	9.773	5.177	0.092
0.30	95.931	0.482	3.587	0.909	0.987	9.762	5.174	0.092
0.45	95.876	0.502	3.622	0.908	0.986	9.582	4.963	0.086
0.60	95.876	0.472	3.652	0.907	0.987	9.614	5.151	0.092

4.3. Comparison with ECUE system using the SpamAssassin corpus

This subsection presents a comparison between two successful CBR spam filters using continuous updating strategies. To carry out the experiments, we have selected the publicly available corpus from the SpamAssassin project using the following measurements: (i) percentages of correct classification, false positive and false negative errors, (ii) batting average, (iii) recall and precision (iv) *F*-score and balanced *F*-score with β values of 1.5 and 2, (v) TCR measurements using $\lambda = 1$, 9 and 999, and (vi) ROC curves.

To compare ECUE (Delany et al., 2004) and SpamHunting (Fdez-Riverola et al., 2007) systems, we have used a continuous updating approach. Therefore, every time a new message is classified, it is included in the system memory containing the solution computed by the classifier. We have assumed the absence of user corrections in the outcome of the systems. The internal operation of the selected models in relation with their editing strategies is as follows: ECUE has been configured for executing BBNR and CRR algorithms every 100 classifications whereas SpamHunting uses the proposed garbage collection technique every 100 classifications.

Fig. 6 shows a comparison of three different configurations of SpamHunting system and ECUE model measuring false positive, false negative and correct classification percentages.

As we can see from Fig. 6, SpamHunting gets the best performance when comparing its results against ECUE using any of the analyzed configurations. Moreover, we highlight the small amount of false positive (FP) errors achieved when our system is applied. The discrepancies using the different analyzed configurations over our SpamHunting system (without editing, $\alpha = 0.15$ and $\alpha = 0.30$) are imperceptible.

From another point of view, we have compared the batting average score obtained by the three analyzed configurations plus the ECUE system. Table 4 shows the two components of the batting average measure.

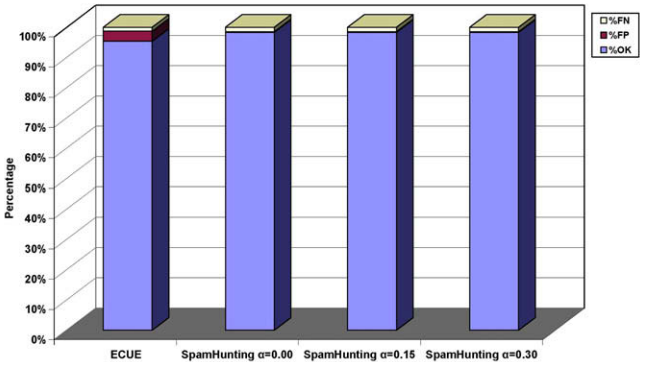


Fig. 6. Percentage of correct classifications, FP errors and FN errors from validation over the SpamAssassin corpus.

Table 4
Bating average results from validation over the SpamAssassin corpus

	ECUE	SpamHunting		
		$\alpha = 0.00$	$\alpha = 0.15$	$\alpha = 0.30$
Hit rate	0.957	0.942	0.942	0.941
Strike rate	0.033	0.003	0.003	0.002

As we can see from Table 4, ECUE is able to detect a greater amount of spam messages but increases the amount of FP errors. Moreover, SpamHunting is able to effectively reduce the amount of this kind of harmful misclassification.

From another perspective, we have analyzed recall and precision scores achieved by the evaluated models. Fig. 7 presents a graphical view of the results obtained.

