An Adaptive, Semi-Structured Language Model Approach to Spam Filtering on a New Corpus

Ben Medlock
Cambridge University Computer Laboratory
William Gates Building
JJ Thomson Avenue
Cambridge, CB3 OFD
bwm23@cam.ac.uk

ABSTRACT

Motivated by current efforts to construct more realistic spam filtering experimental corpora, we present a newly assembled, publicly available corpus of genuine and unsolicited (spam) email, dubbed *GenSpam*. We also propose an adaptive model for semi-structured document classification based on language model component interpolation. We compare this with a number of alternative classification models, and report promising results on the spam filtering task 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, there is a lack of rigorous evaluation of their relative effectiveness in realistic settings. As a result, there is an ongoing research effort to construct representative, heterogeneous experimental corpora for the spam filtering task. In this paper, we present a sizeable, heterogeneous corpus of personal email data to add to the spam filtering research arsenal, dubbed *GenSpam*[2]. We also present and evaluate an adaptive LM-based classification model for spam filtering, or more generally semi-structured document classification.

2. RELATED WORK

Some of the first published work on statistical spam filtering was carried out by Sahami et al. [19] using a multi-variate Bernoulli NB model. However, the training and test sets were small (less than 2000 total messages), and not publicly available, thus rendering the experiments non-replicable.

Androutsopoulos et al. [1] present results for spam filtering on the *LingSpam* corpus. They compare a multinomial NB classifier with a kNN variant, the results favouring NB. Carreras and Marquez [4] build on this work, publishing improved results on the same corpus using boosting decision trees with the *AdaBoost* algorithm.

[1]See research by *MessageLabs* (www.messagelabs.co.uk) and *Ferris* (www.ferris.com).
[2]Available from http://www.cl.cam.ac.uk/users/bwm23/

CEAS 2006 - Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California USA

Drucker et al. [8] 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 *LingSpam* corpus [1] is currently the most widely-used spam filtering dataset. It 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 *adapts* as the nature of spam and/or genuine email changes over time and between different users.

In light of the inadequacy of *LingSpam* and the paucity of publicly available, realistic email data for experimental spam filtering, various efforts have recently been made to construct more realistic spam filtering experimental corpora, most notably the TREC 2005 spam track corpus and a variant of the Enron corpus[14, 6]. Such efforts will provide opportunity for a new generation of more realistic spam filtering experiments.

The spam filtering problem has traditionally been presented as an instance of a *text categorization* 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 [2] 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 [22, 7, 2]. 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 [17], 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 corpus we have assembled consists of:

- 9072 genuine messages (∼154k tokens)
- 32332 spam messages (∼281k tokens)

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. On the whole though, spam messages tend to be significantly shorter than genuine ones, so in terms of total content volume, the balance is somewhat more even, as can be seen from the token count.

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 by the author and compatriots. The messages are from roughly the same time period (predominantly 2002-2003), with the genuine messages more widely time distributed, while the spam messages represent the more recent instances in circulation at the point the corpus was constructed.

Relevant information is extracted from the raw email data and marked up in XML. Retained fields include: *Date*, *From*, *To*, *Subject*, *Content-Type* and *Body*. Non-text attachments are discarded, though the meta-level structure is preserved. If an email consists of multiple sections, these are represented by <PART> tags with a *type* attribute specifying the section type.

Standard and embedded text is identified and marked up in XML with <TEXT_NORMAL> and <TEXT_EMBEDDED> tags respectively. Embedded text is recognised via the '>' marker, with up to four nested levels of embedding.

