An Analysis of Case-Base Editing in a Spam Filtering System[*]

Sarah Jane Delany[1], Pádraig Cunningham[2]

[1]Dublin Institute of Technology, Kevin Street, Dublin 8
sarahjane.delany@comp.dit.ie
[2]Trinity College Dublin, Dublin 2
Padraig.Cunningham@cs.tcd.ie

Abstract. Because of the volume of spam email and its evolving nature, any deployed Machine Learning-based spam filtering system will need to have procedures for case-base maintenance. Key to this will be procedures to edit the case-base to remove noise and eliminate redundancy. In this paper we present a two stage process to do this. We present a new noise reduction algorithm called Blame-Based Noise Reduction that removes cases that are observed to cause misclassification. We also present an algorithm called Conservative Redundancy Reduction that is much less aggressive than the state-of-the-art alternatives and has significantly better generalisation performance in this domain. These new techniques are evaluated against the alternatives in the literature on four datasets of 1000 emails each (50% spam and 50% non spam).

1. Introduction

This paper presents an analysis of case-base editing techniques in a case-based reasoning (CBR) system for filtering spam email. The contributions of this work are twofold. First the analysis exercises the best case-base maintenance techniques currently available on a challenging problem with exacting accuracy requirements, namely spam filtering. Second, we present two new techniques for case-base maintenance, one for noise reduction and the other for redundancy reduction that significantly enhance the competence of the case-base.

While a case-based approach to spam filtering has great promise [1-3], a requirement for a deployed system is a process for maintaining the case-base. This is due to the issue of concept drift and the volume of messages that may be involved. A user may receive over a hundred legitimate emails a week and a multiple of that in spam. Our analysis suggests that between 600 and 1000 cases will provide good coverage for a spam filtering system. So there is an ongoing need to discard cases that are not contributing to competence.

The noise reduction technique we present, which we call Blame-Based Noise Reduction (BBNR), extends the competence based modelling ideas of Smyth and

[*] This research was supported by funding from Enterprise Ireland under grant no. CFTD/03/219 and funding from Science Foundation Ireland under grant no. SFI-02IN.1I111.

colleagues [4,5]. Their case coverage measure, used in case selection, indicates how well a case contributes to correctly classifying other cases in the case-base. We extend this model to include the notion of blame or *liability*. We introduce a measure for a case of how often it is the cause of, or contributes to, other cases being incorrectly classified. Traditional noise reduction mechanisms tend to focus on removing the actual cases that are misclassified. However, a misclassified case could have been classified incorrectly due to the retrieved cases that contributed to its classification. In contrast to traditional approaches we attempt to identify those cases *causing* the misclassifications and use this liability information coupled with coverage information to identify training cases we would be better off without. Our evaluation shows that, in the domain of spam-filtering, this is a better way of identifying noisy cases.

Some analysis of case-base editing techniques in the past has presented algorithms that aggressively prune the case-base at the cost of some classification accuracy [6,7]. This is not acceptable in spam filtering and our technique for redundancy reduction, which we call Conservative Redundancy Removal (CRR), focuses on a more conservative reduction of the case-base. It uses the competence characteristics of the case-base to identify and retain border cases.

This paper begins with a review of existing research on case-base editing techniques in Section 2. The enhanced competence model and our new case editing techniques are presented in Section 3. A comprehensive evaluation of these techniques on four email datasets is presented in Section 4. Some conclusions and directions for future work are presented in Section 5.

2. Review of Existing Case Editing Algorithms

Case base editing techniques involve reducing a case-base or training set to a smaller number of cases while endeavouring to maintain or even improve the generalization accuracy. There is significant research in this area, described in this section.

2.1. Early Techniques

Case editing techniques have been categorised by [8] as *competence preservation* or *competence enhancement* techniques. Competence preservation corresponds to redundancy reduction, removing superfluous cases that do not contribute to classification competence. Competence enhancement is effectively noise reduction, removing noisy or corrupt cases from the training set. Editing strategies normally operate in one of two ways; *incremental* which involves adding selected cases from the training set to an initially empty edited set, and *decremental* which involves contracting the training set by removing selected cases.

An early competence preservation technique is Hart's Condensed Nearest Neighbour (CNN) [9]. CNN is an incremental technique which adds to an initially empty edited set any case from the training set that cannot be classified correctly by the edited set. This technique is very sensitive to noise and to the order of presentation of the training set cases. Ritter [10] reported improvements on the CNN with his

Selective Nearest Neighbour (SNN) which imposes the rule that every case in the training set must be closer to a case of the same class in the edited set than to any other training case of a different class. Gates [11] introduced a decremental technique which starts with the edited set equal to the training set and removes a case from the edited set where its removal does not cause any other training case to be misclassified. This technique will allow for the removal of noisy cases but is sensitive to the order of presentation of cases.

