ECUE: A Spam Filter that Uses Machine Learning to Track Concept Drift

Sarah Jane Delany[1] and **Pádraig Cunningham**[2]

Abstract.

While text classification has been identified for some time as a promising application area for Artificial Intelligence, so far few deployed applications have been described. In this paper we present a spam filtering system that uses example-based machine learning techniques to train a classifier from examples of spam and legitimate email. This approach has the advantage that it can personalise to the specifics of the user's filtering preferences. This classifier can also automatically adjust over time to account for the changing nature of spam (and indeed changes in the profile of legitimate email). A significant software engineering challenge in developing this system was to ensure that it could interoperate with existing email systems to allow easy managment of the training data over time. This system has been deployed and evaluated over an extended period and the results of this evaluation are presented here.

1 INTRODUCTION

Spam email has proved to be a problem that is enduring and difficult to solve. In January 2004 Bill Gates predicted that spam email would be eradicated as a problem within two years[3]. The fact that this prediction did not come to pass demonstrates the severity of the problem. Spam is difficult to prevent because of the very open nature of electronic email and because the cost of sending email is close to zero. So even if the rate of return from spam is very small (less than a fraction of a percent) the practice is still worthwhile and there is a constant *arms race* between spammers and email system administrators as each moves to circumvent the initiatives of the other.

Of the wide range of strategies that have been employed to combat spam some of the more effective have been; whitelists and blacklists[4], authentication based techniques[5], collaborative filters[7] and content-based filters. In this paper we describe a system called ECUE (Email Classification Using Examples) that belongs to the category of content-based filters. The objective in the design of ECUE has been to produce a filter that learns from examples and can update itself over time to handle the changing nature of spam. ECUE is a fully engineered system that has been deployed in trials over an extended period with a number of users. ECUE interoperates with, but is independent of, the mail user agent (MUA) of these users and the trials show that it is effective at tracking the changing nature of spam (see section 4).

[1] Dublin Institute of Technology, Kevin St., Dublin 8, Ireland, email:sarah jane.delany@comp.dit.ie

[2] Trinity College Dublin, Dublin 2, Ireland, email: padraig.cunningham @cs.tcd.ie

[3] http://www.theregister.com/2004/01/26/well_kill_spam_in_two/

[4] www.email-policy.com/Spam-black-lists.htm

[5] www.emailauthentication.org/

Since the focus in ECUE has been on handling *concept drift* in email, the two main challenges in the development of the system have been to select an effective case-base from the volume of training data available and to update this case-base over time [18]. The issue of managing the volume of training data requires a case-base editing policy that selects those training examples that are better at prediction that others. This case-base editing technique is called Competence Based Editing (CBE) and has two stages, a noise reduction phase called Blame Based Noise Reduction (BBNR) and a redundancy elimination phase called Conservative Redundancy Reduction (CRR) [17].

The case-base update policy that handles concept drift centers on two hierarchical levels of learning. The first and simplest level is a continuous case-base update with training examples that have been misclassified by our filter. The second level is a periodic retraining of the classifier to reselect features that may be more predictive of spam and legitimate email. We will show how this update policy combined with the initial CBE case-editing procedure can effectively handle the concept drift that is so evident in the spam filtering domain.

In addition to these research challenges, the deployment of ECUE involved the software engineering challenge of integrating the filter with the users' MUA. The overall architecture of the system is described in detail in section 3 and an assessment of the performance of the system in filtering spam is described in section 4. Before that, a review of other work on using machine learning techniques for spam filtering is described in the next section.

2 REVIEW

Research into the use of machine learning techniques for building spam classifiers fall into two categories, those that are evaluated on static datasets in offline environments and those that are evaluated in online, real-time environments. The majority of the research falls into the former category [2, 4, 5, 6, 8, 10, 15, 19] with little published research showing how effective these techniques are at actually filtering real email over time. There are two key machine learning systems that have been evaluated against live email; Filtron [12] and Spamato [1].