Releasing personal, potentially confidential email data to the academic community requires an anonymisation procedure to protect the identities of senders and recipients, as well as those of persons, organisations, addresses etc. referenced within the email body. We use the RASP [3] part-of-speech tagger as well as finite-state techniques to identify and anonymise proper names, numbers, email addresses and URLs. The following tokens are used in place of their respective references:

- &NAME (proper names)
- &CHAR (individual characters)
- &NUM (numbers)
- &EMAIL (email addresses)
- &URL (internet urls)

The *From* and *To* fields contain the email addresses of the sender and recipient(s) respectively. We retain only top level domain (TLD) information from each field. For US-based sites, the TLD is defined as the string of characters trailing the final dot, i.e. 'com' in 'joe@yahoo.com'. For non-US sites, it is defined as the final 2-char country code, along with the preceding domain type specification, i.e. 'ac.uk' in 'joe@dur.ac.uk' or 'co.uk' in 'freecomputers@flnet.co.uk'. This allows for potentially useful analysis of high-level sending and receiving domains, without any individual identity traceability.

After applying the automatic anonymisation procedures, all of the genuine messages were manually examined by the author and a colleague to anonymise remaining sensitive references. This took a significant amount of time, but resulted in a consensus that the data was sufficiently anonymous to be publicly released.

It is to be expected that spam filtering with anonymised data is somewhat more challenging than it would be otherwise, as potentially useful information is necessarily lost. However, our experiments with both anonymised and unanonymisd versions of *GenSpam* suggest that using unanonymised data results in only marginally better performance (around 0.003 improvement in recall), and that the difference between classification performance on anonymised and unanonymised data is not sufficient to cause concern about misrepresenting the task.