Competence enhancement or noise reduction techniques start with Wilson's Edited Nearest Neighbour (ENN) algorithm [12], a decremental strategy, which removes cases from the training set which do not agree with their *k* nearest neighbours. These cases are considered to be noise and appear as exceptional cases in a group of cases of the same class. Tomek [13] extended this with his repeated ENN (RENN) and his "all *k*-NN" algorithms. Both make multiple passes over the training set, the former repeating the ENN algorithm until no further eliminations can be made from the training set and the latter using incrementing values of *k*. These techniques focus on noisy or exceptional cases and do not result in the same storage reduction gains as the competence preservation approaches.

Competence preservation techniques aim to remove internal cases in a cluster of cases of the same class and can predispose towards preserving noisy cases as exceptions or border cases. Noise reduction on the other hand aims to remove noisy or corrupt cases but can remove exceptional or border cases which may not be distinguishable from true noise, so a balance of both can be useful. Later editing techniques can be classified as hybrid techniques incorporating both competence preservation and competence enhancement stages. Aha et al. [14] presented a series of instance based learning algorithms to reduce storage requirements and tolerate noisy instances. IB2 is similar to CNN adding only cases that cannot be classified correctly by the reduced training set. IB2's susceptibility to noise is handled by IB3 which records how well cases are classifying and only keeps those that classify correctly to a statistically significant degree. Other researchers have provided variations on the IB*n* algorithms [15,16,17].

2.2. Competence-based Editing

More recent approaches to case editing build a competence model of the training data and use the competence properties of the cases to determine which cases to include in the edited set. Measuring and using case competence to guide case-base maintenance was first introduced by Smyth and Keane [5] and developed by Zhu and Yang [18]. Smyth and McKenna [3] introduce two important competence properties, the coverage and reachability sets for a case in a case-base. These are discussed in Section 3. The coverage and reachability sets represent the local competence characteristics of a case and are used as the basis of a number of editing techniques.

McKenna & Smyth [6] presented a family of competence-guided editing methods for case-bases which combine both incremental and decremental strategies. The family of algorithms is based on four features; (1) an ordering policy for the presentation of the cases that is based on the competence characteristics of the cases; (2) an addition rule to determine the cases to be added to the edited set, (3) a deletion rule to determine the cases to be removed from the training set and (4) an update policy which indicates whether the competence model is updated after each editing step. The different combinations of ordering policy, addition rule, deletion rule and update policy produce the family of algorithms.

Brighton and Mellish [8] also use the coverage and reachability properties of cases in their Iterative Case Filtering (ICF) algorithm. The ICF is a decremental strategy contracting the training set by removing those cases *c*, where the number of other cases that can correctly classify *c* is higher that the number of cases that *c* can correctly classify. This strategy focuses on removing cases far from class borders. After each pass over the training set, the competence model is updated and the process repeated until no more cases can be removed. ICF includes a pre-processing noise reduction stage, effectively RENN, to remove noisy cases.

McKenna and Smyth compared their family of algorithms to ICF and concluded that the overall best algorithm of the family delivered improved accuracy (albeit marginal, 0.22%) with less than 50% of the cases needed by the ICF edited set [6].

Wilson & Martinez [7] present a series of Reduction Technique (RT) algorithms, RT1, RT2 and RT3 which, although published before the definitions of coverage and reachability, could also be considered to use a competence model. They define the set of *associates* of a case *c* which is comparable to the coverage set of McKenna & Smyth except that the associates set will include cases of a different class from case *c* whereas the coverage set will only include cases of the same class as *c*. The RT*n* algorithms use a decremental strategy. RT1, the basic algorithm, removes a case *c* if at least as many of its associates would be classified correctly without *c*. This algorithm focuses on removing noisy cases and cases at the centre of clusters of cases of the same class as their associates will most probably still be classified correctly without them. RT2 fixes the order of presentation of cases as those furthest from their nearest unlike neighbour (i.e. nearest case of a different class) to remove cases furthest from the class borders first. RT2 also uses the original set of associates when making the deletion decision, which effectively means that the associate competence model is not rebuilt after each editing step which RT1 does. RT3 adds a noise reduction pre-processing pass based on Wilson's noise reduction algorithm.