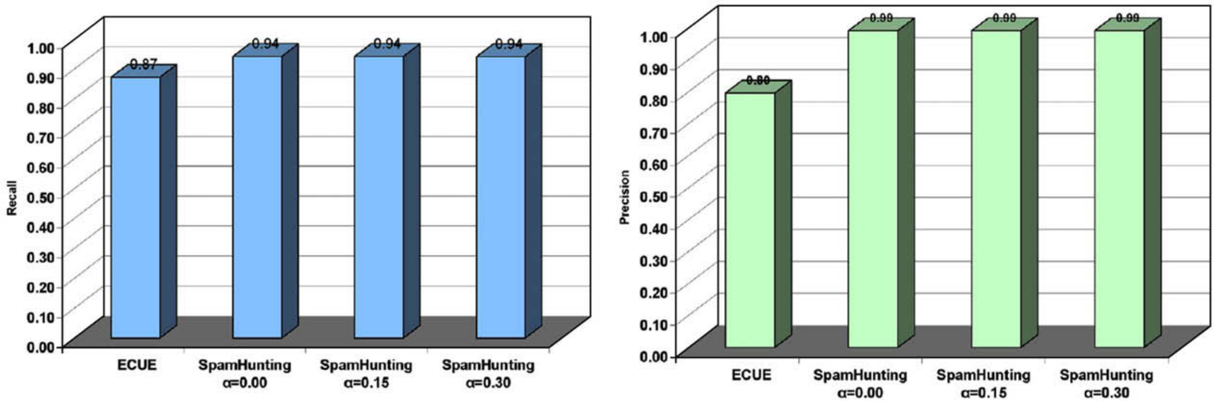
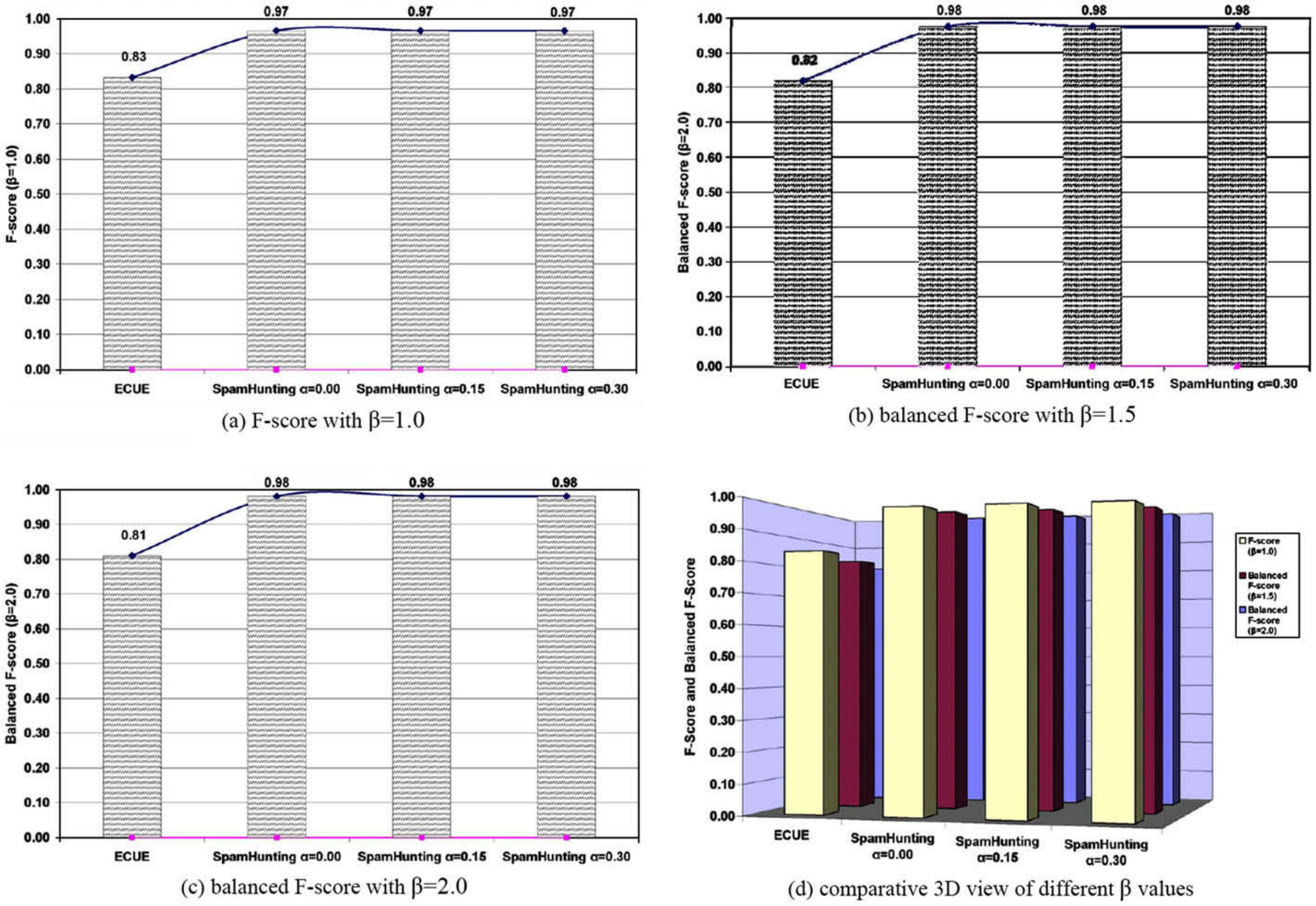


