A Case-Based Approach to Spam Filtering that Can Track Concept Drift

Pádraig Cunningham[1], Niamh Nowlan[1], Sarah Jane Delany[2], Mads Haahr[1]

[1]Department of Computer Science, Trinity College Dublin
[2]School of Computing, Dublin Institute of Technology
Padraig.Cunningham@cs.tcd.ie

Abstract. There are a few key benefits of a case-based approach to spam filtering. First, the many different sub-types of spam suggest that a local learner, such as Case-Based Reasoning (CBR) will perform well. Second, the lazy approach to learning in CBR allows for easy updating as new types of spam arrive. Third, the case-based approach to spam filtering allows for the sharing of cases and thus a sharing of the effort of labeling email as spam. In this paper we introduce a case-based approach to spam filtering and present preliminary evidence of the first two of these advantages.

1. Introduction

Work on spam filtering can be divided into two categories. There is the content-based approach where the classification of an email as spam is based on an analysis of the content of the email. This 'content-based' category can be extended to include approaches that use features extracted from the header of the email to help the classification. The alternative is the collaborative approach that does not consider the content of the email but depends on the collaboration of groups of users who share information about spam. When a new spam message appears, an early receiver of the spam shares a signature for that spam (typically one or more hash codes) with the rest of the group. If the other users also receive this message their filters can identify it as spam based on the shared signature.

Spam filtering is a difficult classification task for a variety of reasons. Spam is constantly changing as spam on new topics emerges. Also, spammers attempt to make their messages as indistinguishable from legitimate email as possible and change the patterns of spam to foil the filters. Another serious issue is the problem of false positives, i.e. a legitimate email classified as spam. For many email users, false positives are simply unacceptable; thus the requirements on the spam filter are very exacting.

As new types of spam emerge and spammers change their behaviour to avoid detection, a content-based system will require updating. Features that are predictive of the new types of spam and rules to handle these features are required. There is a fair deal of interest in using Machine Learning (ML) techniques to automate this process. Because of its proven ability in text classification (Lewis & Ringuette, 1994), the

Naïve Bayes approach is the most popular ML technique in research on spam filtering (e.g. Sahami et al, 1999).

In this paper we present a preliminary evaluation where a Case-Based approach to spam filtering outperforms Naïve Bayes. This is because spam is a disjoint concept; spam about software for pirating DVDs has little in common with spam selling pornography. Case-Based classification works well for disjoint concepts whereas Naïve Bayes tries to learn a unified concept description.

The other advantage the Case-Based approach has is the ease with which it can be updated to catch the *concept drift* (Widmer & Kubat, 1996) in spam. The management of the case-base over time in order to maximize accuracy in the face of concept drift is an interesting challenge. Thus a Case-Based spam filtering system is a long-lived CBR system where the policy on case-selection and case-deletion as new positive and negative examples become available is a major factor in determining the success of the system. In the preliminary evaluation presented here we show that a policy of training on the most recent examples will not always produce the best results. A key challenge in developing a comprehensive Case-Based system for spam filtering is the development of a policy for case-base maintenance.

The final advantage of the case-based approach to spam filtering is that it offers a natural framework to unify learning and collaboration approaches. If the collaborative system exchanges case representations rather than hash-based signatures it can be viewed as a distributed CBR system (Leake & Sooriamurthi, 2002).

The body of this paper begins with a brief introduction to Internet mail and spam in section 2 and then a review of spam filtering in section 3. The case-based system for spam filtering is presented in section 4 where it is compared with the Naïve Bayes alternative and its potential to track concept drift is assessed. The paper finishes with some conclusions and an outline of future work in section 5.

2. What is Spam?

Spam is unsolicited and unwanted email from a stranger that is sent in bulk to large mailing lists, usually with some commercial objective.[1] Some would argue that this definition should be restricted to situations where the receiver is not especially selected to receive the email – this would exclude emails looking for employment or positions as research students for instance. This difficulty in definition demonstrates that the definition depends on the receiver and strengthens the case for personalised spam filtering.