Wilson & Martinez concluded from their evaluation of the RT*n* algorithms against IB3 that RT3 had a higher average generalization accuracy and lower storage requirements overall but that certain datasets seem well suited to the techniques while others were unsuited. Brighton & Mellish evaluated their ICF against RT3 and found that neither algorithm consistently out performed the other and both represented the "cutting edge in instance set reduction techniques".

3. Editing using an Enhanced Competence Model

Smyth and McKenna's competence model defines how well a case performs when classifying other cases in the case-base. We have extended this competence model to include how badly a case performs when classifying other cases. This section firstly discusses our extensions to the competence model and then shows how they can be used in an alternative noise reduction algorithm BBNR that focuses on apportioning blame for misclassifications. We also present our competence-based redundancy reduction algorithm CRR which aims to maintain (and even improve) the generalisation accuracy of the case-base by focussing on less aggressive pruning of cases compared to that performed by many of the existing editing techniques.

3.1. The Enhanced Competence Model

Smyth and McKenna's case-base competence modelling approach proposes two sets which model the local competence properties of a case within a casebase; the *reachability set* of a target case *t* is the set of all cases that can successfully classify *t,* and the *coverage set* of a target case *t* is the set of all cases that *t* can successfully classify. Using the case-base itself as a representative of the target problem space, these sets can be estimated as shown in definitions 1 and 2.

$$Coverage\ Set(t \in C) = \{c \in C : Classifies(c,t)\} \qquad (1)$$

$$Reachability\ Set(t \in C) = \{c \in C : Classifies(t,c)\} \qquad (2)$$

where *Classifies(a,b)* means that case *b* contributes to the correct classification of target case *a*. This means that target case *a* is successfully classified and case *b* is returned as a nearest neighbour of case *a* and has the same classification as case *a*.

We propose to extend the model to include an additional property; the *liability set* of a case *t* which is defined as the set of all cases that *t* causes to be misclassified or contributes to being misclassified, see definition 3.

$$LiabilitySet(t \in C) = \{c \in C : Misclassifies(c,t)\} \qquad (3)$$

where *Misclassifies(a,b)* means that case *b* contributes in some way to the incorrect classification of target case *a*. In effect this means that when target case *a* is misclassified by the case-base, case *b* is returned as a neighbour of *a* but has a different classification to case *a*. For *k*-NN with *k*=1, case *b* causes the misclassification but for *k*>1 case *b* contributes to the misclassification. Case *a* is therefore a member of the liability set of case *b*.

3.2 Blame Based Noise Reduction (BBNR)

Although a number of the competence-based editing techniques described in section 2 are designed to focus on removing redundant cases, all of them include both noise reduction and redundancy reduction stages. The noise reduction stage used by all the techniques is based on Wilson's noise reduction.

Noisy cases can be considered as training cases that are incorrectly labelled. Wilson's noise reduction technique removes from the case-base cases that would be misclassified by the other cases, implying that these are incorrectly labelled and are therefore noisy cases. However, a misclassified case may not necessarily be a noisy case but could be classified incorrectly due to the retrieved cases which contribute to its classification. Mislabelled cases which are retrieved as nearest neighbours of a target case can affect the classification of the target case. Therefore just because a case is misclassified does not imply that it is noise and should be removed.

Our BBNR approach emphasises the cases that *cause* misclassifications rather than the cases that *are* misclassified. In effect we are not just accepting the presumption that if a case is misclassified it must be mislabelled but try to analyse the cause of the misclassification. In our policy on noise reduction we attempt to remove mislabelled cases; we also remove "unhelpful" cases that cause misclassification. For example, a case that represents an actual spam email but looks just like a legitimate email.

The liability set captures this information. The greater the size of the liability set of a case, the more impact it has had on misclassifying other cases within the case-base. It is however important to consider this in light of how well cases are performing, how often they actually contribute to correct classifications. The coverage set captures this information. Our BBNR technique looks at all cases in the case-base that have contributed to misclassifications (i.e. have liability sets with at least one element). For each case *c* with a liability set of at least one element, if the cases in c's coverage set can still be classified correctly without *c* then *c* can be deleted. The BBNR algorithm is described in Figure 1.

```
Blame-based Noise Reduction (BBNR) Algorithm

T, Training Set
/* Build case-base competence model */
For each c in T
   CSet(c) ← Coverage Set of c
   LSet(c) ← Liability Set of c
End-For
/* Remove noisy cases from case-base */
TSet ← T sorted in descending order of LSet(c) size
c ← first case in TSet
While |LSet(c)| >0
   TSet ← TSet - {c}
   misClassifiedFlag ← false
   For each x in CSet(c)
      If x cannot be correctly classified by TSet
         misClassifiedFlag ← true
         break
      End-If
   End-For
   If misClassifiedFlag = true
      TSet ← TSet + {c}
   End-If
   c ← next case in TSet
End-While
```