Fig. 7. Recall and precision values for the analyzed models.



(a) F-score with β=1.0

(b) balanced F-score with β=1.5

(c) balanced F-score with β=2.0

(d) comparative 3D view of different β values

Fig. 8. F-score ($\beta = 1.0$) and balanced F-score for the analyzed models.

From Fig. 7, recall and precision measurements show better performance when using the SpamHunting system. Moreover, there are no significant differences between the three analyzed variants. However, precision and recall measures should not be used independently because they represent complementary information (Shaw et al., 1997). In this sense, *F*-score and balanced *F*-score have been introduced to effectively combine the information achieved by the original measures. To introduce an enhanced representation of recall and precision, Fig. 8 includes a study of *F*-score ($\beta = 1$) and balanced *F*-score (with $\beta = 1.5$ and $\beta = 2$).

As we can see from Fig. 8, SpamHunting achieves a superior performance to ECUE working with continuous updating mode in all the analyzed scenarios. The large amount of FP errors achieved by ECUE can be estimated from the reduction of balanced *F*-score when the precision is important (using higher values for β parameter). Moreover, there are no manifest differences between the analyzed configurations for α parameter in SpamHunting.

From another interesting perspective, we have also tested the performance of the analyzed models in a cost scenario using TCR scores. Fig. 9 shows a graphical comparison of the results obtained when using the values 1, 9 and 999 for λ parameter.