Spam is junk email; junk postal mail and junk faxes are also a problem. However, because of the special nature of the Internet, there are two reasons why junk email is a particular problem. First, spam can be sent with almost no cost to the sender. In fact the costs associated with the spam are paid by people other than the sender (the Internet Service Provider (ISP) and the receiver). Second, it is difficult to imagine

[1] The name "Spam" comes from a Monty Python sketch where a group of Vikings wish to eat in a restaurant where the menu contains so much Spam (the food) that it is difficult to determine what else is available (see www.detritus.org/spam/skit.html)

legal measures that can be taken within a jurisdiction that can prevent the *receipt* of spam within that jurisdiction.

For these reasons junk email is much more of a problem that other junk advertising. Because of this, filtering mechanisms have been developed to detect Spam. In fact something of an 'arms race' has developed between the spammers and the developers of these filtering systems. Examples of this escalation in 'sophistication' of spam are; forged header details to beat blacklisting, disguised words, e.g. Adu1t ('l' replaced with a one) and random text to beat signatures based on text hashing.

3. Spam Filtering

Spam filtering in Internet email can operate at two levels, an individual user level or an enterprise level (see Figure 1). An individual user is typically a person working at home and sending and receiving email via an ISP. Such a user who wishes to identify and filter spam email installs a spam filtering system on her individual PC. This system will either interface directly with their existing mail user agent (MUA) (more generally known as the mail reader) or more typically will act as a MUA itself with full functionality for composing and receiving email and for managing mailboxes.

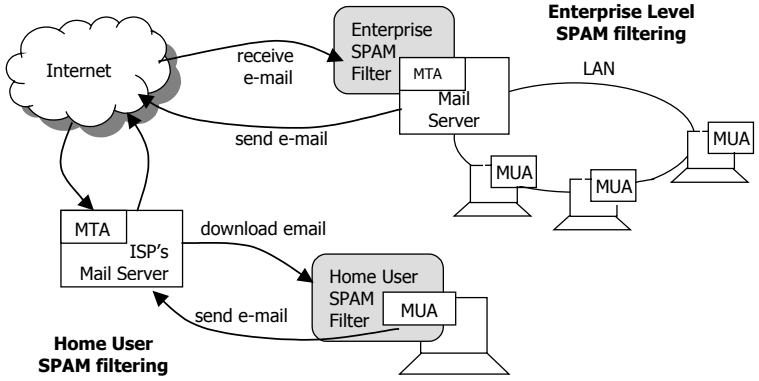

Figure 1. Alternatives for spam filtering in Internet e-mail**.**

Enterprise-level spam filtering filters mail as it enters the internal network of an enterprise. The software is installed on the mail server and interacts with the mail transfer agent (MTA) classifying messages as they are received. Spam email, which is identified by the enterprise spam filter, will be categorised as a spam message for all users on that network. Spam can be filtered at an individual level on a LAN also. A networked user can choose to filter spam locally as it is downloaded to their PC on the LAN by installing an appropriate system.

The vast majority of current spam filtering systems use rule-based scoring techniques. A set of rules is applied to a message and a score accumulates based on the rules that are true for the message. Systems typically include hundreds of rules

and these rules need to be updated regularly as spammers alter content and behaviour to avoid the filters. Systems also incorporate list-based techniques where messages from identified users or domains can be automatically blocked or allowed through the filter.

If the score for an email exceeds a threshold, the email is classified as spam. Limited learning capabilities are beginning to appear in systems such as Mozilla and the MacOS X Mail program but these systems are still in their infancy. Naïve Bayes seems to be the technique of choice for adding a learning capability to commercial spam filtering systems.

3.2. Collaborative Spam Filtering

In addition to this content-based approach to spam filtering there is also some work on a collaborative approach. The collaborative approach does not consider the content of the email but depends on the collaboration of groups of users who share information about spam. When a new spam message appears, an early receiver of the spam shares a signature for that spam (typically one or more hash codes) with the rest of the group. If the other users also receive this message their filters can identify it as spam based on the shared signature. In this approach there are two key issues; an effective signature mechanism needs to be devised and a process for sharing these signatures needs to be developed. Spammers insert random characters into messages to foil hash-based signatures so flexible and clever signatures are needed. The sharing of these signatures can be centralised through a clearing-house or it can be truly distributed using peer-to-peer techniques.