Fig. 1. Blame-Based Noise Reduction Algorithm

This principle of identifying damaging cases is also there in IB3. Aha's IB3 algorithm is an algorithm more applicable for data streams and online learning in that the training set does not exist as a collection of cases before editing can be performed. The decision as to whether cases are kept in the case-base or not is made as the cases are presented.

There are a number of differences between IB3 and BBNR. First, IB3 maintains the classification records during the editing process rather than using the competence of the full training set as BBNR does through use of the competence model. Secondly, the classification record maintained by BBNR is based on actual classifications, whereas that maintained by IB3 is based on possible or potential classifications. IB3 updates the classification record of all cases that could potentially be neighbours whereas BBNR only uses the *k* retrieved neighbours to build its competence model. However, the most significant difference between the two algorithms is how they use case liability information. Although IB3 does collect information on the likely damage that certain cases may cause, it is not used actively to determine whether these potentially damaging cases should be removed or not. IB3 uses the classification accuracy, rather than classification error, to indicate how well a case is performing and waits for a case not to classify correctly at a satisfactory level before removing it. BBNR, on the other hand, uses the liability information available from the competence model of the case-base to decide whether these potentially damaging cases have any merit in being kept in the case-base.

3.3 Conservative Redundancy Reduction

The second stage in our competence-based editing technique is to remove redundant cases. Our proposed algorithm for removing redundant cases is based on identifying cases that are near class borders. The coverage set of a case captures this information. A large coverage set indicates that a case is situated in a cluster of cases of the same classification whereas a small coverage set indicates a case with few neighbours of the same classification. Cases near the class border will have small coverage sets. Cases with small coverage sets are presented first to be added to the edited set. For each case added to the edited set, the cases that this case can be used to classify (that is the cases that this case covers) are removed from the training set. This is the same as McKenna & Smyth's *coverage deletion rule* [6]. The CRR algorithm is presented in Figure 2.

```
Conservative Redundancy Removal(CRR) Algorithm

T, Training Set
/* Build case-base competence model */
For each c in T
   CSet(c) ← Coverage Set of c
End-For
/* Remove redundant cases from case-base */
ESet ← {}, (Edited Set)
TSet ← T sorted in ascending order of CSet(c) size
c ← first case in TSet
While TSet ≠ {}
   ESet ← ESet + {c}
   TSet ← TSet - CSet(c)
   c ← next case in TSet
End-While
```

Fig. 2. Conservative Redundancy Removal Algorithm

Existing editing techniques are very aggressive in their pruning of cases. Various cross validation experiments using existing techniques (ICF, RT*n* and a number of

McKenna & Smyth's algorithmic variations) over our four datasets produced edited case-base sizes ranging from 3.5% to 46.4% of original case-base size with the average edited size of 22%. Such aggressive reductions in case-base size can have a detrimental effect on generalisation accuracy. By adding the cases near class borders to the edited set first, rather than working in the reverse order (that is with cases that are in the centre of a large cluster of cases of the same classification), our coverage deletion rule results in a more conservative reduction of the case-base. This, as shown in Section 4.4, results in larger edited case-bases and improved generalisation accuracy.

4. Evaluation

This section presents our results at two levels; firstly, an evaluation of the performance of our competence-based BBNR algorithm against Wilson's noise reduction as used by a majority of existing case editing techniques and secondly, an evaluation of the performance, in the domain of spam filtering, of existing case-based editing techniques compared with our new two-phased Competence-Based Editing technique incorporating BBNR and CRR.

4.1. Experimental Setup

The objective is to find a suitable case-base editing technique to reduce a case-base of spam and non-spam cases while maintaining case-base accuracy. Four datasets were used. The datasets were derived from two corpora of email collected by two individuals over a period of one year. Two datasets of one thousand cases were extracted from each corpus. Each included five hundred spam emails and five hundred non-spam or legitimate emails. Datasets 1.1 and 2.1 consisted of emails received up to and including February 2003 while datasets 1.2 and 2.2 consisted of emails received between February 2003 and November 2003. Given the evolving nature of spam it was felt that these datasets gave a representative collection of spam.