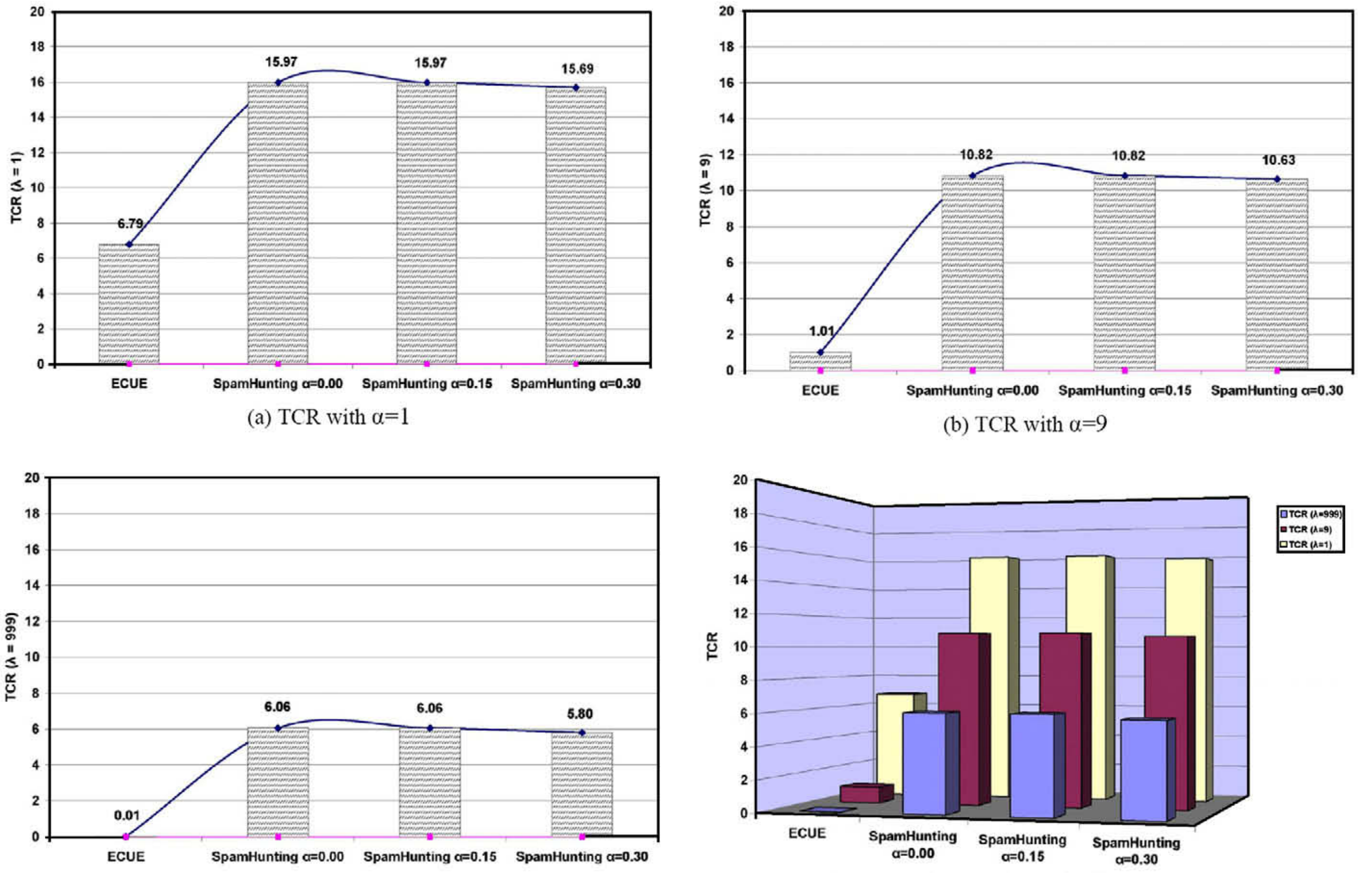
As we can see from Fig. 9, the performance achieved by ECUE system is very poor while the SpamHunting model gets the higher scores. The differences between the distinct configurations for α parameter suggest that the value 0.15 is more appropriate than 0.30.

As in the case of the previous scenario (Section 4.2), we have executed a ROC comparison between the analyzed models to observe their global performance. For calculating the ECUE plot we have selected the measure proposed in Delany et al. (2004), while in SpamHunting we have used the normalized amount of retrieved spam messages. Fig. 10 shows the ROC plot for the three configurations of SpamHunting plus the ECUE model.

As we can see from Fig. 10, there are no significant differences between the proposed configurations of SpamHunting. Nevertheless, both the performance achieved by using the ECUE model and its measure introduced for ROC plots (Delany et al., 2004) seems to be very limited.

Finally, we show the number of existing terms in the EIRN structure of our SpamHunting system when α parameter changes. This analysis reflects an appealing idea about the amount of irrelevant knowledge detected by the proposed technique. Table 5 shows this information.

From Table 5 we can realize that several terms can be eliminated without diminishing the system performance.



(a) TCR with α=1

(b) TCR with α=9

(c) TCR with α=999

(d) comparative 3D view of different α values

Fig. 9. Comparison of analyzed models taking into consideration a cost scenario.

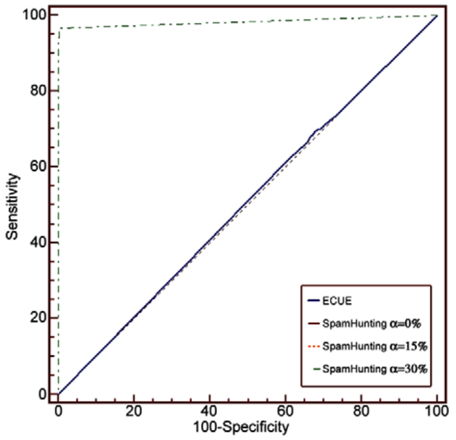


Fig. 10. ROC curves plot for analyzing the behavior of ECUE and SpamHunting models over the SpamAssassin corpus.