Filtron is a prototype anti-spam filter that was designed based on a comprehensive off-line empirical evaluation of four learning algorithms, Naïve Bayes, Flexible Bayes, LogitBoost and Support Vector Machines (SVMs) [3]. It is a Java implementation that runs on Unix platforms only and was evaluated by a single user over seven months. The classifier used was an SVM as that showed the best performance in the offline evaluation, although the system is configurable and different classifiers can be selected. The system was initially trained on 2313 legitimate emails received by the user and 1826 general spam

messages and used 520 features. It was never retrained over the period it was used and the performance was very good with 52 False Positives (legitimate emails incorrectly classified as spam) reported out of 5109 legitimate mails received (approx 1.0%) and 173 out of 1623 spam received missed by the filter (10.6%).

Albrecht et al.'s Spamato filter [1] is an open extendable spam filter framework implemented in Java utilising a plug-in filter architecture. The author's initial beta-test evaluation used a variety of plug-in filters including a number of URL-based filters, a Naïve Bayes classifier, a rule-based filter and a collaborative filter. This evaluation which filtered approximately 90,000 messages from a dozen users, resulted in 0.5% False Positives and 7% spam emails that were missed by the filter.

3 ECUE

A key requirement of our spam filtering system is that it integrates with or works alongside the MUA or mail reader rather than replacing it. This allows the user to continue to use the mail reader software with which they are familiar. To this end, the system architecture has been designed to support initially the Internet Message Access Protocol (IMAP) protocol [9]. One advantage of IMAP over POP3 (the other mail protocol) is that IMAP supports the storing of messages on the central server for access from multiple sites. By using IMAP to access the mailbox, messages can be filtered and flagged on the server and this allows the user to use any client mail reader application that supports IMAP to access and read their email. All the popular mail reader applications including MS Outlook, Mozilla, Netscape and Thunderbird support IMAP.

The user and the spam filtering system have to be able to interact for two reasons. Firstly the filter has to let the user know of emails categorised as spam and secondly the user has to be able to alert the filter to emails have been classified incorrectly. Since a requirement of the system is to integrate with existing mail readers rather than replace them, it is important to define a way of interacting with the user that is consistent across mail readers.

ECUE uses the IMAP mail folders as the means of system-user interaction. The filter places any emails it categorises as spam into a user-defined spam folder. It leaves any email that it classifies as non-spam in the Inbox. If the user finds any mails classified incorrectly they indicate this to the system by moving the emails from the folders they were in to or from the spam folder. So a False Positive (FP) email should be moved from the spam folder (where the filter had placed it) into any other folder, indicating that it should not be in the spam folder. Similarly a False Negative (FN) email, a spam email missed by the filter, should be moved to the spam folder to indicate that it should have been classified as spam. With this model for system-user interaction, existing mail clients do not have to be extended to enable them to be used with the spam filtering system. All interaction is at the mail server level. This is also a familiar spam interaction model for users. Figure 1 depicts the state transition diagram for an email message which shows all the possible states for an email message.

The main drawback of this process of user interaction is the requirement to monitor the spam folder for false positives. To address this ECUE produces a measure of classification confidence that can be used to partition messages classified as spam into definitely-spam and maybe-spam categories. The definitely-spam category need not be monitored for FPs [16].

In order to track the email message as it arrives and is filtered, an ECUE application specific header field is added to the email message. The value of this header field, which represents the state of the message as described in Figure 1, in conjunction with the folder the message is in indicates if and how the email has been moved by the user.

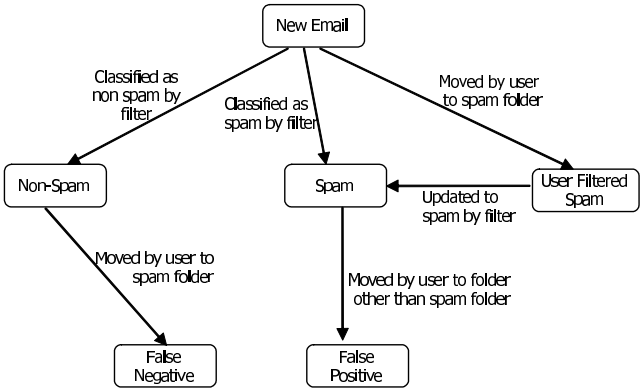

Figure 1. State Transition Diagram for an email message

Mail folders are also used to identify the initial training data for the ECUE system. The user identifies the training emails which are to be used for the initial case-base set up by placing examples of their spam and legitimate emails into separate 'training' mail folders. These folders are identified in ECUE's configuration file and all emails in these folders are taken to be the initial training data.