The emails were not altered to remove HTML tags and no stop word removal, stemming or lemmatising was performed. Since the datasets were personal it was felt that certain headers may contain useful information, so a subset of the header information was included. Each email, e_i was reduced to a vector of features $e_i = \langle x_1, x_2, \mathrm{K}, x_n \rangle$ where each feature is binary. If the feature exists in the email, $x_i = 1$, otherwise $x_i = 0$. It is more normal in text classification for lexical features to convey frequency information but our evaluation showed that a binary representation works better in this domain. We expect that this is due to the fact that most email messages are short. Features were identified using a variety of generic lexical features, primarily by tokenising the email into words. No domain specific feature identification was performed at this stage although previous work has indicated that the efficiency of filters will be enhanced by their inclusion [19].

Feature selection was performed to reduce the dimensionality of the feature space. Yang and Petersen's evaluation of dimensionality reduction in text categorisation found that Information Gain (IG) [20] was one of the top two most effective techniques in aggressive feature removal without losing classification accuracy [21]. Using IG with a *k*-nearest neighbour classifier, 98% removal of unique terms yielded an improved classification accuracy. The IG of each feature was calculated and the top 700 features were selected. Cross validation experiments, varying between 100 and 1000 features across the 4 datasets, indicated best performance at 700 features.

The classifier used was *k*-nearest neighbour with $k=3$. Due to the fact that false positives are significantly more serious than false negatives the classifier used unanimous voting to determine whether the target case was spam or not. All neighbours returned had to have a classification of spam in order for the target case to be classified as spam.

4.2. Evaluation Metrics

Previous studies into case editing techniques have compared performance on two measures; the accuracy of the edited casebase and the resulting size of the edited casebase. In the domain of spam filtering size and accuracy are not adequate measures of performance. A False Positive (FP), a legitimate email classified incorrectly as spam, is significantly more serious than a False Negative (a spam email incorrectly classified as a legitimate email). The occurrence of FPs needs to be minimised, if not totally eliminated. Accuracy (or error) as a measure, does not give full transparency with regard to the numbers of FPs and FNs occurring. Two filters with similar accuracy may have very different FP and FN rates.

Previous work on spam filtering use a variety of measures to report performance. The most common performance metrics are precision and recall [8]. Sakkis *et al.* [3] introduces a weighted accuracy measure which incorporates a measure of how much more costly a FP is than a FN. Although these measures are useful for comparison purposes, the FP and FN rate are not clear so the base effectiveness of the classifier is not evident. For these reasons we will report the error rate, the rates of FPs and the rate of FNs. For information purposes we will also indicate the resulting sizes of the edited case-bases.

A final justification for reporting this set of metrics is the fact that it reflects how commercial spam filtering systems are evaluated on the web and in the technical press.

4.3 Evaluation Methods

For each dataset we used 20 fold cross-validation, dividing the dataset into 20 stratified divisions or folds. Each fold in turn is considered as a test set with the remaining 19 folds acting as the training set. For each test fold and training set combination we calculated the performance measures for the full training set without editing and the performance measures for the training set edited with each selected editing technique. Where one case-base editing technique appeared to out perform another, confidence levels were calculated using a *t*-test on the paired fold-level results.

The case editing techniques that we evaluated include ICF, RT2, RT3 and a selection of the McKenna & Smyth's family of case editing techniques described in Section 2. The McKenna & Smyth algorithms can be identified as "*adc_o"*; where *a* indicates whether the addition rule is used (True/False), *d* indicates whether the deletion rule is used (T/F), *c* indicates whether the competence model is updated (T/F) and *o* indicates the order of presentation of cases. Their top two performing algorithms are FTF_*o* and FTT_*o*, where the addition rule is not used (*a*=F) and the deletion rule is used (*d*=T) irrespective of whether the competence model was rebuilt or not. The top two ordering sequences are order by relative coverage (RC) and reach for cover (RFC) [6]. Preliminary tests indicated those algorithms which require the competence model to be rebuilt after each editing step (i.e. FTT_RC and FTT_RFC) were not significantly different in accuracy but were prohibitively computationally expensive and were discarded.

4.4. Results

Figure 3 shows the results of comparing BBNR with RENN across the 4 datasets and the overall average results across all datasets. The graphs show percentage values for error, FP and FN rates. The average size across all 20 folds of the edited casebase is indicated (on the x-axis) as a percentage of the unedited training case-base size for the individual datasets.