Table 5
Garbage term identification study over the SpamAssassin corpus

	SpamHunting		
	$\alpha = 0.00$	$\alpha = 0.15$	$\alpha = 0.30$
Terms	91,703	91,592	91,243
Terms deleted	–	111	460

The amount of deleted terms mainly depends on two factors: (i) the passage of the time (in the analyzed corpus 1 year) and (ii) the amount of messages stored in the system memory.

From the experiments carried out we have shown the superiority of our SpamHunting system using continuous updating strategies, as well as the suitability of the proposed technique for irrelevant knowledge identification and removal. Then final section presents the main conclusions achieved from this work and some future research lines.

5. Conclusions and further work

In this work we have introduced a successful approach for irrelevant knowledge identification in spam filtering. Due to the particularities of this domain, some features used to represent the knowledge managed by models become inadequate for distinguishing between spam and legitimate classes. SpamHunting has previously introduced the possibility of incorporating new features for message representation, to adequately manage up-to-date knowledge. Now, we present a novel technique for identifying and removing those features that become irrelevant due to the passage of the time.

The main weaknesses of classical spam filtering models are the effects that the time passage causes over their initial knowledge representation. The best strategy for overcoming this drawback is the usage of continuous updating strategies. While some successful previous work (such as ECUE) has been exclusively focused on the indexation of new knowledge, SpamHunting has introduced a new approach for continuous updating, working at two different levels: (i) including the indexation of the all available and (ii) continuously searching for the best representation of the existing knowledge.

The proposed technique for irrelevant knowledge detection has shown a great performance level during the experimental stage of this study. Motivated by our novel approach and the underground ideas of our previous SpamHunting model, it is currently the most reliable solution for long-life anti-spam filters. Moreover, the usage of the garbage collection technique introduced in this paper does not limit other well-known techniques for case-base editing such as BBNR and CRR.

Future work will be focused on the development of a three-dimensional plot viewer able to extract more information from the knowledge stored in our EIRN memory structure. We believe that visual representations of stored knowledge are the most reliable way for studying the target problem.

Another interesting research line is related to the use of lexical and semantic information. We believe that the inclusion of lexical anthologies and thesauruses (Seco, Veale, & Hayes, 2004) may be useful for identifying underground relationships between words used for indexing instances. This kind of knowledge can improve the retrieval stage of our SpamHunting system to accurately recover relevant knowledge for identifying spam messages.

References

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 *Workshop on machine learning and textual information access, Fourth European conference on principles and practice of knowledge discovery in databases* (pp. 1–13), Lyon, France.

Androutsopoulos, I., Koustias, 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, 11th European conference on machine learning* (pp. 9–17), Barcelona, Spain.

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

Carreras, X., & Màrquez, L. (2001). Boosting trees for anti-spam e-mail filtering. In *Proceedings of the fourth international conference on recent advances in natural language processing* (pp. 58–64), Tzigov Chark, Bulgaria.

Crocker, D. (1982). Standard for the format of ARPA internet text messages. STD 11, RFC 822. http://www.faqs.org/rfcs/rfc822.html.

Daelemans, W., Jakub, Z., Sloot, K., & Bosh, A. (1997). *TiMBL. Tilburg memory based learning, version 5.1, Reference Guide*. ILK, Computational Linguistics, Tilburg University. http://ilk.uvt.nl/software.html#timbl.

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

Delany, S. J., Cunningham, P. & Coyle, L. (2004). An assessment of case-base reasoning for spam filtering. In *Proceeding of 15th Irish conference on artificial intelligence and cognitive science* (pp. 9–18), GMIT, Castlebar.

Delany, S. J., Cunningham, P., Tsymbal, A., & Coyle, L. (2004). A case-based technique for tracking concept drift in spam filtering. *Knowledge Based Systems, 18*, 187–195.

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

Egan, J. P. (1975). Signal detection theory and ROC analysis. New York: Academic Press.

Fdez-Riverola, F., Iglesias, E. L., Díaz, F., Méndez, J. R., & Corchado, J. M. (2007). SpamHunting: An instance-based reasoning system for spam labeling and filtering. *Decision Support Systems, 43*(3), 722–736.

