An Adaptive Approach to Spam Filtering on a New Corpus

Ben Medlock
Cambridge University Computer Laboratory
William Gates Building
JJ Thomson Avenue
Cambridge, CB3 OFD

Abstract

Motivated by the absence of rigorous experimentation in the area of spam filtering using realistic email data, we present a newly-assembled corpus of genuine and unsolicited (spam) email, dubbed *GenSpam*, to be made publicly available. We also propose an adaptive model for semi-structured document classification based on smoothed n-gram language modelling and interpolation, and report promising results when applying the classifier to the spam filtering problem using a specifically assembled test set to be released as part of the *GenSpam* corpus.

1 Introduction

The well-documented problem of unsolicited email, or *spam*, is currently of serious and escalating concern[1]. In lieu of effective legislation curbing the dissemination of mass unsolicited email, *spam filtering*, either at the server or client level, is a popular method for addressing the problem, at least in the short-term. While various spam filters have begun to find their way onto the market, little in the way of rigorous testing has been carried out to evaluate their relative effectiveness. This is due, in part, to the lack of a realistic, heterogeneous corpus of email data containing both spam and genuine messages. In this paper, we present such a corpus, dubbed *GenSpam*[2], along with an adaptive LM-based classification model for spam filtering, or more generally semi-structured document classification, that builds on recent work in the field.

2 Related Work

Some of the first published work on statistical spam filtering was carried out by Sahami et al. (1998) using a multi-variate Bernoulli NB model; however the training and test sets are small (less than 2000 total messages), and not publicly available, thus rendering the experiments non-replicable.

Androutsopoulos et al. (2000) present results for spam filtering on the most widely-used spam filtering data set, the *LingSpam* corpus. They compare a multinomial NB classifier with a kNN variant, the results favouring NB. Carreras and Marquez (2001) build on this work, publishing improved results on the same corpus using boosting decision trees with the *AdaBoost* algorithm.

The *LingSpam* corpus consists of messages drawn from a linguistics newsgroup, and as such the genuine messages are largely homogeneous in nature (linguistic discussion) and thus non-representative of the general spam-filtering problem, where genuine messages typically represent a wide range of topics. Additionally, the corpus consists predominantly of genuine messages (2412 genuine, 481 spam) whereas in reality the balance is more often in favour of spam, and is too small to allow experimentation into the important issue of how a classifier

[1] See research by *MessageLabs* (www.messagelabs.co.uk) and *Ferris* (www.ferris.com).

[2] The corpus will be made publicly available on the web at the time of publication of this paper.

adapts as the nature of spam and/or genuine email changes over time and between different users.

Drucker et al. (1999) publish results comparing the use of SVM's with various other discriminative classification techniques on the spam filtering problem, with binary-featured SVM's and boosting decision trees performing best overall. Unfortunately the test sets they used are not publicly available.

The spam filtering problem is usually presented as an instance of a *text classification problem* on the basis that most email contains some form of identifiable textual content. In reality, the structure of email is richer than that of flat text, with meta-level features such as the fields found in MIME compliant messages. Researchers have recently acknowledged this, setting the problem in a *semi-structured document classification* framework. Bratko and Filipič (2004) take this approach on the *LingSpam* corpus, reporting a significant reduction in error rate compared with the flat text baseline. The semi-structured document classification framework is of course applicable to a wider range of problems than just spam filtering, as in (Yi and Sundaresan, 2000; Denoyer and Gallinari, 2004; Bratko and Filipič, 2004). In all these cases the NB classification model is extended to take account of the componential document structure in question. We note that the limiting *conditional independence assumption* of NB can be relaxed in a classification framework based on smoothed higher-order n-gram language models. This is also recognised by Peng and Schuurmans (2003), who report state-of-the-art results using a higher-order n-gram based LM text classifier on a number of data sets. We define a similar classification model, but extend it into an adaptive semi-structured framework by incorporating recursive structural component *interpolation*. We apply the resulting classification model to the newly assembled *GenSpam* email corpus.