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

```
<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

Figure 1 gives an example of the *GenSpam* email representation in XML format. 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 (Nov 2002–June 2003), retaining both spam and genuine messages. We take this approach rather than simply 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.
- *Class*: a well-defined (possibly infinite) set of documents.

A semi-structured document is a singly-rooted tree (see Fig. 2). Non-leaf nodes represent structural document sections and leaf nodes represent content bearing sections.

The classification model we present is an *interpolated generative model*. That is, non-leaf (structural) node posterior probabilities are computed as an interpolation of sub-node posteriors, while leaf (content) node posteriors are estimated in the traditional generative fashion. The interpolation weights are optimised under the discriminative classification function; consequently the model bears some relation to the class of *hybrid generative/discriminative* classifiers,

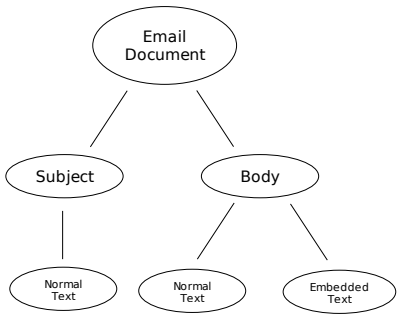

Figure 2: Example of semi-structured document

[18]. 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 [20]. Additionally, through the use of interpolation, we incorporate an efficient, 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 readily scales 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.* This is clearly appropriate for spam filtering, though it is in principle quite simple to extend the model to allow documents to belong to multiple classes.
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).

We use the standard Bayes decision rule to choose the class with the highest posterior probability for a given document:

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

The posterior probability of a non-leaf document node is calculated as a weighted linear interpolation of the posteriors of its N sub-nodes:

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

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

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

An *interpolation scheme* is used to determine values for the λ's (see subsection 4.5).

Leaf-node posteriors are computed via *Bayes Rule*:

$$P(C_j^n|D_i^n) = \frac{P(C_j^n) \cdot P(D_i^n|C_j^n)}{P(D_i^n)} \qquad (3)$$

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

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

The document node prior, $P(D_i^n)$, is constant with respect to class and thus often ignored in Bayesian classification models; however, valid interpolation requires true probabilities; thus we retain it. This carries the additional benefit of normalising for imbalanced field lengths. For instance, the amount of text in the *subject* field is usually significantly less than in the *body* field and therefore the class conditional likelihood for the *body* field will be disproportionately lower. However, scaling the class-conditional likelihood of each by the document node prior, which is multiplicatively proportional to the length of the field, counteracts the imbalance.

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

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

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.

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

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

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

4.3 LM Construction

We adopt the basic formalisation for higher-order n-gram smoothing introduced by Katz [13]. This approach has been shown to perform well across a number of recognised data sets [5], and is widely used in mature language modelling fields such as speech recognition. In the bigram case, the formula is as follows:

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

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 introduce a simple new discounting scheme called *confidence discounting* which performs well in our experiments, and is highly efficient to compute:

$$d(r) = \frac{r}{R}\omega \quad (4)$$

R is the number of distinct n-gram frequencies and ω represents a ceiling on discount mass. For our experiments we use $\omega = \frac{n_3}{T}$ where n_i is the number of n-grams occuring i times in the training data, and T is the total number of words.

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 8.1).

4.4 Adaptivity