Fdez-Riverola, F., Iglesias, E. L., Díaz, F., Méndez, J. R., & Corchado, J. M. (2007). Applying lazy learning algorithms to tackle concept drift in spam filtering. *Expert Systems With Applications, 33*(1), 36–48.

Graham-Cumming, J. (2004). *Understanding spam filter accuracy. JGC spam and anti-spam newsletter*. http://www.jgc.org/antispam/11162004-baafcd719ec31936296c1fb3d74d2cbd.pdf.

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* (pp. 1137–1143). Montreal, Canada.

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

McKenna, E., & Smyth, B. (2000a). Competence-guided case-base editing techniques. In *Proceedings of the fifth European workshop on case-based reasoning* (pp. 186–197), Trento, Italy.

McKenna, E., & Smyth, B. (2000b). Competence-guided editing methods for lazy learning. In *Proceedings of the 14th European conference on artificial intelligence* (pp. 60–64), Berlin, Germany.

Méndez, J. R., Fdez-Riverola, F., Díaz, F., Iglesias, E. L., & Corchado, J. M. (2006). Tracking concept drift at feature selection stage in SpamHunting: An anti-spam instance-based reasoning system. In *Proceedings of the eighth European conference on case-based reasoning* (pp. 504–518), Ölüdeniz/Fethiye, Turkey.

Méndez, J. R., González, C., Glez-Peña, D., Fdez-Riverola, F., Díaz, F., & Corchado, J. M. (2007). Assessing classification accuracy in the revision stage of a CBR spam filtering system. In *Proceedings of the seventh international conference on case-based reasoning system* (pp. 374–288), Belfast, Northern Ireland.

Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam filtering with naive bayes – Which naive bayes? In *Proceedings of the third conference on email and anti-spam* (http://www.ceas.cc), Mountain View, California.

Rhiolite software. (2006). *Distributed checksum clearinghouse stats*. http://www.rhyolite.com/anti-spam/dcc/.

Rigoutsos, I., & Huynh, T. (2004). Chung-Kwei: A pattern-discovery-based system for the automatic identification of unsolicited e-mail messages (SPAM). In *Proceedings of the first conference on e-mail and anti-spam* (http://www.ceas.cc), Mountain View, California.

Rijsbergen, K. (1979). Information retrieval. London: Butterworth.

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

Seco, N., Veale, T., & Hayes, J. (2004). An intrinsic information content metric for semantic similarity in WordNet. In *Proceedings of the 16th European conference on artificial intelligence* (pp. 1089–1090), Valencia, Spain.

Shaw, W. M., Burgin, R., & Howell, P. (1997). Performance standards and evaluations in IR test collections: Cluster-based retrieval models. *Information Processing and Management, 33*(1), 1–14.

Smith, B., & McKena, E. (1998). Modelling the competence of case-bases. In *Proceedings of the 4th European workshop on case-based reasoning* (pp. 208–220), Dublin, Ireland.

Smyth, B. (1998). Case-base maintenance. In *Proceedings of the 11th international conference on industrial and engineering applications of AI and expert systems* (pp. 507–516), Benicasim, Spain.

Smyth, B., & Keane, M. T. (1995). Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In *Proceedings of the 14th international joint conference on artificial intelligence* (pp. 377–383), Montreal, Canada.

Smyth, B., & Cunningham, P. (1996). The utility problem analysed: A case-based reasoning perspective. In *Proceedings of the third European workshop on case-based reasoning* (pp. 392–399), Lausanne, Switzerland.

Smyth, B., & McKenna, E. (1999). Building compact competent case-bases. In *Proceedings of the third international conference on case-based reasoning* (pp. 329–342), Seeon Monastery, Germany.

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

Vapnik, V. (1999). *The nature of statistical learning theory* (2nd ed.). *Statistics for Engineering and Information Science*. Berlin: Springer.

Widmer, G., & Kubat, M. (2001). Learning in the presence of concept drift and hidden contexts. *Machine Learning, 23*(1), 69–101.

Yang, Y. (1999). An evaluation of statistical approaches to text categorizations. *Information Retrieval, 1*(1–2), 69–90.