3 A New Email Corpus

The need for a large-scale corpus of realistic email data is evident from the lack of rigorous experimentation and comparison in the area of spam filtering. The corpus we have assembled consists of:

- 9072 genuine messages
- 32332 spam messages

The imbalance in the number of messages is due in part to the difficulty of obtaining genuine email - persuading people to donate personal email data is a challenge. In fact, the corpus is not as unbalanced as it may appear; on the whole, spam messages tend to be significantly shorter than genuine ones, so in terms of actual content the balance is somewhat more even.

The genuine messages are sourced from fifteen friends and colleagues and represent a wide range of topics, both personal and commercial in nature. The spam messages are sourced from sections 10-29 of the *spamarchive*[3] collection, as well as a batch of spam collected over recent months.

Releasing personal, potentially confidential email data to the academic community requires an anonymisation process that protects the identities of senders and recipients, as well as those of persons, organisations, addresses etc. referenced within the email body. We have investigated various anonymisation procedures utilising NER and other statistical NLP techniques, which will be discussed in detail in a subsequent paper. Fig 1 gives an example of the *GenSpam* email representation in XML format.

```
<MESSAGE>
<FROM> net </FROM>
<TO> ac.uk </TO>
<SUBJECT>
<TEXT_NORMAL> ^ Re : Hello everybody </TEXT_NORMAL>
</SUBJECT>
<DATE> Tue, 15 Apr 2003 18:40:56 +0100 </DATE>
<CONTENT-TYPE> text/plain; charset="iso-8859-1" </CONTENT-TYPE>
<MESSAGE_BODY>
<TEXT_NORMAL>
^ Dear &NAME ,
^ I am glad to hear you 're safely back in &NAME .
^ All the best
^ &NAME
^ - On &NUM December &NUM : &NUM &NAME ( &EMAIL ) wrote :
...
</TEXT_NORMAL>
</MESSAGE_BODY>
</MESSAGE>
```

Figure 1: *GenSpam* representation

The corpus is divided as follows:

- *Training set*: 8018 genuine, 31235 spam
- *Adaptation set*: 300 genuine, 300 spam
- *Test set*: 754 genuine, 797 spam

We source the adaptation and test sets from the contents of two users inboxes, collected over a number of months, retaining both spam and genuine messages. We take this approach rather than simply

[3] http://www.spamarchive.org

extracting a test set from the corpus as a whole, so that the test set represents a real-world spam filtering instance. The 600 messages making up the adaptation set are randomly extracted from the same source as the test set, facilitating experimentation into the behaviour of the classifier given a small set of highly relevant samples and a large background corpus.

4 Classification Model

4.1 Introduction

We use the following terminology and definitions:

- *Document*: a discrete item of information (i.e. a single email message).
- *Token*: an atomic unit within a document.
- *Component*: a section of a document. A component can either be *recursive*, consisting of one or more sub-components, or *non-recursive*, consisting of a finite set of tokens. A document is itself a recursive component.
- *Field*: a non-recursive component within a document (eg. *Subject* field).
- *Class*: a well-defined (possibly infinite) set of documents.

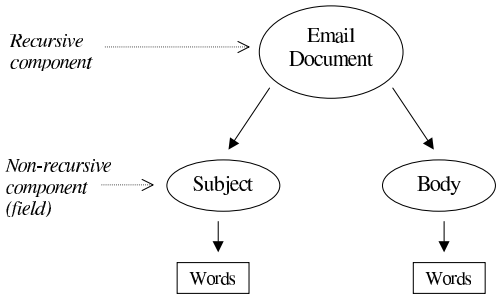

Figure 2: Example document structure

Given these definitions, a semi-structured document is a tree with nodes as recursive components and leaves as non-recursive components (see Fig. 2).

The classification model we present is an *interpolated generative model*. That is, recursive component posterior probabilities are computed as an interpolation of sub-component posteriors, while non-recursive component (field) posteriors are estimated in the traditional generative fashion. The interpolation weights are estimated under the discriminative classification function; consequently the model bears some relation to the class of *Hybrid Generative/Discriminative* classifiers, eg. (Raina et al., 2004). By incorporating smoothed higher-order n-gram language models[4], local phrasal dependencies are captured without the undesirable independence violations associated with mixing higher and lower-order n-grams in a pure Naïve Bayesian framework (Tan et al., 2002). Additionally, through the use of interpolation, we incorporate a well-studied technique for combining probabilities to exploit document structure.

