# A Neural Network Based Approach to Automated E-mail Classification

James Clark, Irena Koprinska, and Josiah Poon
*School of Information Technologies, University of Sydney, Sydney, Australia*
*{jclark, irena, josiah}@it.usyd.edu.au*

## Abstract

*In this paper we present a neural network based system for automated e-mail filing into folders and anti-spam filtering. The experiments show that it is more accurate than several other techniques. We also investigate the effects of various feature selection, weighting and normalization methods, and also the portability of the anti-spam filter across different users.*

## 1. Introduction

The volume of e-mail that we get is constantly growing. We are spending more and more time filtering e-mails and organizing them into folders in order to facilitate retrieval when necessary. The rate of unsolicited (spam) e-mail is also rapidly increasing. It may vary significantly in content, e.g. from get-rich and selling items, to offensive e-mails and pornographic sites.

Most modern e-mail software packages provide some form of programmable filtering, typically in the form of rules that organize mail into folders or dispose of spam mail based on keywords detected in the header or body. However, most users avoid customizing software. In addition, manually constructing robust rules is difficult as users are constantly creating, deleting and reorganizing their folders. Even if the folders remain the same, the nature of the e-mails within the folder may well drift over time. The characteristics of the spam e-mail (e.g. topics, frequent terms) also change over time. Hence, the rules must be constantly tuned by the user, that is time consuming and error-prone. A system that can automatically learn how to classify e-mails into a set of folders and filter spam e-mails is highly desirable.

Several systems for automatic e-mail classification based on Text Categorization (TC) have been developed. Cohen [5] uses the propositional learner Ripper to induce “keyword-spotting rules" for filing e-mails into folders. Sahami et al. [8] applied Baysian networks for spam e-mail filtering using bag of words representation and binary encoding. The integration of hand-crafted phrases and domain-specific features improved the results. Rennie’s iFile [7] uses a Naïve Bayes (NB) to file e-mails into folders. In [2] was reported that NB and a k-NN technique (TiMBL) clearly outperform the keyword-based filter of Outlook 2000 on the LingSpam corpora. Ensembles of classifiers were also used for spam filtering. Stacked NB and kNN resulted in better accuracy [9]. Boosted trees were shown to outperform decision trees (DT), NB and kNN in [3].

There have been, however, little studies in applying neural networks (NNs). The main disadvantage of NNs is that they require considerable time for parameter selection and training. On the other hand, previous research has shown that NNs can achieve very accurate results, that are sometimes more accurate than those of the symbolic classifiers. NNs have been successfully applied in many real-world tasks. In this paper we present LINGER - a NN-based system for automatic e-mail classification.

## 2. LINGER

Although LINGER was tested in the domain of email classification, it is a generic architecture for all kinds of TC. It is highly flexible and uses configurable options for most of its operation. It consists of two main modules: preprocessing and classification.

### 2.1. Preprocessing module

LINGER uses the *bag of words* representation which is the most commonly used in TC. All unique words in the entire training corpus are identified. A feature selection is applied to choose the most important words and reduce dimensionality. Each document is then represented by a vector that contains a normalized weighting for every word according to its importance.

**2.1.1. Preprocessing for words extraction.** The e-mail’s body, subject, sender, and recipient were parsed and tokenized. Attachments are considered parts of the body and processed in their original format (binary, text, html). These fields were treated equally and a single bag of words was created for each e-mail. The following symbols were used as delimiters and then discarded: <>()[]{}/\|-_#%^&*,.:;@~`+="'. Three other (!, $, ?) were used as delimiters and then kept as they often appear in spam e-mail. No stemming or stop wording were used.

Next, words that only appear once in each corpus were discarded. These are typically useless strings in html formatted email or binary and html attachments. Finally, words longer than 20 characters were removed from the

0-7695-1932-6/03 $17.00 © 2003 **IEEE**

body as they are usually strings in binary attachments As a result, the initial number of unique words is reduced from about 9000 to 1000 for each corpus.

**2.1.2. Feature selection, weighting and normalization.** LINGER includes two feature selectors: information gain (IG) and variance (V). Several weighting schemes are incorporated: binary, term frequency, term frequency inverse document frequency (tf-idf). Finally, LINGER allows the weight normalization to be done at three different levels: e-mail, mailbox and global (corpus) level

### 2.2. Classification module

A fully-connected multilayer perceptron trained with the backpropagation algorithm was used as a classifier. For each user, it learns from a set of e-mails with assigned mailboxes. The current assumption is that each email is only assigned to one mailbox. The NN classifier is a multi-class one, i.e. there is one output neuron for each mailbox. One hidden layer of 20-40 neurons was found to work best. Early stopping based on validation set accuracy in conjunction with maximum number of epochs (10 000) were used as stopping criteria.