The dominant system in this area is Vipul's Razor (razor.sourceforge.net), also available as SpamNet (www.cloudmark.com). Vipul's Razor uses a centralised clearing-house for sharing signatures and much of the research has focused on developing sophisticated signatures. They have developed fuzzy signatures where signatures for similar texts are also similar. They also use ephemeral signatures where the hashing process changes over time. The accuracy of Vipul's Razor is somewhat disappointing at 60%-90%; this is due in part to the fact that it is a generic rather than a personalised system. The centralised clearing-house approach can be seen as a shotgun approach to filtering: it requires each user to filter each incoming mail against all signatures stored by the clearing-house. Apart from the obvious scalability issues related to the distribution of a continually growing body of signatures to a potentially large number of users, this approach is also problematic because it relies on the implicit assumption that any one user is being targeted by all spammers. In reality, this is unlikely, and the centralised clearing-house approach is therefore likely to result either in sub-optimal accuracy (cf. Vipul's Razor) or increased risk of false positives.

4. Case-Based Spam Filtering

Figure 2 shows the structure of a Case-Based spam filtering system. (So far, the idea has only been tested off-line on a corpus of spam). The Mail User Agent (MUA) is

extended to allow the user to label messages as spam and non-spam. There is a Case Retention Process that maintains a Personalised Case-Base of spam. This involves selecting the appropriate features to represent spam and non-spam messages and selecting the cases that give the best coverage. Finally there is the spam classifier that intercepts the download of email and tags the spam. Because of the problem of concept drift with spam the cycle in Figure 1 of which Case Retention is a long-lived process. Cases need to be added to the Case-Base to cover new types of spam and cases need to be deleted as older types of spam disappear. There will also be drift in the characteristics of legitimate email. In section 4.2 we present a preliminary analysis of how this might be done.

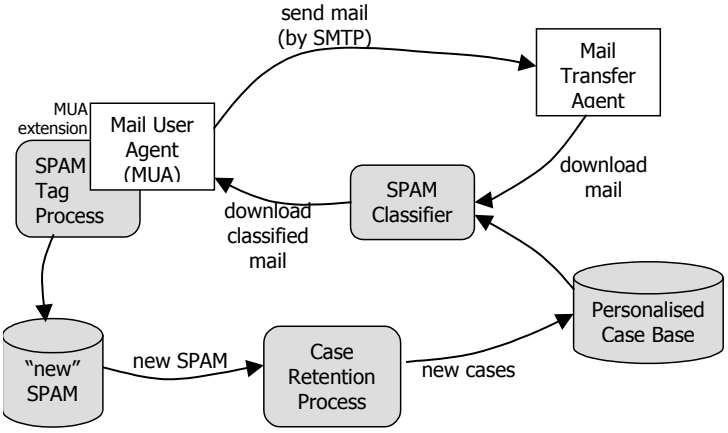

Figure 2. The components of a Case-Based spam filtering system.

It was mentioned in the introduction that Naïve Bayes is a popular classification mechanism for spam filtering. This is probably due to its ability to deal with high dimension data. A spam classifier can use words in the text augmented with features extracted from the header (e.g. forged "From" field) and aggregate features from the text (e.g. One full line of Capitals). Because of this, dimension reduction is a big issue if ML techniques such as decision trees, neural networks or nearest neighbour are to be used.

In this paper we are proposing a nearest neighbour approach to spam filtering that begins by reducing the dimension of the data. The process we use for extracting features and reducing the feature set to a manageable size is as follows:

Step 1. Gather training sets of spam and non-spam mail messages.

Step 2. Remove stop-words and stem the remaining words.

Step 3. Generate vocabularies of words occurring in the spam and non-spam emails. Count the frequencies of each word in the training data and update the vocabularies with this count.

Step 4. Sort the vocabularies based on the odds-ratio described below.

Step 5. Select the top l terms (typically 40) from each vocabulary to be part of the feature representation.

Step 6. Augment this feature list with some features (currently 6) extracted from the mail header.