The results can be summarised as follows:

- BBNR performs very well and has a lower error rate than RENN (significant at confidence level 99.9% across all datasets). There are also significant improvements in FP rate and FN rate (at 99.9% level).
- The individual training sets reduced with BBNR have error rates that are at least as good as or better than the unedited training sets with the overall average showing significant improvement in FN rate and error rate at 99.9% level and FP rate at 99% level.

As BBNR shows better performance than Wilson noise reduction in the spam domain, we also evaluated replacing the noise reduction stage of those competence based case editing techniques with BBNR. Figure 4 displays these results for ICF, FTF_RC and FTF_RFC. Technique *X* with the Wilson based noise reduction phase replaced by BBNR is labelled as *X-bbnr* in Figure 4. Although RT2 and RT3 could be considered competence-based editing techniques, they use a different competence model without a liability set so BBNR was not applied to these. Figure 4 also includes overall average results across all datasets. The results can be summarised as follows:

- Using BBNR to perform the noise reduction stage improves the overall performance across all the datasets for techniques ICF, FTF_RC and FTF_FRC with significant improvements in FP, FN and error rates at 99.9% level or higher.
- Using BBNR for noise reduction in each editing technique improves performance in average error, FP and FN rates over the unedited training sets for ICF-bbnr (at levels of 95% or higher) and FTF_RFC-bbnr (at 90% level or higher). Although FTF_RC-bbnr's FP rate shows significant improvement (at

99.9% level) its deterioration in FN rate leads to an overall deterioration in error rate.

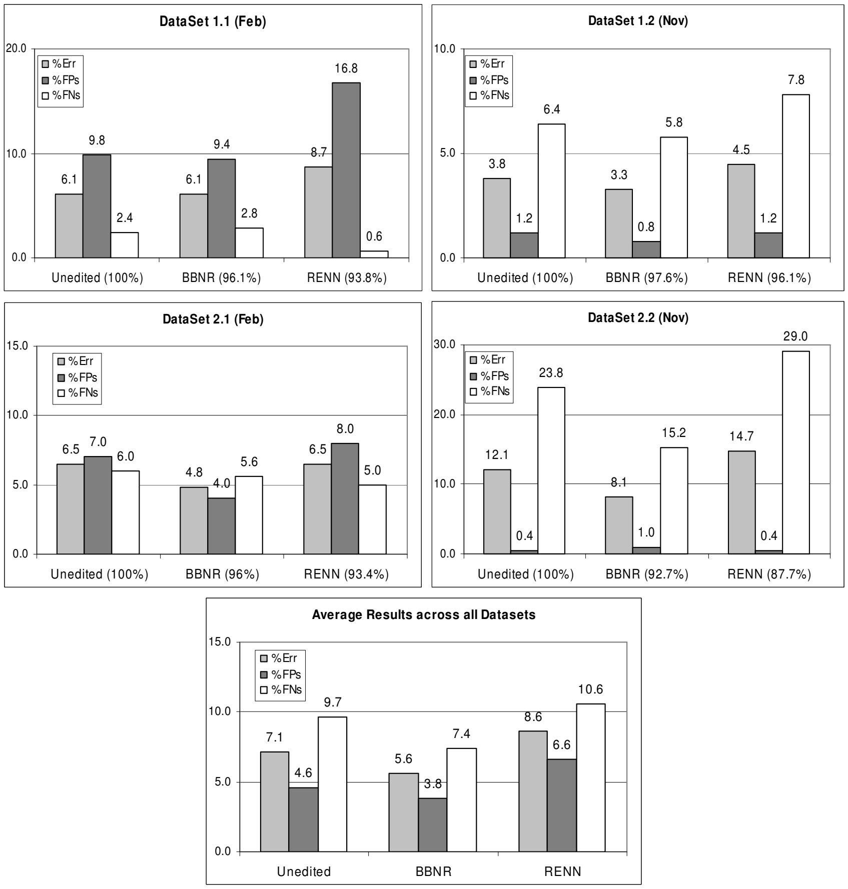

Fig. 3. Results of BBNR versus RENN.

Figure 4 also includes results for RT2 and our new Competence-Based Editing (CBE) technique (i.e. BBNR+CRR). Results for RT3 were not included as RT2 outperformed RT3 for these datasets. The results for CBE can be summarised as follows:

- Taking average results across all datasets, CBE significantly improves (at 99.9% level) the generalisation accuracy achieved on an unedited training set of cases. The FP rate is reduced (significant at 99.9% level) as is the FN rate (significant at 97% level).
- CBE and FTF-RFC-bbnr are the best performing editing techniques on average across all datasets with the lowest average error rates (significant at 90% level).

