How Important is Contextual Understanding in Spam Filtering?

Andrew Georgiou
King's College
apjg4@cam.ac.uk

1 INTRODUCTION

Spam Filtering is a critical task for most internet users, the ability to prevent attention theft or the more harmful email attacks such as phishing or malware now lays at the foundation of modern email services. Current state-of-the-art approaches to spam filtering is not algorithm or technique related but rests mostly on the quality and quantity of input data. Previous spam corpora lacked in both areas respectively.
GenSpam introduced in (Medlock, 2006) attempted to improve on the previous state-of-the-art corpora *LingSpam* (Sakkis et al., 2003) by increasing the quantity of spam emails, improving the quality with manually analysed messages and adding structure to the email documents within the dataset.
In this paper we attempt to replicate the experimental methods defined in the original paper, as well performing an extrinsic evaluation of Medlock's model introduced for the task of spam filtering named *ILM (Interpolated Language Model)* against the transformer architecture and FastText(Joulin et al., 2016). We intend to identify if deep contextual understanding really matters when it comes to detecting spam. Giving a detailed comparison of these models performance and a discussion of the techniques used and experiments performed before coming to a conclusion of our findings.

2 RELATED WORK

The most common early approaches to the task of spam filtering use multi-variate Bernoulli distribution where we assign a conditional probability where the prior is the word *W* and the posterior is the class *C*:

$$P(W|C = genuine) \; and \; P(W|C = spam)$$

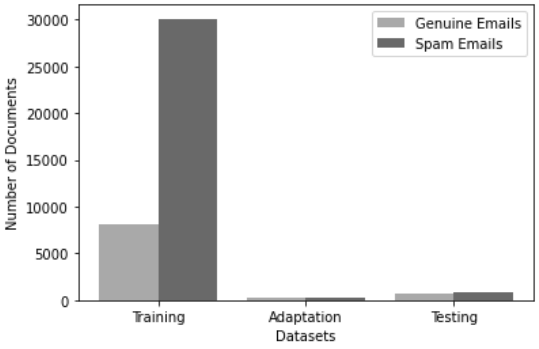

Figure 1: Bar Chart showing document count of divided representation corpora.

Then for predicting a given document we instead intend to predict a class given a message *M*:

$$P(C = genuine|M) \; and \; P(C = spam|M)$$

This approach relies on the fact that language differs between spam and genuine email documents, language and especially spam evolves over time to evade detection, attacks such as Bayesian poisoning can escape detection through including a litany of words often found in genuine emails to increase the occurrences of classes most commonly occurring in the genuine class.
Support Vector Machines (SVM) were first proposed in 1992 by Vapnik et al. (Cristianini and Ricci, 2008) When applied to spam filtering SVM's have shown to be a state-of-the-art algorithm in machine learning and still remain to perform competitively in the task. A paper by Islam et al. (Islam et al., 2005) performed an analysis of four machine learning models including SVM on four important datasets for spam filtering, one of these datasets being the LingSpam Dataset. Their results showed SVM significantly outperformed Linear Regression on all datasets except the TREC 2007 Spam Corpus. The results from their paper can be argued to perform worse than the expected SVM perfor-

mance if the regularisation parameter C is tuned correctly on all datasets as work by Liu et al. (Dada et al., 2019) showed.
Deep Learning Techniques regularly improve state-of-the-art on many downstream tasks, spam-detection is harder to track current state-of-the-art techniques since data and often model architecture and performance is private for sensitivity of user emails in genuine documents and prevention of malicious spam generators from finding attack vectors to circumvent these spam filtering systems. Therefore we can only look at publicly available spam filtering techniques on open-source datasets for our analysis. One such analysis of deep learning techniques by AbdulNabi et al. (AbdulNabi and Yaseen, 2021) achieved a F1-Measure of 0.9866 using the transformers model BERT (Vaswani et al., 2017) trained on a combination of open-source corpora showing it performed the best over a selection of other deep learning models therefore influencing our model selection in this paper.

3 CORPUS ANALYSIS

Introduced by Ben Medlock in 2006, GenSpam attempts to address the problems with spam filtering corpora available at the time. It attempted to due this by introducing a more realistic dataset by extracting data from genuine email accounts to give a true representation of the task. Previous datasets focused entirely on the subject and text body of email documents and GenSpam was intended to introduce the idea that the structure of the email is much richer than that of flat text and therefore includes additional data such as *Date, From, To, Content-Type* in an XML format.
The corpus consists of 9072 genuine messages (154k tokens) and 32,332 spam messages (281k tokens). The representation data then available to us are split up accordingly:

- Training set: 8018 genuine, 31,235 spam.
- Adaptation set: 300 genuine, 300 spam.
- Test set: 754 genuine, 797 spam.