The odds-ratio is calculated as follows:

$$OR(w, c_i) = \frac{P(w \mid c_i) \times [1 - P(w \mid \overline{c}_i)]}{[1 - P(w \mid c_i)] \times P(w \mid \overline{c}_i)} \quad \textbf{(1)}$$

where $P(w \mid c_i)$ is the probability of word w occurring in category (i.e. spam or non-spam) c_i. When a word does not occur in a category the associated probability is assigned a small fixed value so that the odds-ratio can still be calculated. Words that are equally likely to occur in spam and non-spam have an odds-ratio close to 1 while words that are indicative of a category have a value greater than one. Thus the feature selection process described in Steps 4&5 above selects the top l features predictive of spam and the top m features predictive of non-spam.

So each mail message is represented by a feature vector of length $l+m+n$ in the classification process (n is the number of features extracted from the header).

4.1. Comparison with Naïve Bayes

Our preliminary research has shown that a Case-Based approach to spam filtering can outperform the Naïve Bayes approach (see Figure 3). In this evaluation, cases are represented by 30 spam words, 30 non-spam words and 7 header features. The classifiers are trained on 200 spam and 200 non-spam cases and evaluated on (roughly) 150 spam and 150 non-spam at each test point. The Case-Based approach outperforms Naïve Bayes probably because of the different sub-types of spam. Naïve Bayes is trying to learn a unified spam *concept* whereas the Case-Based approach will classify an email as spam if it looks like *any* spam in the training data.

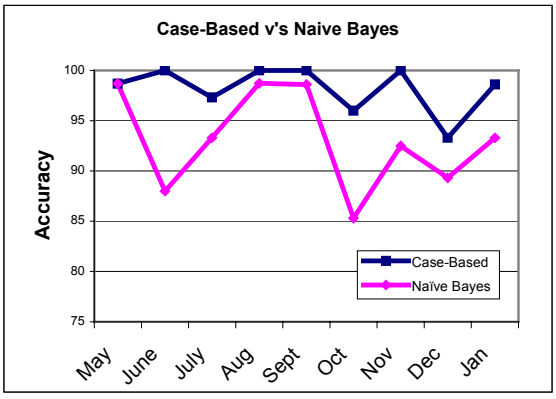

Figure 3. A comparison of the classification accuracy of Case-Based and Bayesian spam filtering.

While this approach to spam filtering shows great promise, there are still a few problems to be addressed. Accuracy could be increased by addressing the more rare types of spam. Feature selection based purely on the odds-ratio focuses too much on commonly occurring spam and can select features that are correlated and redundant. Later in this research, more comprehensive Wrapper-based feature subset selection techniques will be evaluated (Cunningham & Carney, 2002).

4.2. Addressing Concept Drift

In addition to its advantages of a local learner in spam filtering, the case-based approach also has advantages for addressing concept drift because it is a *lazy* learner. In an ongoing mail filtering situation there are two modes in which the case-based spam filter can be retrained:

Full Retraining: The performance of the filter can be monitored by the user and once the false negative (or false positive) rate goes above a certain threshold the system is retrained. This involves selecting a case-base of more recent training examples and redoing the feature selection process.

Continuous Retraining: As false positives and false negatives are identified by the user they are added to the case base with older less useful cases being deleted. The feature selection process is not redone.

So far we have only evaluated the first of these scenarios. Figure 4 shows how the Case-Based spam filtering mechanism can be retrained periodically on more recent training data (i.e. examples of spam and non-spam) to increase accuracy. In order to accentuate the effect of retraining the classifier has been *nobbled* by reducing the number of non-spam features to 10. This reduces the overall accuracy of the system significantly. In this example a full retraining of the system was performed; i.e. the feature selection and case selection stages were repeated. The case-selection process in operation is the most basic; the most recent cases are used for training.

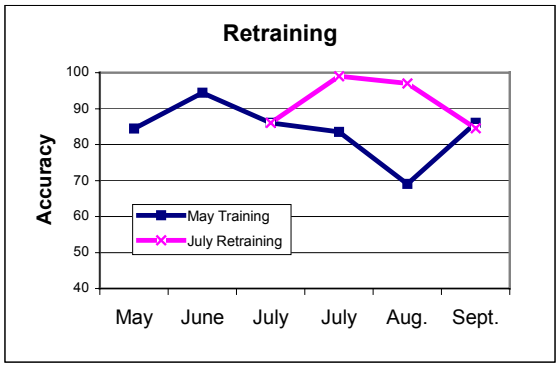

Figure 4. Retraining of the Case-Based spam filtering system improves accuracy.