A realistic classification model for spam filtering should take account of the fact that spam evolves over time. It should also account for the fact that each individual spam filtering instance will have its own characteristics, due to the variation in email usage, but at the same time much evidence about the nature of spam versus genuine email will be common across all (or at least most) instances. In light of this we extend our model to incorporate both a *static* and *dynamic* element. The static element represents evidence contributed by LMs trained on a large background corpus, while the dynamic element represents smaller, instance-specific evidence from LMs that are regularly retrained as new data is accrued.

The decision rule (1) is expanded to:

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

The subscripts s and d denote the static and dynamic elements, which are separate but identically structured estimates, derived from the static and dynamic LMs respectively. The modified decision rule can be interpreted as adding a binary-branching recursive top-level node to the document structure with both branches structurally identical but using different sets of LMs (Fig. 3). The adaptive decision rule can thus be rewritten as:

$$Decide(D_i \rightarrow C_j) \quad \text{where } j = \arg\max_k [P(C_k^a|D_i^a)] \quad (6)$$

with the superscript a denoting use of the adaptive structure.

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 [11]. In our case, a component is represented by the posterior probability for a particular tree node. 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 estimated with derivative-based 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 6) so a simple hill-climbing algorithm suffices. However if a greater number of fields were utilised, 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.

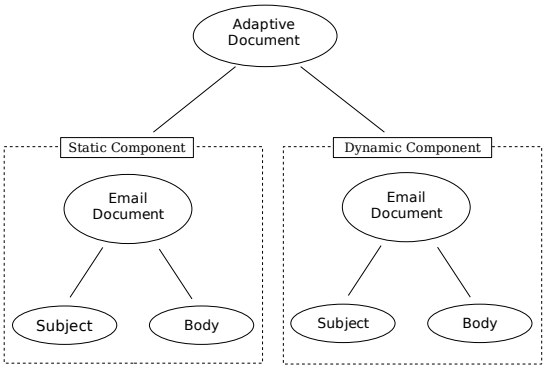

Figure 3: Example of adaptive document structure

5. COMPARISON

To provide a frame of reference for the performance of our classification model, we also present results for the multinomial naïve Bayes (MNB), support vector machine (SVM) and Bayesian logistic regression (BLR) classification models on the *GenSpam* corpus.

5.1 MNB

Multinomial naïve Bayes is commonly used as a baseline in classifier comparisons, due to its ease of implementation and computational efficiency. It tends to underperform more sophisticated techniques as it often suffers from overfitting and excessive bias.

5.2 SVM

Support vector machines [21] have yielded state of the art results on various classification tasks over recent years. They are robust to overfitting and reasonably efficient especially for small to medium sized datasets.

5.3 BLR

Bayesian logistic regression is a technique that falls into the family of regression techniques for classification. A prior over feature weights is used to prefer sparse classification models and thus avoid overfitting and increase efficiency. Such a model was shown to perform competitively with the state of the art on various TC datasets in [10].

5.4 Implementation

We will henceforth refer to the classification model presented in this paper as ILM (Interpolated Language Model), which we have implemented in perl. We have also implemented our own version of MNB following the standard model [16], and use Joachims' SVM^{light} [12][5], reporting results for the best performing linear

[5]http://svmlight.joachims.org/

kernel. We use the open source implementation of Bayesian logistic regression, BBR (Bayesian Binary Regression) provided by Genkin et. al [**?**][6].

6. EXPERIMENTAL METHOD

We use held-back sections of the training data to tune the ILM hyperparameters: unseen term estimates, n-gram cutoff and interpolation weights, as well as the regularization parameter in SVM^{light}. MNB doesn't have any hyperparameters, and BBR has an inbuilt '–autosearch' parameter to optimise the prior variance via 10-fold cross validation. We then evalute each of the classifiers on the test data in three sets of experiments, using as training data:

1. Just the *Training* data
2. Just the *Adaptation* data
3. A combination of both

6.1 Data

Our experiments make use of only two email fields - *Subject* and *Body*. These are of primary interest in terms of content, though other fields such as *From*, *To*, *Date* etc. are also of potential use. This is an avenue for further research.

We pre-process the corpus by removing punctuation and tokens that exceed 15 characters in length. We do not carry out stopword removal as it had a significantly detrimental effect on performance, especially in the SVM case. This is presumably due to the fact that stopword usage differs between spam and genuine email, and exemplifies the disparity between spam filtering and traditional text categorization.