Although we only consider application of the proposed classification model to the 2-class classification problem, it is in principle scalable to the more general N-class problem.

4.2 Formal Classification Model

We make the following assumptions:

1. *A document belongs to exactly one class* (though the model can be extended to the multi-class variant).
2. *Classification is carried out within a single domain, and within that domain, all documents have the same structure.*

Given a set of documents $\mathbf{D}$ and a set of classes $\mathbf{C}$, we seek to discover a set of classifications of the type $D_i \rightarrow C_j$ for $i = 1 \ldots |\mathbf{D}|$ where j ranges from $1 \ldots |\mathbf{C}|$ (given assumption 1).

By analogy to the standard generative classification model, the interpolated generative decision rule chooses the class with the highest interpolated posterior probability for the document in question:

$$Decide(D_i \rightarrow C_j) \quad \text{where } j = \arg\max_k [P(C_k|D_i)] \qquad (1)$$

The posterior probability of a recursive component (such as a document) is calculated as a

[4] We use n-grams for efficiency and simplicity, though more advanced LM technology could be investigated.

weighted linear interpolation of the posterior probabilities of its N sub-components:

$$P(C_j|D_i) = \sum_{n=1}^{N} \lambda_n [P(C_j^n|D_i^n)] \quad (2)$$

where

C_j^n is the nth sub-component of class C_j

D_i^n is the nth sub-component of doc D_i

λ_n is the nth sub-component weight

The formula is applied recursively to each of the recursive sub-components. An *interpolation scheme* is used to find the optimal values for the λ's (4.5).

The non-recursive component (field) posterior probability is expanded using *Bayes Rule*:

$$P(C_j^n|D_i^n) = \frac{P(C_j^n) \bullet P(D_i^n|C_j^n)}{P(D_i^n)} \quad (3)$$

C_j^n represents a specific field within class C_j, and D_i^n the corresponding field within the document. Under the structure uniformity assumption (2), these fields are necessarily equivalent.

$P(C_j^n)$ is the *prior probability* for the field in question. We take all field priors within a given class to be equal to the class prior, i.e. $P(C_j)$.

The denominator, $P(D_i^n)$, is constant with respect to class and thus often ignored in Bayesian classification models; however, valid interpolation requires true probabilities; thus we retain the denominator. Dividing by $P(D_i^n)$ can also be seen as normalising for unequal field lengths, i.e. scaling the class-conditional probability of a field (dependent on the length of the field by virtue of the fact that probabilistic intersection equates to multiplication) by a value constant with respect to class but multiplicatively proportional to the length of the field.

$P(D_i^n)$ can be expanded to

$$\sum_{k=1}^{|\mathbf{C}|} P(C_k^n) \bullet P(D_i^n|C_k^n)$$

which is the sum over all classes of the prior times the class-conditional likelihood for the given field.

$P(D_i^n|C_j^n)$ is the *language model probability* of the field D_i^n given C_j^n. In other words, it is the likelihood that the LM chosen to model field C_j^n generated the sequence of tokens comprising D_i^n. Bearing in mind the definition of a field, we represent the LM probability by the following formula:

$$P(D_i^n|C_j^n) = P(D_i^n = \{t_i \dots t_K\}|LM_{C_j^n}) \quad (4)$$

For our experiments we use n-*gram* LM's. The n-gram model is based on the assumption that the existence of a token at a given position in a sequence is dependent only on the previous $n-1$ tokens. Thus the n-gram LM probability for a K-length token sequence can be defined (with allowances for the initial boundary cases) as

$$P_N(t_1, \dots, t_K) = \prod_{i=1}^{K} P(t_i|t_{i-N+1}, \dots, t_{i-1})$$

The formula is specialised for $n = 1, 2, 3 \dots$

4.3 LM Construction

We adopt the basic formalisation for higher-order n-gram smoothing introduced by Katz (1987). This approach has been shown to perform well across a number of recognised data sets (Chen and Goodman, 1996), and is the most widely used smoothing technique in fields such as speech recognition. In the bigram case, the formula is as follows:

$$P(t_j|t_i) = \begin{cases} d(f(t_i,t_j))\frac{f(t_i,t_j)}{f(t_i)} & \text{if } f(t_i,t_j) > C \\ \alpha(t_i)P(t_J) & otherwise \end{cases}$$

where

f is the frequency-count function

d is the discounting function

α is the back-off weight

C is the n-gram cutoff point

For higher-order n-grams the same principles are applied to form a *back-off chain* from higher to lower-order models. The n-gram cut-off point, C, is the threshold below which the observed number of occurrences is too low to draw reliable statistics from. The discounting function, d, is used to deduct some of the probability mass from observed events, making it available to unobserved events. We use mixed *Good-Turing/Linear* discounting for our experiments, a widely-used strategy, though arguably weaker than some more recent variants, eg. (Orlitsky et al., 2003). The discounted probability mass is

spread over lower-order distributions with the back-off weight insuring conformance to the probability model. A small probability must also be assigned to events that remain unobserved at the end of the back-off chain. We can use this to model the likelihood of encountering unknown tokens given a particular class. This can be useful in modelling problems such as spam filtering (see results, 7.1).

4.4 Adaptivity

A realistic classification model for spam filtering must be able to accomodate the evolving nature of spam and the variation in genuine email between different users. In light of this, we extend our classification model to incorporate a top-level interpolation of posteriors from both a *static* and *dynamic* element.

The decision rule (1) is thus expanded to:

$$Decide(D_i \rightarrow C_j) \text{ where}$$
$$j = \arg\max_k [\lambda_s P_s(C_k|D_i) + \lambda_d P_d(C_k|D_i)] \quad (5)$$

This is equivalent to adding a new binary-branching recursive top-level node to the document structure, one branch representing the static information, the other the dynamic (as in Fig. 3). The subscripts s and d denote the static and dynamic elements respectively, and the probabilities are interpolated such that $\lambda_s + \lambda_d = 1$ (see 4.5). In practice, the static element represents a classification model built from a large background corpus, while the dynamic element represents a much smaller model that is regularly retrained on newly-classified data.

4.5 Interpolation

The purpose of an interpolation scheme is to optimise the weights of two or more interpolated components with respect to their performance on a given data set, under a specified objective function, eg. (Jelinek and Mercer, 1980). In our case, a component is the posterior probability for a particular recursive or non-recursive structural component. We choose the classification function itself (under a suitable evaluation metric) as the objective function, which has the advantage of precisely reflecting the nature of the problem. On the negative side, the classification function is *non-differentiable*, thus optimality of the interpolation weights cannot be derived

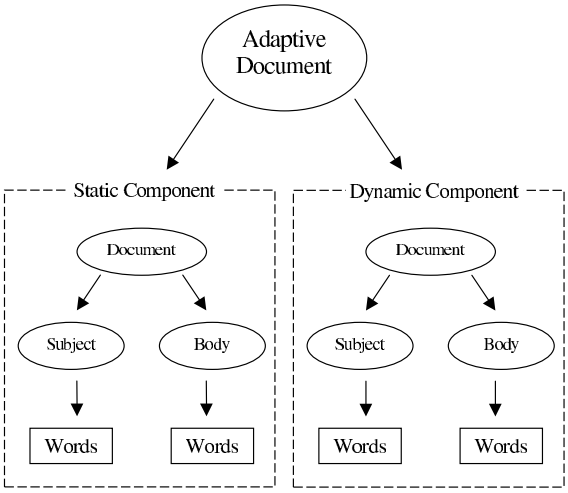

Figure 3: Example adaptive doc structure

using differential-motivated optimisation techniques which converge to optimality in a reasonably efficient manner. Rather, we must use an approximation algorithm to achieve near-optimality. In our experiments we only interpolate two components (see 5) so a simple hill-climbing algorithm suffices. However if a greater number of fields were available, a more complex algorithm would need to be investigated.