3.1 The ECUE Learning System

The architecture of the learning system is described in Figure 2. The system uses previous examples of both spam and legitimate email received by the user as training data. In the initial training phase, the first process that the emails undergo is Feature Extraction which involves parsing or tokenising the text content of the training emails into features. No stop-word removal or stemming is performed on the text. Email attachments are removed before parsing but any HTML text present in the email is included in the tokenisation. As ECUE is a personalised filter, the header fields may contain useful information and a selection of header fields, including the *Subject*, *To* and *From* headers are included in the tokenisation. This idea is supported by a number of researchers [5, 14, 19] who concluded that the information from the message header is as important as the message body.

Three types of features are extracted; word features, letter features and statistical or structural features. The feature extraction process results in a large number of features for each training email. In addition, the representation of each email will be sparse, with only a small number of the total feature set having a value greater than zero. The Feature Extraction process for a typical training corpus will produce some tens of thousands of features. The task of Feature Selection is to identify which of these features are most predictive of spam or legitimate mails. The technique used for feature selection is Information Gain [13]. The output of feature selection is a reduced feature set where each training example email has a reduced set of feature-value pairs, including only those features identified as the most predictive. The number of features used by ECUE is configurable.

The task of Case Selection is to apply the Competence-Based Editing technique [17] which uses the competence properties of the examples in the case-base to remove noisy and redundant cases from

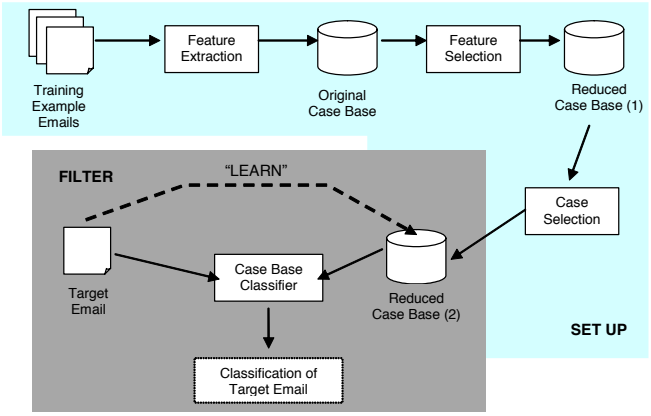

Figure 2. ECUE application structure

the case-base. In effect Case Selection reduces the size of the case-base while ensuring that the generalisation accuracy of the system is not adversely affected.

In an example-based learner such as ECUE, the training examples are represented as cases in a case-base. Each training example is a case e_i represented as a vector of feature values, $e_i = (f_{1j}, f_{2j}, \ldots f_{nj}, s)$. The classification of new (or target) emails is performed using the k-Nearest Neighbour algorithm. As the case representation is binary, i.e. if the feature exists in the email the feature value $f_{ij} = 1$ otherwise $f_{ij} = 0$, a Case Retrieval Net [11] was implemented to speed up the retrieval process. The value of k is configurable and set up in the configuration file. A value of $k = 3$ was used for all evaluations presented in this paper. Due to the significance of FPs the classification process uses unanimous voting to bias the classifier away from such FP classifications. ECUE's unanimous voting requires all k neighbours retrieved by the Nearest Neighbour algorithm to be of class *spam* before the target case can be classified as spam.