The ILM and MNB classifiers do not require scaling or normalisation of the data. For SVM and BLR, we construct *tfc*-weighted (normalised *tf∗idf*) input vectors.

7. EVALUATION MEASURES

The binary classification task is often evaluated using the *accuracy* measure, which represents the proportion of documents correctly classified. We also report *recall* for each class separately, defined in the usual manner:

$$accuracy = \frac{\text{TP}}{\text{T}} \quad recall(c) = \frac{\text{TP}^c}{\text{T}^c}$$

where TP is the number of true positives, T the total number of documents, TP^c the number of true positives in class c and T^c the number of documents in c.

Assessing the recall performance of the classifier on spam and genuine email separately is important in the area of spam filtering, where high recall of genuine messages is of utmost importance. This imbalance in the nature of the task necessitates evaluation schemes that recognise the asymmetric cost of misclassification.

8. RESULTS AND ANALYSIS

We present results for the various classifiers on the *GenSpam* corpus under symmetric and asymmetric evaluation schemes.

8.1 Hyperparameter Tuning

We varied certain features of the ILM classifier and observed results on held-back sections of the training data to determine the better-performing configurations. The results led us to draw a number of conclusions:

- We use only unigram and bigram language models, as higher order n-gram models degrade performance due to excessive sparsity and over-fitting.
- Intuitively, we might expect spam to contain more unknown words than genuine email, due to the additional lexical noise. The LM unseen event probability can be used to model this phenomenon. We optimise unseen event probabilities empirically from held out sections of the training data, and arrive at the following values:

Unigram	GEN	1×10^{-8}
	SPAM	1.2×10^{-8}
Bigram	GEN	1×10^{-8}
	SPAM	1×10^{-7}

- 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 for bigram LMs. Hence, we use C=1 for GEN and C=2 for SPAM in the body field LMs generated from the *Training* dataset, and C=1 for all other LMs.

After tuning on held-back sections of the training data, we use the linear kernel and choose the value C=1 for the regularization parameter in SVM^{light}. We use a Gaussian prior distribution and the '–autosearch' parameter in BBR to optimise the prior variance via 10-fold cross validation.

8.2 Symmetric Classification

Table 1 displays the performance of the classifiers on the *Test* dataset under the standard symmetric evaluation scheme. For the *Combined* results we merge the *Training* and *Adaptation* sets in the case of MNB, SVM and BLR, and combine them by the adaptive decision rule (5) for ILM.

The amount of adaptation data is too small to reliably estimate interpolation weights for the adaptive decision rule. In practice, therefore, we would set these manually. Given that the distribution of interpolation weights can 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 component weights.

As expected, MNB is highly efficient, but performs somewhat worse than the best performing model in each category.

The SVM classifier performs well when trained only on the *Adaptation* data, but relatively poorly when trained on the *Training* data. This is because the *Adaptation* set has certain properties that suit the SVM model: the distribution of the training data matches that of the test data (they are both roughly balanced), the data is linearly separable and the diminutive number of training samples allows the wide-margin effect to have a significant impact. Conversely, the *Training* set does not particularly suit the SVM model: the training distribution does not match the test distribution and the training data is unbalanced and non linearly separable. It has been shown empirically that the SVM model chooses a suboptimal decision boundary in the presence of divergence between the training and test distributions [9] and this is supported by our results.

[6]http://www.stat.rutgers.edu/ madigan/BBR/

Training Data	Classifier	GEN recall	SPAM recall	accuracy
Training	MNB	0.9589	0.9322	0.9452
	SVM	0.9005	0.9837	0.9433
	BLR	0.8926	0.9862	0.9407
	ILM Unigram	0.9496	0.9674	0.9587
	ILM Bigram	0.9735	0.9636	**0.9684**
Adaptation	MNB	0.9682	0.9335	0.9504
	SVM	0.9854	0.9724	**0.9787**
	BLR	0.9642	0.9737	0.9691
	ILM Unigram	0.9775	0.9373	0.9568
	ILM Bigram	0.9682	0.9649	0.9665
Combined	MNB	0.9629	0.9297	0.9458
	SVM	0.9310	0.9887	0.9607
	BLR	0.9244	0.9887	0.9574