- McKenna & Smyth's FTF_RFC technique with the noise reduction stage replaced by BBNR is a close second to CBE. It also demonstrates improved accuracy in average error, FP and FN rates when compared with an unedited training set, however, the improvements are at a lower level of significance.
- It may appear that CBE is out performed in specific datasets by other techniques, e.g. by RT2 in dataset 2.1 or ICF-bbnr in dataset 1.2. However CBE demonstrates the most consistent performance across all datasets.

It is interesting to note that CBE and FTF_RFC-bbnr (the top two editing techniques) result in the largest average edited casebase size (69% for CBE and 43% for FTF_RFC-bbnr).

6. Conclusions and Further Work

We have argued that a key component in any operational Machine Learning based spam filtering system will be procedures for managing the training data. Because of the volume of the training data a case-base editing process will be required. We have presented a novel competence-based procedure which we call CBE for this. CBE has two stages, a noise reduction phase called BBNR and a redundancy elimination phase called CRR.

BBNR focuses on the damage that certain cases are causing in classifications. Comparative evaluations of this algorithm with the standard Wilson's noise reduction technique in the domain of spam filtering have shown an improved performance across all four datasets. Experiments incorporating BBNR into existing competence-based case-base editing techniques have shown that BBNR improves all these techniques over the four datasets on which it was evaluated.

Our redundancy reduction process (CRR) was motivated by the observation that state-of -the-art techniques were inclined to be too aggressive in removing cases and tended to result in some loss of generalisation accuracy – at least in this domain. This is in effect a tendency to overfit the training data by finding minimal sets that cover the data. CRR is much more conservative in removing cases and produces larger edited case-bases that have the best generalisation accuracy in this domain.

This research will continue along two lines. We will continue working on case-base management for spam filtering, focusing next on managing concept drift. We will also evaluate CRR and BBNR in other domains to see if the good generalisation performance we have found on spam is replicated elsewhere.

References

1. Cunningham, P., Nowlan, N., Delany, S.J., Haahr, M., A Case-Based Approach to Spam Filtering that Can Track Concept Drift, *The ICCBR'03 Workshop on Long-Lived CBR Systems,* Trondheim, Norway, (2003).
2. Androutsopoulos, I., Koutsias, J., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., & Stamatopoulos, P..: 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) (2000)

3. Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos C.D., &. Stamatopoulos, P., A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists Information Retrieval, Vol 6 No 1, Kluwer (2003) 49-73
4. Smyth, B., McKenna, E.: Modelling the Competence of Case-Bases. In: Smyth, B. and Cunningham, P. (eds.): Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence, Springer-Verlag (1998) 208-220
5. Smyth, B., Keane, M.: Remembering to Forget: A Competence Preserving Case Deletion Policy for CBR Systems. In: Mellish, C. (ed.): Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann (1995) 337-382
6. McKenna, E., Smyth, B.: Competence-guided Editing Methods for Lazy Learning. In Proceedings of the 14th European Conference on Artificial Intelligence, Berlin (2000)
7. Wilson, D.R., Martinez, T.R.: Instance Pruning Techniques. In: Fisher, D. (ed.) Proceedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, C.A. (1997) 404-411
8. Brighton.H., & Mellish. C.: Advances in Instance Selection for Instance-Based Learning Algorithms. In: Data Mining and Knowledge Discovery, Vol. 6. Kluwer Academic Publishers, The Netherlands (2002) 153-172
9. Hart, P.E.: The Condensed Nearest Neighbor Rule. IEEE Transactions on Information Theory. Vol. 14, No. 3 (1968) 515-516
10. Ritter, G.L., Woodruff, H.B., Lowry, S.R., Isenhour, T.L.: An Algorithm for a Selective Nearest Neighbor Decision Rule. IEEE Transactions on Information Theory, Vol. 21, No. 6 (1975) 665-669
11. Gates, G.W. : The Reduced Nearest Neighbor Rule. IEEE Transactions on Information Theory, Vol. 18, No. 3 (1972) 431-433
12. Wilson, D.L.: Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Transactions on Systems. Man, and Cybernetics, Vol. 2, No. 3 (1972) 408-421
13. Tomek, I.: An Experiment with the Nearest Neighbor Rule. IEEE Transactions on Systems, Man, and Cybernetics, Vol 6. No. 6 (1976) 448-452
14. Aha, D.W., Kibler, D., Albert, M.K.: Instance-Based Learning Algorithms. Machine Learning, Vol. 6 (1991) 37-66
15. Zhang, J.: Selecting Typical Instances in Instance-Based Learning. In: Proceedings of the Ninth International Conference on Machine Learning (1992) 470-479
16. Cameron-Jones, R.M.: Minimum Description Length Instance-Based Learning. In: Proceedings of the Fifth Australian Joint Conference on Artificial Intelligence (1992) 368-373
17. Brodley, C.: Addressing the Selective Superiority Problem: Automatic Algorithm/Mode Class Selection. In: Proceedings of the Tenth International Machine Learning Conference (1993) 17-24
18. Zhu, J., Yang, Q.: Remembering to Add: Competence Preserving Case-Addition Policies for Case-Base Maintenance. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann (1997) 234-239
19. Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E., A Bayesian Approach to Filtering Junk Email, In: AAAI-98 Workshop on Learning for Text Categorization pp. 55-62, Madison ,Wisconsin. AAAI Technical Report WS-98-05, (1998).
20. Quinlan, J. Ross: C4.5 Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA. (1993).
21. Yang Y., Pedersen J.O.: A comparative study on feature selection in text categorization. In: Proceedings of ICML-97, 14th International Conference on Machine Learning, Nashville, US, (1997) 412–420.