## 3. Corpora used for evaluation

### 3.1 General e-mail classification

We used the e-mail corpus collected by [6] that contains messages from four different users (U1-4). In addition, we also used the e-mail of the first author, collected over the last one year (U5). E-mail corpora characteristics are given in Table 1.

**Table 1. Statistics for general e-mail corpora**

| corpus | # e-mails | # folders |
|---|---|---|
| U1 | 545 | 7 |
| U2 | 423 | 6 |
| U3 | 888 | 11 |
| U4 | 926 | 19 |
| U5 | 982 | 6 |

**Table 2. Statistics for spam filtering corpora**

| corpus | # e-mails | # spam | # legit. |
|---|---|---|---|
| PU1 | 1099 | 481 | 618 |
| LingSpam | 2893 | 481 | 2412 |
| U5Spam | 282 | 82 | 200 |

### 3.2 Spam filtering

The size of the three corpora used is shown in Table 2. The first two are publicly available [1,2], the third one is a subset of the U5 corpus. PU1 is encrypted for privacy reasons and contains personal and spam messages. LingSpam contains e-mails sent to the Linguist mailing list mixed with spam e-mails. There are four versions of PU1 and LingSpam depending on whether stemming and stop word list were used (*bare* - both disabled; *lemm* – only stemming used; *stop* – only stop wording used; *lemm_stop* – both used. Each of these versions is partitioned into 10 stratified folds in order to facilitate evaluation using 10-fold cross validation.

## 4. Results and discussion

In all experiments IG and V were used as feature selection criteria and the best scoring 256 features were chosen. Term frequency with mailbox level normalization were used unless otherwise noted.

### 4.1. General e-mail classification

**4.1.1. Performance measures.** To evaluate performance we calculated accuracy (A), recall (R), precision (P) and F1 measure (F1). In the multi class task of general mail classification, P, R and F1 were calculated for each class and the results averaged. Stratified ten-fold cross validation was used in all experiments.

**4.1.2. Overall performance.** Table 3 shows that the simpler feature selector (V) was more effective than IG.

**Table 3. Performance on e-mail clasification**

| feature sel. | A [%] | R [%] | P [%] | F1 [%] | epochs |
|---|---|---|---|---|---|
| U1 | | | | | |
| V | 86.23 | 67.28 | 71.02 | 69.10 | 2150 |
| IG | 88.08 | 77.77 | 81.12 | **79.44** | 7960 |
| U2 | | | | | |
| V | 68.54 | 41.27 | 45.11 | **43.10** | 702 |
| IG | 61.46 | 37.07 | 41.94 | 39.36 | 1738 |
| U3 | | | | | |
| V | 92.34 | 76.18 | 77.45 | **76.81** | 3898 |
| IG | 74.55 | 45.02 | 47.41 | 46.18 | 5620 |
| U4 | | | | | |
| V | 79.48 | 40.39 | 39.46 | **39.92** | 4616 |
| IG | 65.23 | 24.97 | 22.51 | 23.68 | 4250 |
| U5 | | | | | |
| V | 83.40 | 81.66 | 84.75 | **83.18** | 4220 |
| IG | 70.16 | 64.79 | 71.32 | 67.90 | 9032 |

It can also be noticed that the e-mails of U2 and U4 were harder to classify than those of U1, U3 and U5. This is due to the different classification styles. While U1, U3 and U5 (like most users) categorized e-mails based on the topic and sender, U2 do this totally based on the action performed (e.g. Read&Keep, ToActOn) while U4 uses all strategies - based on the topic, sender, action and also when e-mails needed to be acted upon (e.g. ThisWeek). Thus, some mailboxes of U2 and U4 contain e-mails grouped by action and time which complicates learning. The classifier cannot determine the priority of an e-mail