	ILM Unigram	0.9907	0.9674	0.9787
	ILM Bigram	0.9854	0.9737	**0.9794**

Table 1: *GenSpam Test* set results (best results for each dataset in bold)

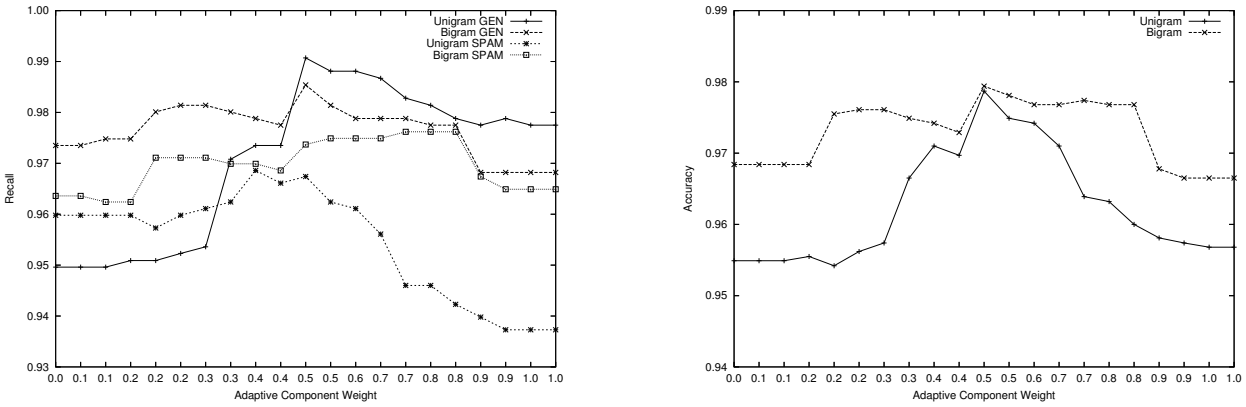

Figure 4: ILM *recall* (GEN and SPAM) and *accuracy* under adaptive weight interpolation

The results for BLR are quite similar to SVM in these experiments, though slightly inferior. Estimation of the prior variance by cross validation does improve performance, though it dramatically increases training time.

The ILM classifier performs competitively across the board, and particularly when the adaptive decision rule is used. Figure 4 plots classification performance as a function of the adaptive interpolation component weight, so that $x = 0$ represents only the *Training* models and $x = 1.0$ represents only the *Adaptation* models. For both unigram and bigram LMs, the ILM classifier benefits from the combined estimates; however the benefit is most significant in the unigram case. It is interesting to note that both classifiers reach a performance peak at the point where the static and dynamic weights are balanced, i.e. when there is an equal contribution from both models.

8.3 Asymmetric Classification

While the symmetric results are informative, they do not present a realistic view of the spam filtering problem, in which the correct classification of genuine mail is of much greater import than the occasional misclassification of spam. There are a number of ways to evaluate spam filters in the presence of asymmetric misclassification cost; we will use a scenario in which a predefined recall threshold for genuine mail must be reached by the classifier. We set this threshold at recall=0.995 i.e. we allow, on average, no more than one genuine message in every 200 to be misclassified.

We control the bias in the MNB, SVM and BLR classifers by adjusting the decision threshold at a granularity of 0.001. The SVM model can also be biased by increasing the misclassification cost for a given class (-j option in SVM^{light}); however we found that even for highly skewed values of this parameter (ratio of 1/1000) the requisite genuine mail recall threshold remained unreached.

The language modelling aspect of the ILM classifier allows various ways of biasing of the model in favour of a given class. We control the bias by reducing the unseen term estimate for the SPAM body LMs until the genuine threshold is reached.

Table 2 displays the results of biasing the models to reach the genuine mail recall threshold. The full ILM model, trained on the combined static and dynamic data, significantly outperforms any of the other classifiers, with the accuracy of the bigram variant actually increasing as the genuine recall threshold is reached. This suggests that the ILM classifier is well suited to the spam filtering task.

Figure 5 plots the receiver operator characteristic (ROC) curves for each of the best-performing classifier configurations (BLR and SVM – adaptation data; ILM – combined data). This provides fur-

Training Data	Classifier	GEN recall	SPAM recall	accuracy
Training	MNB	0.9960	0.1556	0.5642
	SVM	0.9960	0.7064	0.8472
	BLR	0.9960	0.8105	0.9007
	ILM Unigram	0.9960	0.7340	0.8614
	ILM Bigram	0.9960	0.8331	**0.9123**
Adaptation	MNB	0.9960	0.4090	0.6944
	SVM	0.9960	0.9147	0.9491
	BLR	0.9960	0.9097	**0.9542**
	ILM Unigram	0.9960	0.8269	0.9091
	ILM Bigram	0.9960	0.8934	0.9433
Combined	MNB	0.9960	0.4103	0.6950
	SVM	0.9960	0.8808	0.9368
	BLR	0.9960	0.9021	0.9478
	ILM Unigram	0.9960	0.9573	0.9761
	ILM Bigram	0.9960	0.9674	**0.9813**

Table 2: Asymmetric results (best results for each dataset in bold)

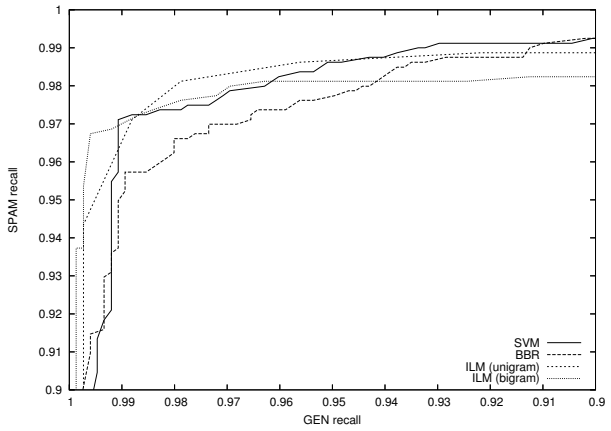

Figure 5: Classification ROC curves

ther evidence of the effectiveness of biasing the language models by adjusting unseen word estimates, as opposed to biasing the decision boundary. Both the unigram and bigram ILM classifiers, when trained on the combined data, are able to maintain high genuine recall values without sacrificing overall accuracy.

9. DISCUSSION

Interpolating LM-based structural components provides a natural way to efficiently combine estimates from different distributions. With n-gram LMs, the classifer uses efficient maximum likelihood estimation and hence has a training and classification time complexity roughly linear in the input size. However, an approach such as the one presented in this study has its drawbacks, as it requires estimates for a significant number of hyperparameters. These must be derived either empirically or by potentially expensive cross validation. The parametricity of the ILM model also makes it potentially sensitive to changes in the nature of the problem domain, a relevant issue when dealing with the ever-changing nature of spam, and email in general. An obvious line of future research is to investigate methods for estimating the ILM hyperparameters both robustly and efficiently.