The corpus is therefore separated with a 96/4 split between training and testing set, which is quite uncommon in evaluation as we often use a 80/20 split. This is explained in the paper as the adaption and test dataset are sourced from the contents of two users inboxes collected over a number of months in order to retain the true representative value of the corpus. The data is extracted raw and presented in an XML format with tags such as <TEXT_NORMAL> and <CONTENT-TYPE>, this presents the first of a few issues with pre-processing the dataset as the format was not friendly with standalone python or xml processing libraries. Some of the recognised issues with the dataset during experimentation and development are listed below:

- Unrepresentative training/test split meant less precision when evaluating results.
- XML format is difficult to utilise as <TEXT_EMBEDDING> tag creates unnecessary tree structure for message body that is very slow to process using XML libraries.
- Certain character's used in encoding XML completely break standard XML processing library from processing ElementTree structure even when altering encoding type on input.
- Imbalanced training set for true representation of email occurrences are not representative of test set occurrence count.

These problems could be addressed quite easily though, such as supplying a JSON version of the dataset which would allow for easier parsing as a dictionary or splitting the training set representatively for development and fine-tuning on the adaptation set for testing.

4 CLASSIFICATION MODELS

4.1 Support Vector Machine

For this paper we utilise Support Vector Machines (SVM), which have proved to have state-of-the-art performance on a variety of tasks. In Medlock's original paper he compared the performance of SVM against their ILM and it proved competitive against their model specifically built for the task of spam filtering making it an important base metric for our paper. It also enables us to attempt to replicate Medlock's results and experimental methods. For our model We use the SkLearn library to import their SVM component, the original paper listed some hyper-parameters such as a using C=1 for the regularisation parameter although we did experiments on altering this parameter which shown to have great effect. We used tf-idf input vectors with positive results between 1000-2000 features.

Model Architecture	Genuine Recall	Spam Recall	Accuracy
Baseline Models - From Original Paper			
SVM	0.9310	0.9887	0.9607
ILM Unigram	0.9907	0.9674	0.9787
ILM Bigram	0.9854	0.9737	0.9794
Our Evaluated Models			
SVM	0.9376	0.9723	0.9555
FastText Unigram	0.9549	0.9861	0.9709
FastText Bigram	0.9706	0.9874	0.9787
BERT	**0.9873**	0.9887	**0.9877**
RoBERTa	0.9801	**0.9900**	0.9850

Table 1: Comparison of models performance against Medlock's original baselines, Genuine Recall, Spam Recall and Accuracy metrics utilised.

4.2 FastText

FastText represents a strong baseline for word embeddings, it is really quick to train a FastText model in comparison to a deep learning model and allows for easy to prototype character n-grams and n-gram cutoff points. which were included in the higher-order n-gram smoothing defined in Medlock's original paper for construction of their ILM model. When developing the FastText model we followed from the evaluation measures outlined in the original paper, experimenting with both unigram and bigram models, we utilised different n-gram cutoff points for and found similar results with positive effects on the bigram model.

4.3 Bidirectional Encoder Representations from Transformers

BERT is a transformer model and was chosen for this analysis over a simpler bi-directional LSTM model as it allows for true bi-directionality and can allow us to analyse how important the context is in spam classification. The model was imported and fine-tuned through the SimpleTransformers library which is flexible library built on top of HuggingFace Transformers that enables importing and training and parameter tuning large language models very quickly, since a large focus of this project is on creating a base line performance and evaluating the experimental methods of the corpus it makes sense to reduce time wasted in importing each language model and its architecture into PyTorch. Since we do not train the model from scratch but fine-tune we have control over the following hyper-parameters:

- Early Stopping Measures
- Training/Evaluation Batch Sizing
- Epsilon Hyper-parameter in AdamOptimiser
- Learning Rate

For this paper we did not deviate from the default parameters defined by the SimpleTranformers library as they proved to perform the best on our validation set.

4.4 Robustly Optimised BERT Pre-Training Approach

This model is a Transformer model similar to BERT in that it optimises the BERT architecture in order to reduce time during pre-training. This model removed the Next Sentence Prediction (NSP) objective of BERT as well as training with larger batch sizing and longer sequences of upto 8k over BERT's 512 max sequence length. We chose this model for exactly these features as many of the sequences in GenSpam exceed the 512 max sequence length once tokenised and single words are broken down into multiple tokens. Similarly to BERT we also fine-tuned this model through SimpleTransformers where the only parameters that required tuning for both models is handling early stopping.