To maintain efficiency, we estimate interpolation weights in a bottom-up fashion, propagating upwards through the structural tree rather than iteratively re-estimating throughout the whole structure.

5 Experimental Method

Our experiments consisted of two phases:

- *Phase 1*: divide the GenSpam *Training Set* into two disjoint sections, one for training the classifier and the other for measuring performance, allowing us to experiment with basic model features such as term-capitalisation and n-gram cutoff.
- *Phase 2*: train the classifier using the *Training Set* and *Adaptivity Set*, classify the *Test Set* and record results.

To train the classifier we extract a number of messages from the training data for each class to estimate the interpolation weights, and use the rest to estimate LM parameters.

Our experiments make use of only two email fields - *Subject* and *Body*. These are of primary in-

terest in terms of content, though other fields such as *From*, *To*, *Date* etc. are also of potential use. This is an avenue of further research.

6 Evaluation Measures

The 2-class classification task is often evaluated using the *accuracy* measure, which combines precision and recall for both classes and represents a percentage of correctly classified documents. We report both *accuracy*, and *recall* for both classes separately, defined in the usual manner:

$$accuracy = \frac{TP}{T} \qquad recall(C) = \frac{TP^C}{T^C}$$

where

TP = number of true positives

T = total number of documents

Assessing the performance of the classifier on spam and genuine email separately is vital in the area of spam filtering, where high recall of genuine messages is of utmost importance.

7 Results and Analysis

We present results and analysis for our classifier using unigram and bigram language models. Experiments with higher order n-gram models resulted in degraded performance, perhaps due to:

- insufficient quantity of data
- noisy and inconsistent data
- over-fitting

7.1 Phase 1 Experiments

During the *phase 1* experiments (see 5) we varied certain features of the language models and observed results on held-back sections of the training data to determine the better-performing configurations. We also ran experiments to assess the advantage of modelling fields separately. The results led us to draw a number of conclusions:

- In both the unigram and bigram cases, modelling each field with a specific language model is more effective than using a combined LM.
- Ignoring tokens that occur only once in the training data provides a marginal increase in performance. This is intuitive, as such tokens are unlikely to well-represent the language being modelled, especially when dealing with noisy data. Ignoring single occurrences is also beneficial for reducing LM size, especially in the spam case where noise is deliberately introduced to confuse filters.
- Case normalisation has a detrimental effect in the unigram case, presumably because the positive effect in terms of reducing sparsity is outweighed by the negative effect of losing certain intended semantic distinctions. However, in the bigram instance case normalisation has a marginally positive effect on performance, due, we suspect, to the extra contextual information in the bigram model making up for the lost semantic information in certain cases.
- Intuitively, we might expect spam to contain more unknown words than genuine email, due to the additional lexical noise. Thus, by assigning a slightly higher probability to unknown words in the SPAM class, we would expect some performance gain. This was bourne out in the bigram case, but not in the unigram case (perhaps due to the prior removal of single word occurences). We suspect that a more principled, data-driven method for assigning unknown event probabilities would evidence an improvement in performance across the board.
- The discrepancy in LM size between different classes as a result of unbalanced training data can lead to classification errors because parameters in larger LMs receive proportionally less of the overall probability mass. This is especially noticeable in higher-order LMs where the potential feature space is much larger. One method for countering this is to raise the n-gram cutoff point (see 4.3) for the larger class. We call this technique *LM balancing*, and found it to have a positive effect on performance, especially in the bigram case.

7.2 Phase 2 Experiments

Having discussed general model features established in *phase 1*, we now present results for our experiments on the unseen test data using the *Training* and

LM type	Recall GENUINE	Recall SPAM	Accuracy
unigram	98.94	96.11	97.49
bigram	98.94	98.75	98.84

Table 1: Results for adaptive LM classifier on *GenSpam* corpus.