Bearing in mind the success of the ILM classifier at combining evidence from distinct training distributions, it would be interesting to investigate analagous techniques for discriminative models such as the SVM and BLR. A possible starting point would be to examine the effects of combining judgements from separate SVM or BLR models trained on distinct data. An interpolative method could potentially be used in this setting, which would not harm the tractability of the base classifier, though it would introduce new hyperparameters. A further avenue of research is to investigate alternative methods of biasing the discriminative classifiers to improve their asymmetric performance. One possible approach would be to investigate recent research into utilising uneven margins in the SVM model [15]. This technique has shown some promise when dealing with skewed training data, though it has not been examined in the context of handling asymmetric classification costs.

10. CONCLUSIONS

We have presented a new corpus of genuine and unsolicited email, *GenSpam*, which we hope will aid in providing opportunity for more realistic spam filtering experiments and ultimately enhance efforts to build more effective real-world spam filters. Obtaining spam is relatively easy, and a potentially important task for the future is to update the corpus with more recent spam, improving its relevance. 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 efficient, adaptive classification model for semi-structured documents that extends similar work in the semi-structured and hybrid generative/discriminative classification fields. We demonstrate that our classifier is effective at combining evidence from distinct training distributions (an important attribute for adaptive classification), and experiments on *GenSpam* suggest that the model is well suited to spam filtering, maintaining high levels of genuine recall without loss of overall accuracy.

11. ACKNOWLEDGEMENTS

This work was funded by the University of Cambridge Millennium Scholarship programme. Many thanks to Prof. Ted Briscoe for his extensive input both in terms of validating the email anonymisation process and guiding the work from a theoretical perspective. Thanks also to Dr. Mark Gales and Dr. Thomas Hain for their helpful comments in the early stages.

12. REFERENCES

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

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

[3] T. Briscoe and J. Carroll. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1499–1504, 2002.

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

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

[6] G. Cormack and T. Lynam. Spam corpus creation for trec. In *Proceedings of Second Conference on Email and Anti-Spam CEAS 2005*, 2005.

[7] L. Denoyer and P. Gallinari. Bayesian network model for semi-structured document classification. *Inf. Process. Manage.*, 40(5):807–827, 2004.

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

[9] G. Forman and I. Cohen. Learning from little: Comparison of classifiers given little training. In *PKDD*, pages 161–172, 2004.

[10] A. Genkin, D. D. Lewis, and D. Madigan. Large-scale bayesian logistic regression for text categorization. 2005.

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

[12] T. Joachims. Making large-scale support vector machine learning practical. In A. S. B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1999.

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

[14] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *ECML*, pages 217–226, 2004.

[15] Y. Li, K. Bontcheva, and H. Cunningham. Using uneven margins SVM and perceptron for information extraction. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 72–79, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[16] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

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

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

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

[20] C.-M. Tan, Y.-F. Wang, and C.-D. Lee. The use of bigrams to enhance text categorization. *Inf. Process. Manage.*, 38(4):529–546, 2002.

[21] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[22] J. Yi and N. Sundaresan. 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, 2000.