ECUE uses different training data depending on the case-base that has to be built. If it is the first time the application is run after installation, ECUE uses training data that is placed by the user into two training folders in their mailbox. If a case-base is required to be built at any other time, e.g. when a feature reselection occurs, ECUE uses the most recent emails received by the user as training data. In these circumstances a percentage of the training data is made up of a selection of the most recently misclassified emails. The total number of emails to be used as training data is configurable as is the proportion of the training data that should comprise recently misclassified emails. This percentage is made up of all the FP emails previously identified (this number will be small) and an appropriate number of the previously identified FNs, randomly selected. An appropriate number of most recently correctly classified spam and non spam are then added to bring the training set up to the specified size.

When the user identifies emails that have been incorrectly classified by the system learning should take place. There are two levels of learning built into the system:

(i) incorrectly classified emails with their appropriate classification are regularly added to the current case-base;
(ii) a feature re-selection process and a case-base rebuild is performed on more recently received emails.

In order to help reduce and eliminate false positives, the system includes simple whitelisting which operates at two levels: Firstly, the user can define, in the configuration file, domains that will be acceptable to the filter. Any email that comes from these domains will be considered as legitimate. Secondly, the sender of all legitimate emails are maintained on a list and for all emails a case feature is set that indicates whether the sender is on the whitelist or not. This feature is used in the case-base retrieval process when identifying the most similar neighbours of the new email.

4 EVALUATION

ECUE was evaluated in a real-time online setting. The aim of the evaluation was to install the example-based spam filter in a 'live' environment and evaluate its performance. Since ECUE was designed to handle the concept drift in email the specific objective of the evaluation was to assess the *online* learning capabilities of the system. Over the evaluation period records were maintained of how ECUE performed both with and without the learning facilities.

ECUE was installed on the PCs of a number of users within the Computer Science department in two third level institutions in ZZZ. The users were lecturers, postgraduate students and researchers within the departments. Since both institutions run a gateway spam filter (SpamAssassin) some users was asked to turn off SpamAssassin and to use ECUE for filtering their email. Others used ECUE as a second-level spam defense, filtering email that had passed through the gateway filter first.

When users identified emails that were misclassified by ECUE, they were asked to move the emails to the appropriate mail folders. Users were asked to initiate a feature reselection process when they felt that the performance of the filter was deteriorating or at least every couple of weeks.

The evaluation metrics used include:

(i) The error rate; the overall proportion of emails that ECUE did not filter correctly (labeled *%Error* in all diagrams)
(ii) The FN rate; the proportion of spam emails that ECUE missed (labeled *%FNs*).
(iii) The FP rate; the proportion of legitimate emails that ECUE classified as spam (labeled *%FPs*).

For all these measures, figures are included for how ECUE performed when it attempted to handle the concept drift (i.e. with learning capabilites), labelled *with update*, and when the ECUE simply used the initial training data in filtering and did not attempt to handle the concept drift, labelled *no update*.

4.1 Evaluation Results

The evaluation was split into two phases. A preliminary evaluation involved an initial version of ECUE which included the case-base update facility, the first level of learning. The main online evaluation included both levels of learning, the regular update capability and the periodic feature reselection capability.

4.1.1 Preliminary Evaluation

Table 1 presents the results of the preliminary evaluation for six users. In addition to the performance figures, the table lists for each user the number of days that ECUE filtered the user's email and the number of spam and legitimate emails that were filtered during that time period.

Table 1. Preliminary evaluation results

User		1	2	3	4	5	6
#days		28	34	30	50	70	41
Emails Filtered	#spam	969	618	80	890	3053	3248
	#legit	191	422	390	2110	747	1352
% Error	No update	8.1	9.6	7.5	20.7	22.0	15.5
	With update	5.6	6.5	4.5	8.0	10.7	13.6
% FPs	No update	1.6	2.4	5.0	0.6	1.5	8.4
	With update	1.0	2.1	1.8	0.7	2.7	2.4
% FNs	No update	9.4	14.5	20.0	68.3	27.0	18.5
	With update	6.5	9.5	17.5	25.4	12.6	18.2

Analysis of these results show that ECUE performed better in all cases when update was enabled. Figure 3 shows a graph of the performance of ECUE for User 4. The graph shows the accumulated error (y-axis) over a certain number of emails (x-axis).