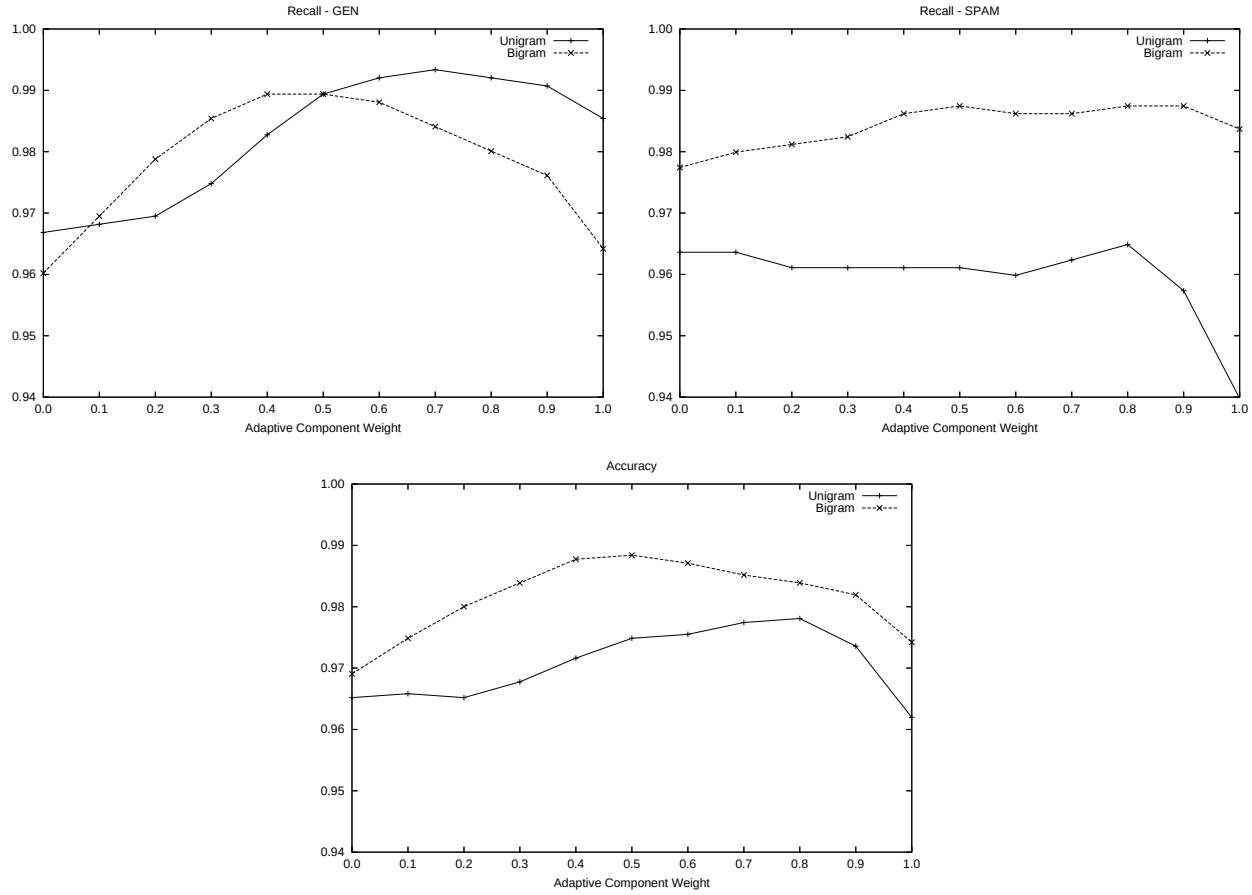

Figure 4: Classifier *Recall* (GEN and SPAM) and *Accuracy* under adaptive weight interpolation

Adaptivity sets. Given a small amount of data for adaptivity (300 messages per class in our case), we are unlikely to be able to estimate from it reliable interpolation weights for the adaptivity decision rule (5). In practice, therefore, we might choose to set these manually. Given that the distribution of interpolation weights can itself be interpreted as a probability distribution with each weight representing the probability that a particular component contains relevant information, we choose the distribution that is most uncertain, governed by the principle of *maximum entropy*. Without any prior knowledge about the optimal weight distribution, this equates to balancing the weights across all components (two in this case).