Hyperparameter Tuning For tuning our hyperparameters we initially looked at the original paper with intention to follow any of their experimental methods for the SVM model to give a fair comparison between our work, we chose to also remove punctuation and tokens that exceeded 15 characters in length, as well as setting both a minimum and maximum frequency on tokens. The FastText model made use of a gridsearch to optimise hyperparameters and found that with the current size of our dataset 25 epochs, minimum count of 2 and a learning rate of 0.1 performs the best on our subset

of the training set we used for evaluation. Both BERT and RoBERTa used early stopping using the Matthews Correlation Coefficient as the chosen stopping metric with 3 epochs each and weight biases set in the ratio of genuine to spam documents.

5 EVALUATION

Metrics We intend to keep the evaluation metrics the same as the original GenSpam paper as it allows us to directly performance. The original paper utilises accuracy and recall, they report recall for each class separately which they utilise when evaluating symmetric vs asymmetric performance. In this paper we do not make use of such comparisons which is discussed later although understanding the recall of each class allows us to understand how valuable our given system is if we wish to prioritise either genuine email detection or spam email detection as a result.

6 RESULTS AND ANALYSIS

We will present our results in a similar fashion to the original paper, with exception of differentiating symmetric and asymmetric classification through altering the decision threshold values as we found better results through weighting our classes appropriately during training to maximise genuine document recall but also discovered issues with using a asymmetric system in a real world setting.

6.1 Maximise Accuracy or Recall?

In calculating results of spam increase in relation to genuine email filtering we actually chose to entirely disagree with Ben Medlock's premise that a desired genuine email recall should be thresholded at 0.9906 as most baseline systems would become unusable at that trade-off. We believe maximising the F_1 measure produces a better trade off for an end user of spam filtering system. This analysis comes from Medlock's results, when we set the recall of genuine spam emails to a threshold of 0.9960 through fine-tuning our decision thresholds until we reach this point in our recall of our genuine email class we find that spam recall as an effect reduces in relation to the confidence of the model which is also visible in the original paper. Analysing the SVM model which performed quite highly on the symmetric evaluation we can see spam recall dropped from 0.9887 to 0.8808 on the recall metric for spam emails. This reduction means that for every 100 spam emails we receive, we would misclassify 14 of them as genuine emails. We can calculate this by rearranging the equation for calculating recall where x is equal to the number of spam emails we might receive and y is the number of spam emails we misclassify, the recall value we can set at the reduced spam email recall rate of 0.8808 which happens as a result of thresholding our genuine recall rate at 0.9906 as specified by Medlock in his paper which is r.

$$\frac{x}{r} - x = y$$

This is reasonable for Medlock in their implementation as they have built an perfectly balanced test dataset for both spam and genuine emails where having poor spam recall does not significantly ruin accuracy in a balanced system. In reality in 2010 85% of the world's daily email traffic was spam (SecureList, 2010) with 120 billion spam emails sent per day.

What this means is that for every 100 genuine emails we receive we also receive on average 567 spam emails, if we thresholded our recall as in the asymmetric evaluation for our SVM model we would have a spam recall rate of 0.8808, therefore we would allow 75 spam emails into our inbox for every 100 genuine emails at a rate of 3 spam for every 4 genuine. This would in most cases render a users inbox redundant and vulnerable to many phishing and other email attacks that are found in spam documents. The actual best method to reduce this and maximise seperation of both spam and genuine email and this is done through maximising accuracy not genuine document recall.

6.2 Parameter Tuning and Model Training

As part of our training we followed the experimental methods for our models as in the original paper, we also experimented with pre-processing our data in the same way and include an evaluation of the comparison between the original methods and our method were we actually choose not to remove stop words from our data and experiment with lowering and removing thresholds as these features negatively impacted our model's performance. The results of the features on the SVM model and how it impacted our performance are included in Table 2 where each feature was evaluated separately. The only 2 features which improved performance over no filtering were punctuation removal and using unigram and bigram mixed tfidf vectors. For our input data we maintained to only train on subject

SVM Model Features			
Features	**Genuine Recall**	**Spam Recall**	**Accuracy**
No Filtering	0.9336	0.9636	0.9490
PR	0.9270	**0.9723**	0.9503
SF	0.9336	0.9611	0.9477
C=1	0.9244	0.9686	0.9471
C=3	0.9204	0.9661	0.9439
Unigram Only	0.9204	0.9661	0.9439
Bigram Only	0.8687	0.9673	0.9194
Unigram and Bigram	0.9363	0.9673	0.9522
Unigram/Bigram with PR	**0.9376**	**0.9723**	**0.9555**

Table 2: Evaluation of features on Test Set. PR = Punctuation Removal, SF = Stopword Filtering, C = Regularisation Parameter, For all other features when not listed uses sklearn SVC default parameters and 8000 features as found to perform well on isolated train set.

and body email fields and tuned our hyperparameters on a subsection of the train set, we then trained the models on a combination of the train and adaptation set. All of the results of our evaluations are performed on the test dataset.