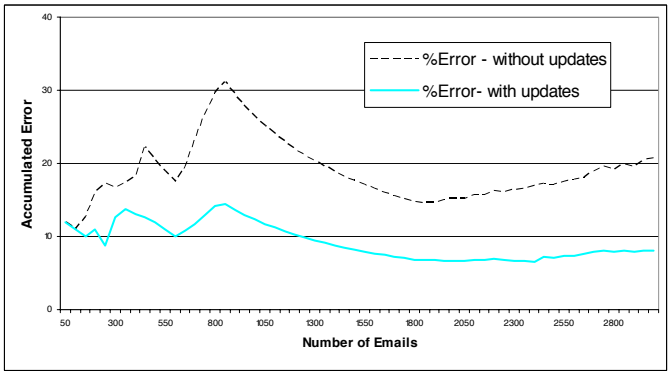

Figure 3. Performance of ECUE for User 4 in the preliminary evaluation.

However, improvements in the FP rate are not as consistent as the improvements in the FN rate. Four of the six users show improvements in the FP rate. Of the two remaining, one user (user 5) shows a considerable increase in FP rate from 1.5% without updating to 2.7% with updates. This may be explained by the fact that this user received very high levels of spam and the evaluation ran for a long period (70 days). As ECUE was very successful in handling the concept drift in the spam emails (the FN rate dropping from 27% to 12.6%) the system is being updated with a large number of spam emails to cope with this change and as a result becomes biased toward predicting spam.

The preliminary evaluation also showed that ECUE did not perform as well for users who receive high numbers of spam emails (users 5 and 6). These users had less than 90% of their email classified correctly whereas all other users had 92% or higher classified correctly. This indicated that it is necessary to include a further level of learning to allow a feature reselection to be performed to better handle the concept drift in the spam emails.

4.1.2 Full Evaluation

Table 2 displays the results of the full evaluation of ECUE. For each user the table lists the start and end date that ECUE ran on the user's PC and the number of spam and legitimate emails that were filtered during that time period. The table also lists information about the training data used - the number of emails used (*initial size*) and what proportion of the training data was spam email (*%spam*). The number of times the user initiated the feature reselect process during the evaluation period (*#ftr reselects*) is included along with the performance achieved both with and without update, when update included a feature reselection process.

Table 2. ECUE full evaluation results

User		1	2	3	4
Filter Period	Start date	18-11-04	9-3-05	20-4-04	7-9-05
	End date	15-07-05	15-07-05	16-10-05	1-11-05
Emails Filtered	#spam	3689	4081	742	75
	#legit	1161	469	1480	917
Casebase	Initial size	308	299	201	308
	%spam	56%	54%	79%	60%
	#ftr reselects	3	3	1	2
% Error	No update	32.8	21.8	17.3	8.1
	With update	6.1	4.7	12.1	4.3
% FPs	No update	0.3	0.0	1.3	7.3
	With update	0.7	0.2	1.1	0.4
% FNs	No update	43.2	24.4	49.3	17.3
	With update	7.8	5.2	34.0	52

Analysis of these results also show that ECUE performed better with update in all cases. Figure 4 shows a graph of the performance of ECUE for user 1.

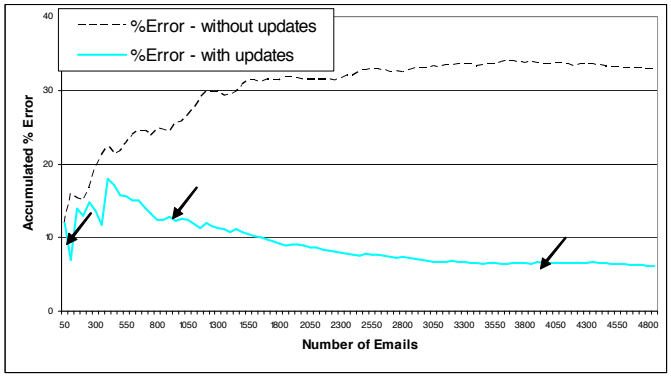

Figure 4. Performance of ECUE for User 1 in the full evaluation. The arrows show when the feature re-selection process occurred.