Sometimes the accuracy of the retrained classifier (as shown in Figure 4) can fall below the earlier version. This is due to a sub optimal policy on case-selection. In the

current system the most recent training data is used and older data is discarded. In later work we will look at case-selection techniques such as those proposed by McKenna & Smyth (2000) to produce a better policy for managing the training data.

4.3. Collaborative Case-Based Spam Filtering

While the case-based approach to spam filtering shows considerable promise, it has the drawback that it places the burden of labeling the training data on the user. The collaborative approach has the advantage that the labeling task is shared across the community of users. We propose to develop an integrated system that will combine the advantages of case-based and collaborative spam filtering. The case-based framework is a natural structure for this. The users can share cases in the manner that signatures are shared in a collaborative system. This will draw on recent research on on distributed case-based reasoning, see (McGinty & Smyth, 2001; Plaza & Ontanon, 2001; Leake & Sooriamurthi, 2002) for instance.

5. Conclusions

In this paper, we report some preliminary work on a case-based approach to spam filtering. Our initial evaluation suggests that a case-based classifier will outperform the popular Naïve Bayes approach and we suggest that this is because the 'local learner' characteristics of case-based classification is particularly suited to spam filtering. We believe that the performance of the case-based classifier could be further improved by more careful feature subset selection.

The case-based approach should also be useful for handling concept drift. Two policies for case-base update have been outlined and one of these has been evaluated and is seen to work. Much work remains to be done on case-base maintenance (i.e. case selection) policies in order to maximize performance.

We also outline how the case-based approach to spam filtering can unify the content-based and collaborative approaches. A distributed case-based approach would have the benefits of the case-based approach and would reduce the burden of labeling on the users of the system.

References

Androutsopoulos, I., Koutsias, J., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., & Stamatopoulos, P., (2000) Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. *in Workshop on Machine Learning and Textual Information Access, at 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD).*

Androutsopoulos, I., Koutsias, J., Konstantinos, V., Chandrinos, V., Paliouras, G., Spyropoulos, C., (2000) An evaluation of Naive Bayesian anti-spam filtering, *in*

Proceedings of the workshop on Machine Learning in the New Information Age, G. Potamias, V. Moustakis and M. van Someren (eds.), 11th European Conference on Machine Learning, Barcelona, Spain, pp. 9-17, 2000

Cunningham, P., Carney, J., (2000) Diversity versus Quality in Classification Ensembles based on Feature Selection, *11th European Conference on Machine Learning (ECML 2000)*, Lecture Notes in Artificial Intelligence, R. López de Mántaras and E. Plaza, (eds) pp109-116, Springer Verlag.

Lewis, D., D., & Ringuette, M., (1994) Comparison of two learning algorithms for text categorization, *in SDAIR,* 81-93.

McGinty, L., Smyth, B., (2001) Collaborative Case-Based Reasoning: Applications in Personalised Route Planning. ICCBR 2001, Springer Verlag, pp362-376.

McKenna, E., & Smyth, B., (2000) Competence-Guided Case-Base Editing Techniques. *EWCBR 2000,* E.Blanzieri, L. Portinale (eds.), pp186-197, Springer Verlag.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E., (1998) A bayesian approach to filtering junk e-mail. *in AAAI-98 Workshop on Learning for Text Categorization*.

Sebastiani, F., (2002) Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, Vol. 34, No. 1, March 2002, pp. 1–47.

Leake, D.B., Sooriamurthi, R., (2002) Automatically Selecting Strategies for Multi-Case-Base Reasoning. *ECCBR 2002*, eds S. Craw & A. Preece, LNAI 2416, pp204-233, Springer Verlag.

Widmer, G., Kubat, M., (1996) Learning in the Presence of Concept Drift and Hidden Contexts, Machine Learning, Vol 23, pp69-101.

Plaza, E., & Ontanon, S., (2001) Ensemble case-based reasoning: Collaboration policies for multiagent cooperative CBR. In Proceedings of the *Fourth International Conference on CaseBased Reasoning, ICCBR-01*, Berlin, Springer-Verlag.