6.3 Symmetric Classification

After training on our models on the shuffled training data with the shuffled adaptation dataset appended to the end of the set the performance of both Transformer models exceeded original expectations achieving results much higher than the baseline performance of the original paper. One thing we noted is that genuine recall is lower across all our models against the ILM unigram model but performed much greater in Spam Recall, this could be due to the weights needing further biasing for the genuine document class and further experiments could be useful to experiment with this although one key piece of information found during fine-tuning is that BERT and RoBERTa require significant time fine-tune on the whole training set. RoBERTa training for 10 epochs took on average 5-6 hours which meant it could not be iterated as often as expected. With the same parameters BERT outperformed all other models in overall accuracy producing state-of-the-art performance for the task of Spam Detection on the GenSpam corpus. The results for SVM are very similar to the original paper's results meaning we are quite confident in our replication of the hyperparameter and data processing techniques employed therefore validating the performance of our transformer models over the originally proposed ILM model. One such factor that could have aided this performance increase is that BERT and RoBERTa train on entire text sequences and employ true bi-directionality. Since we see the performance improvements between unigram and bigram this is a much larger extension of a simple n-gram model in that we encode every word in relation to all other in the sequence.

7 DISCUSSION

Utilising models with greater understanding of word's in a given sequence such as word embeddings or transformers proves to be highly effective in the task of Spam Detection, from manually analysing both genuine and spam documents it is clear that what differentiates the two exceeds beyond just the vocabulary but the sentence structure and formation of coherent English. Spam tends to struggle as the majority of spam emails originate from countries where English is not a first-language and therefore lacks the semantic structure found in genuine emails from within the academic community.

In the original paper one of the discussions is around the significant number of hyperparameters that require tuning for the ILM model as it is sensitive to changes where as the opposite is true in training transformer models as they are pre-trained and suffers when stop words and low frequency tokens are removed thereby removing the ability to tune hyper-parameters which may allow an equally large transformer model to perform better at the task if trained from scratch albeit.

An interesting way to expand upon this idea would be to utilise the semi-structured document format of the corpus in a better way that would allow unused XML fields such as *Date, From, To and Content-Type* to be embedded into the sequence and tuned

on a similarly large language model.
If we could have improved this project we would have assigned more resources to the training and testing time of both language models, as they required significant computational power and time to train. We would have also have liked to develop our own LSTM model in order to tackle the task of utilising the semi-structured nature of the GenSpam corpora which has yet been utilised in its full form. This could have allowed us to investigate the performance in relation to the weights set to each of the XML fields.

8 CONCLUSIONS

We have presented a detailed comparison and evaluation of the GenSpan corpus as well as producing our own models for evaluation against the baseline ILM model proposed by Medlock in the original GenSpam paper. We experimented and trained 3 new competitive models of which 2 exceed the baseline performance of the original ILM Bigram model in terms of Accuracy at the task of classifying genuine and spam emails. We believe we have also discussed some of the problems with the original paper and opened the doorway to a wide range of language models that may be further trained specifically for the task of spam detection. We believe we have answered the question of if context matters in spam filtering as we have shown with our results that sentence structure and context plays a huge role as language models once again demonstrate state-of-the-art performance.

References

Isra'a AbdulNabi and Qussai Yaseen. 2021. Spam email detection using deep learning techniques. *Procedia Computer Science*, 184:853–858. The 12th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 4th International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

Nello Cristianini and Elisa Ricci. 2008. *Support Vector Machines*, pages 928–932. Springer US, Boston, MA.

Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, and Opeyemi Emmanuel Ajibuwa. 2019. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6):e01802.

Md Rafiqul Islam, Morshed Chowdhury, and Wanlei Zhou. 2005. An innovative spam filtering model based on support vector machine. pages 348–353.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Ben Medlock. 2006. An adaptive approach to spam filtering on a new corpus.

Georgios Sakkis, Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Costantine Spyropoulos, and Panagiotis Stamatopoulos. 2003. A memory-based approach to anti-spam filtering. *Inf. Retr.*, 6.

SecureList. 2010. SecureList spam in the third quarter of 2010.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.