ECUE learned to recognise new examples of spam over the time period as is evident from the reduction in the FN rate for all users except user 4. The FN rate of user 3 did not drop as significantly as that of the others. This may be accounted for by the fact that all email received by user 3 was subject to initial organisation-level spam filtering on the mail server before it was forwarded to user 3's personal mailbox. As such, the spam email received by user 3 was spam that had been missed by the organisation-level filter and is possibly more difficult to recognise as spam. ECUE still identified 66% of user 3's spam correctly.

The bias of the classifier appears to be influenced to a large extent by the proportions of spam and legitimate email in the training data. User 4 received more than 12 times the amount of legitimate email as spam (possibly due to the success of the organisation level filter to which user 4's email was subject). User 4 has a significant increase in

FN rate, possibly due to the classifier becoming biased towards predicting non spam over the period of the evaluation. Similarly, users 1 and 2 receive significantly more spam email that legitimate email and display a slight increase in the FP rate. For these users the system is being updated constantly to try to cope with the spam concept changes and as a result loses some accuracy in the prediction of legitimate emails. Increasing the value of k in the k-NN classifier may be a way of controlling this bias.

A shortcoming evident from the preliminary evaluation of ECUE was that it did not perform as well for users who receive high numbers of spam emails. These users had less than 90% of their email classified correctly. Users 1 and 2 in the full evaluation have this profile with the proportion of spam received varying from 76% to 89% respectively. Both of these users had over 93% of their mail classified correctly. (93.9% for user 1 and 95.3% for user 2). In addition the average error across both evaluations dropped from 8.15% in the preliminary evaluation to 6.8% in the full evaluation indicating that the additional level of learning, the periodic feature reselection process, allows the system to better handle the concept drift in the spam emails.

5 COMMERCIAL PROSPECTS

Our preliminary commercial analysis suggests that the commercial prospects for message classification systems that can learn are considerable. While this area of spam filtering is a rather crowded market there is a range of other areas where large volumes of messages need to be filtered or routed. There is considerable potential for message routing systems for incoming email into volume accounts such as *info@company.com*. Compliance legislation such as the Sarbanes-Oxley Act in the US[6] generates a requirement to be able to monitor outgoing message streams to ensure compliance. There is also a considerable market for systems for routing XML messages in the financial sector. An attractive aspect of many of these application areas is that a low level of error can be tolerated. If a message arrives at the wrong destination it can simply be re-routed manually.

6 CONCLUSIONS

In this paper we describe ECUE, an example-based spam filtering application that learns from new examples of spam and legitimate email. We believe that this is a landmark deployment of an AI application that incorporates online-learning in an unobtrusive way. As a lazy local learner, ECUE offers distinct advantages over alternative eager approaches to spam filtering such as Naïve Bayes or Support Vector Machines, approaches that are more common in commercial filters. It provides capabilities to learn seamlessly without the need for a separate learning process. Also, the fact that spam is a diverse concept makes a local learner, an appropriate choice.

The evaluations show that ECUE is successful at filtering mail at a personal level, identifying between 92% and 94% of spam correctly with less than 1.0% false positives identified. For users who used ECUE as a second-level spam defense, operating the filter on email that had already been passed through a organisation-level gateway spam filter, ECUE still successfully filters between 48% and 66% of the spam correctly albeit with a slightly higher false positive rate for one of these users of just over 1%.

To conclude, there is no single approach that will be 100% effective at handling spam. The solution to spam is currently a multi-layered approach, utilising legislative measures, authentication techniques and filtering. Filtering plays and will continue to play a significant role in this fight against spam. We believe that ECUE, our example-based approach to filtering can be a important contributor to content-based spam filtering. We have shown how it can handle the changes in spam emails with relative ease without putting too much burden on the individual using it.