| | λ=1 | | | | | λ=9 | | λ=999 | |
|---|---|---|---|---|---|---|---|---|---|
| | A [%] | SR [%] | SP [%] | SF1 [%] | TCR | WA [%] | TCR | WA [%] | TCR |
| LingSpam | | | | | | | | | |
| LINGER-V | 98.20 | 93.56 | 95.62 | 94.58 | 9.24 | 99.01 | 2.18 | 99.13 | 0.02 |
| LINGER-IG | **100** | **100** | **100** | **100** | ∝ | **100** | ∝ | **100** | ∝ |
| NB | 96.93 | 82.40 | 99.00 | 89.94 | 5.41 | 99.43 | 3.82 | 99.99 | N/Av |
| k-NN | N/Av | 88.60 | 97.40 | 92.79 | 7.18 | 99.40 | 3.64 | N/Av | N/Av |
| Stacking | N/Av | 91.70 | 96.50 | 93.93 | 8.44 | 99.46 | 3.98 | N/Av | N/Av |
| Stumps | N/Av | 97.92 | 98.33 | 98.12 | N/Av | 99.76 | N/Av | 99.95 | N/Av |
| TreeBoost | N/Av | 97.30 | 98.53 | 97.91 | N/Av | 99.77 | N/Av | 99.99 | N/Av |
| PU1 | | | | | | | | | |
| LINGER-V | 93.45 | 88.36 | 96.46 | 92.23 | 6.68 | 96.69 | 2.4 | 97.41 | 0.03 |
| LINGER-IG | **100** | **100** | **100** | **100** | ∝ | **100** | ∝ | **100** | ∝ |
| NB | 89.80 | 78.14 | 98.25 | 87.05 | 4.29 | 97.18 | 2.83 | 99.32 | 0.11 |
| keywords | 78.25 | 53.01 | 95.15 | 68.09 | 2.01 | 94.32 | 1.40 | 97.86 | 0.04 |
| DT | N/Av | 89.81 | 88.71 | 89.25 | N/Av | N/Av | N/Av | N/Av | N/Av |
| Stumps | N/Av | 96.47 | 97.48 | 96.97 | N/Av | 98.58 | N/Av | 99.66 | N/Av |
| TreeBoost | N/Av | 96.88 | 98.52 | 97.69 | N/Av | 99.14 | N/Av | 99.98 | N/Av |

**Table 4. Performance on spam filtering for lemm corpora**

based merely on its content, unless it possesses the same background knowledge as the user. Another difficulty is the smaller number of e-mails per folder for U2 and U4. LINGER has been compared with five other classifiers and the results indicate competitive performance, see [4].

**4.1.3. Effect of normalization and weighting.** The effect on accuracy of the various weightings (frequency, tf-idf and boolean) and normalizations (at e-mail, mailbox and global level) has been studied, see [4] for more details. The best results were produced by normalization at the mailbox level. This is a reasonable finding as the aim is to put an e-mail to an appropriate mailbox. Normalizing at the global level will dilute the significance of a keyword under a certain mailbox, while normalizing at the message level will distort the significance of those keywords across many messages in a mailbox. Not surprisingly, the most popular weighting schemes in TC - frequency and tf-idf - were found to be the best in our experiments.

## 4.2. Spam filtering

**4.2.1. Performance measures.** In addition to accuracy, *spam recall (SR), spam precision (SP)* and *spam F1 measure (SF1)* were calculated. SR is the proportion of spam e-mails that are classified as spam, i.e. the spam e-mails that the filter manages to block. SP is the proportion of e-mails classified as spam that are truly spam.

Discarding a legitimate e-mail is of greatest concern than classifying a spam message as legitimate. This means that high SP is particularly important. In addition, cost-sensitive measures such as *weighted accuracy (WA)* and *total cost ratio (TCR)* [1] were calculated. Blocking a legitimate e-mail is considered a bigger error (λ times more costly) than non-blocking a spam e-mail. To make accuracy sensitive to this cost, WA is defined: when a legitimate e-mail is misclassified/correctly classified, this counts as λ errors/successes, respectively. The TCR compares the performance of the spam filter to a baseline classifier. The case where there is no filter (i.e. all e-mail is assumed to be legitimate) is used as baseline. Greater TCR indicates better performance; if TCR<1, it is better not to use the filter.

We also follow the proposed three cost scenarios: a) no cost considered (λ=1), e.g. flagging spam messages, b) semi-automatic scenario for moderately accurate filter (λ=9), e.g. notifying senders about blocked messages and c) completely automatic scenario for a highly accurate filter (λ=999), e.g. removing blocked messages.

For evaluation of the results we used stratified ten-fold cross validation based on the pre-defined folds of PU1 and LingSpam.

**4.2.2. Overall performance.** The results for the lemm versions of LingSpam and PU1 are given in Table 4. For comparison, the reported results of NB [1,2], kNN, stacking [13], stumps and boosted trees [3] are also included. As it can be seen LINGER-IG did extremely well and achieved perfect results on both LingSpam and PU1 (on all corpora versions, for all λ). LINGER-V did not perform as well as on the general e-mail multi-class classification, yet it compares favourably in terms of F1 with most of the other approaches and is only slightly outperformed by the tree stumps and boosted trees. TCR ratio for LINGER-V is <1 only for λ=999 on both corpora. Hence, for λ=999, it is better not to use the filter with LINGER-V. Similar results for λ=999 were reported for the other methods used for comparison.

**4.2.3. Effect of stemming and stop words.** LINGER's performance on the four different versions of LingSpam and PU1 has been evaluated, see [4]. The results show that the use of stemmer and stop wording improves the performance on LingSpam with only 0.7% and worsen it with 1% on PU1 which confirms the observations of [1].