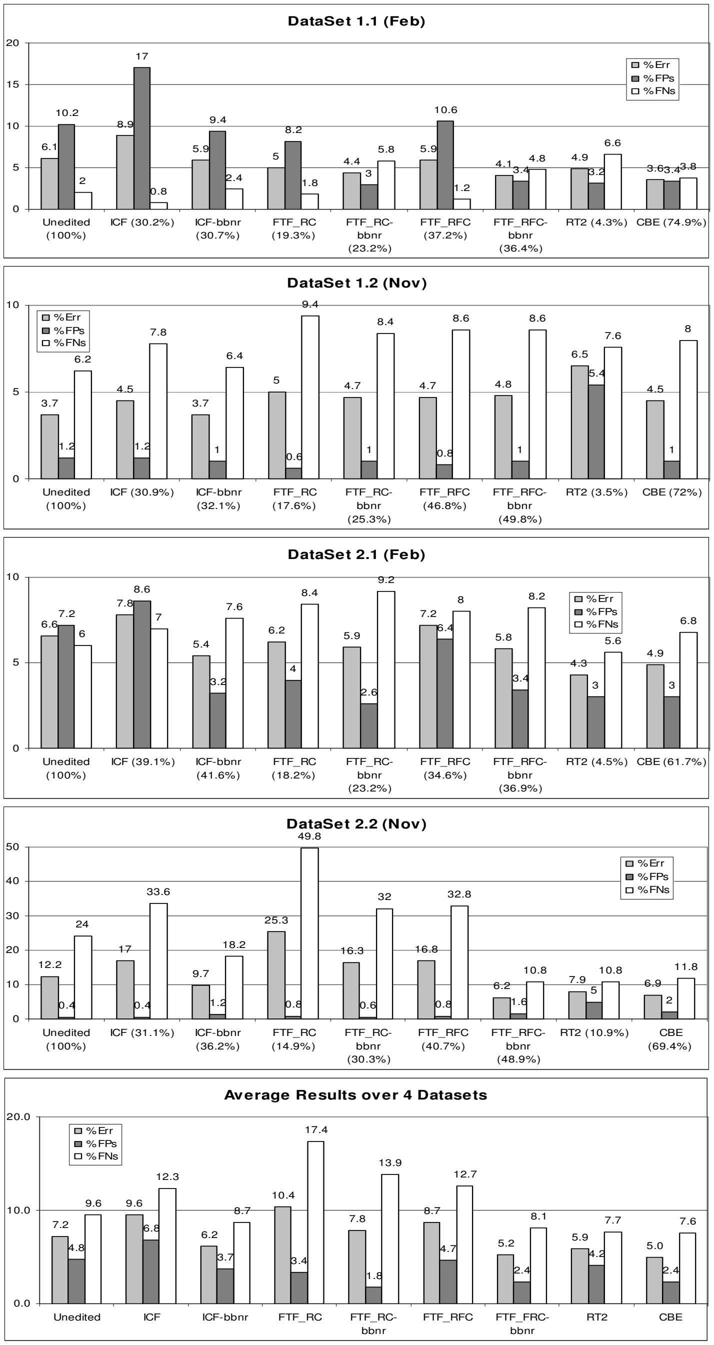

Fig. 4. Results of various case editing techniques