[6] http://www.sarbanes-oxley.com/

REFERENCES

[1] K Albrecht, N Burri, and R Wattenhofer, 'Spamato - an extendable spam filter system', in *Proceedings of the 2nd Conference on Email and Anti-Spam (CEAS'05)*, (2005).

[2] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, and P. Stamatopoulos, 'Learning to filter spam email: A comparison of a naive bayesian and a memory based approach', in *Proc of Workshop on Machine Learning and Textual Information Access, PKDD 2000*, eds., H. Zaragoza, P. Gallinari, and M. Rajman, pp. 1–13, (2000).

[3] I. Androutsopoulos, G. Paliouras, and E. Michelakis, 'Learning to filter unsolicited commercial email', *Technical Report 2004/02, NCSR "Demokritos"*, (2000).

[4] Z. Chuan, L. Xianliang, H. Mengshu, and Z. Xu, 'An lvq-based neural network anti-spam email approach', *SIGOPS Operating Systems Review*, **39**(1), 34–39, (2005).

[5] H. Drucker, D. Wu, and V. Vapnik, 'Support vector machines for spam categorisation', *IEEE Transactions on Neural Networks*, **10**(5), 1048–1055, (1999).

[6] K. R. Gee, 'Using latent semantic indexing to filter spam', in *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, pp. 460–464. ACM Press, (2003).

[7] A. Gray and M. Haahr, 'Personalised, collaborative spam filtering', in *Proceedings of 1st Conference on Email and Anti-Spam*, (2004).

[8] Jose Maria Gomez Hidalgo, M. Mana López, and E. Sanz, 'Combining text and heuristics for case-sensitive spam filtering', in *Proceedings of the 4th Computational Natural Language Learning Workshop (CONLL-2000)*, (2000).

[9] L. Hughes, *Internet e-mail: protocols, standards and implementation*, Artech House Inc., 1998.

[10] A. Kolcz and J. Alspector, 'Svm-based filtering of email spam with content-specific misclassification costs', in *TextDM'2001 (IEEE ICDM-2001 Workshop on Text Mining)*, pp. 123–130. IEEE, (2001).

[11] M. Lenz, E. Auriol, and M. Manago, 'Diagnosis and decision support', in *Case-Based Reasoning Technology, From Foundations to Applications*, eds., M. Lenz, B. Bartsch-Spörl, HD. Burkhard, and S. Wess, LNCS, pp. 51–90. Springer-Verlag, (1998).

[12] E. Michelakis, I. Androutsopoulos, G. Paliouras, G. Sakkis, and P. Stamatopoulos, 'Filtron: A learning-based anti-spam filter', in *1st Conference on Email and Anti-Spam (CEAS 2004)*, (2004).

[13] J. R. Quinlan, *C4.5 Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Mateo, CA., 1997.

[14] J. Rennie, 'ifile: an application of machine learning to e-mail filtering', in *Proc of the KDD-2000 Workshop on Text Mining 6th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining*, (2000).

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

[16] WWW, XXX, and YYY, 'Generating estimates of classification confidence for a case-based spam filter', in *Proceedings of the 6th International Conference on Case-Based Reasoning (ICCBR 2005)*, volume 3620 of *LNAI*, pp. 170–190. Springer, (2005).

[17] WWW, XXX, YYY, and ZZZ, 'An analysis of case-based editing in a spam filtering system', in *7th European Conference on Case-Based Reasoning (ECCBR 2004)*, eds., P. Funk and P. González-Calero, volume 3155 of *LNAI*, pp. 128–141. Springer, (2004).

[18] WWW, XXX, YYY, and ZZZ, 'A case-based technique for tracking concept drift in spam filtering', *Knowledge-Based Systems*, **18**(4–5), 187–195, (2005).

[19] L Zhang, J Zhu, and T Yao, 'An evaluation of statistical spam filtering techniques', *ACM Transactions on Asian Language Information Processing (TALIP)*, **3**(4), 243–269, (2004).