**4.2.4. Anti-spam filter portability evaluation.** Portability of an anti-spam filter is an interesting question and an important issue in real applications. We tested the portability across corpora using LingSpam and U5Spam (PU1 cannot be used as it is encrypted), Table 5. In the first experiment we trained a filter on LingSpam and then test it on U5Spam (and vice versa), using features selected from the training corpus. The results are unsatisfactory. Typical confusion matrices are given in Table 6 a) and b). The low SP in both cases is due to the fact that many legitimate e-mails were misclassified as spam. This can be explained with the different nature of the legitimate e-mail of LingSpam (linguistics related) and U5Spam (more diverse). The features selected based on LingSpam are too specific and do not act as good predictor for non-spam e-mails of U5Spam. While in the first case the high SR indicates that the spam e-mail is mainly correctly classified, in the second case a substantial amount of spam messages (322) are misclassified as legitimate (low SR). It seems that a feature selection based on the considerably smaller U5Spam corpus performs worse and misclassifies also many spam e-mails despite their similar nature across the two corpora.

**Table 5. Portability across corpora**

| | features | train | test | A | SR | SP | SF1 |
|---|---|---|---|---|---|---|---|
| | | | Experiment 1 | | | | |
| a | Ling Spam | Ling Spam | U5 Spam | 36.8 | 99.0 | 31.4 | 47.7 |
| b | U5 Spam | U5 Spam | Ling Spam | 60.6 | 29.5 | 15.1 | 20.0 |
| | | | Experiment 2 | | | | |
| c | Ling Spam | U5 Spam | U5 Spam | 87.9 | 62.2 | 94.4 | 75.0 |
| d | U5 Spam | Ling Spam | Ling Spam | 98.8 | 94.6 | 98.1 | 96.3 |

In the second experiment the training and testing were on the same data but the feature selection was based on the other data set. The results are considerably better and indicate that even if the feature selection is not perfect, NN is able to recover by training and achieve good performance. Thus, training based on the e-mail collection of the user seems to be more important than feature selection. It is interesting to note (see Table 6 c) and d)) that the number of spam as non-spam misclassifications is higher than the number of non-spam as spam. Hence, based on our experiments we can not conclude that the anti-spam filter is portable. More extensive experiments with diverse, non topic specific corpora, are needed to determine the portability of anti-spam filters across different users.

**Table 6. Typical confusion matrices**

| | a) | | b) | |
|---|---|---|---|---|
| # assigned as | spam | not spam | spam | not spam |
| spam | 81 | 1 | 159 | 322 |
| not spam | 177 | 23 | 832 | 1580 |
| | c) | | d) | |
| # assigned as | spam | not spam | spam | not spam |
| spam | 51 | 31 | 455 | 26 |
| not spam | 3 | 1970 | 9 | 2403 |

## 5. Conclusions

We have shown that NNs can be successfully used for automated e-mail filing into mailboxes and spam mail filtering. The backpropagation-based system LINGER outperforms several other algorithms in terms of classification performance. We have explored the effects of various feature selection, weighting and normalization techniques and found that V is a better feature selection than IG in the multiclass task, while LINGER-IG obtained a perfect performance in the binary spam mail filtering. Frequency and tf-idf weighting with mailbox level normalization produced the best results in e-mail filing. More experiments are needed to determine the portability of the anti-spam filter across different users.

## References

[1] I. Androutsopoulos et al., "An Experim. Comparison of NB and Keyword-Based Anti-Spam Filtering", *ACM SIGIR,* 2000.

[2] I. Androutsopoulos et al. "Learning to Filter Spam E-mail: A Comp. of a NB and Memory-Based Approach", *PKDD, 2000.*

[3] X. Carreras and L. Marquez, "Boosting Trees for Anti-Spam Email Filtering", *4th Int. Conf. Recent Advances in NLP*, 2001.

[4] J. Clark, I. Koprinska, J. Poon, "LINGER – A Smart Personal Assistant for E-mail Classif.", ICANN/ICONIP 2003.

[5] W. Cohen, "Learning Rules that Classify E-Mail", *AAAI Symp. on Machine Learning in Inf. Access*, 1996, pp.18-25.

[6] E. Crawford, E. McCreath, J. Kay, "IEMS - The Intellient Email Sorter", *19th Int. Conf. on Machine Learning,* 2002.

[7] J. Rennie, "An Application of Machine Learning to E-Mail Filtering", *KDD-2000 Text Mining Workshop*, 2000.

[8] M. Sahami, S. Dumais, et al. "A Bayesian Approach to Filtering Junk E-Mail", *AAAI Workshop Learning for TC*, 1998.

[9] G. Sakkis et al., "Stacking Classifiers for Anti-Spam Filtering of E-mail", *6th Conf. Empir. Meth. in NLP*, 2001, 44-50.