Table 1 shows the performance of the classifier on the *Test Set* from the *phase 2* experiments, with balanced static/dynamic weights as described above. The bigram classifier cannot improve on the unigram performance in terms of genuine email recall, though it reduces the error rate for spam significantly by around two-thirds. Most of the misclassified genuine messages were either corporate circulars or 'impersonal' automated responses, and thus hard to distinguish as genuine. We suspect that a short *whitelist* of trusted corporations would eliminate almost all such mistakes.

Figure 4 shows how classification performance

varies as the adaptive interpolation weight distribution ranges from static to dynamic. Primarily this highlights the fact that a combination of static and dynamic models performs better than either of the models separately. We can also see that the unigram classifier is quite skewed towards genuine messages in the dynamic case, making its behaviour less predictable as the weights vary. Finally, it is interesting to note that the recall distribution for genuine messages is significantly more rounded than the corresponding distribution for spam (especially in the bigram case), suggesting that combining static and dynamic information about genuine email is particularly important. This may reflect the fact that genuine email varies more between users than spam.

8 Conclusions

We have presented a new corpus of genuine and unsolicited email, *GenSpam*, which we believe represents a more realistic test-bed for the spam filtering problem than anything currently available. Obtaining spam is relatively easy, thus the corpus can be updated to include the latest unsolicted email without a great deal of effort. We believe that the anonymised genuine email content represents a significant contribution in itself, and may be useful for a wider range of NLP tasks than just spam filtering.

We have also presented an adaptive classification model for semi-structured documents that builds on similar work in the semi-structured and hybrid generative/discriminative classification fields. We achieve quite promising results when applying the classifier to the spam filtering task on the new corpus, demonstrating the benefit of using smoothed n-gram language models and exploiting document structure. We suspect that carefully-constructed higher-order language models could prove especially beneficial in addressing NLP problems in which the text follows stricter grammatical patterns and contains less noise than in the email domain.

References

I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, and P. Stamatopoulos. 2000. Learning to filter spam email: A comparison of a naive bayesian and a memorybased approach. *Workshop on Machine Learning and Textual Information Access*, 4.

A. Bratko and B. Filipič. 2004. Exploiting structural information in semi-structured document classification. In *Proc. 13th International Electrotechnical and Computer Science Conference, ERK'2004*.

X. Carreras and L. Marquez. 2001. Boosting trees for anti-spam email filtering. *Proceedings of RANLP2001*, pages 58–64.

S.F. Chen and J.T. Goodman. 1996. An empirical study of smoothing techniques for language modeling. *Proceedings of the 34th Annual Meeting of the ACL*.

Ludovic Denoyer and Patrick Gallinari. 2004. Bayesian network model for semi-structured document classification. *Inf. Process. Manage.*, 40(5):807–827.

H. Drucker, D. Wu, and V.N. Vapnik. 1999. Support vector machines for spam categorization. *IEEE Trans. On Neural Networks*, 10(5):1048–1054.

F. Jelinek and R.L. Mercer. 1980. Interpolated estimation of markov source parameters from sparse data. *Proceedings of the Workshop on Pattern Recognition in Practice*.

E. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. ASSP*, 35(3).

Alon Orlitsky, Narayana P. Santhanam, and Junan Zhang. 2003. Always good turing: Asymptotically optimal probability estimation. In *FOCS*, pages 179–188.

F. Peng and D. Schuurmans. 2003. Combining naive bayes and n-gram language models for text classification.

R. Raina, Y. Shen, A. Ng, and A. McCallum. 2004. Classfication with hybrid generative/discriminative models. *NIPS 16, 2004*.

M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. 1998. A bayesian approach to filtering junk e-mail. *Learning for Text Categorization - Papers from the AAAI Workshop*, pages 55–62.

Chade-Meng Tan, Yuan-Fang Wang, and Chan-Do Lee. 2002. The use of bigrams to enhance text categorization. *Inf. Process. Manage.*, 38(4):529–546.

Jeonghee Yi and Neel Sundaresan. 2000. A classifier for semi-structured documents. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 340–344. ACM Press.