Email Spam Classification using LSTM and Sentiment Analysis

{basantelhussein, mariam_fawzy, daliawk}@aucegypt.edu
Basant Abdelaal, Mariam Fawzy, Dalia Elnagar
CSCE Department, AUC The American University in Cairo

ABSTRACT

Spam email is unsolicited and unwanted junk email sent out to random people for commercial purposes. The growing volume of spam emails raised the urge of solving this problem. This paper proposed a model for a combined dataset. After balancing, cleaning, and preprocessing the data, different models, such as CNN and LSTM, are used to classify the spam emails. To improve the accuracy, several features were added to both models; one of the features was the polarity of the email which was obtained through sentiment analysis. Sentiment analysis is the use of natural language processing to extract subjective information from the data. After trying different architectures and manipulating their hyper parameters, we found that the best model is the combination of LSTM and sentiment analysis. This model got an accuracy of 97.4%, which beats the baseline.

INTRODUCTION

With the increased usage of the Internet for social and professional networking, online communication became an essential part of our daily life. E-Mails represent one of the most used mediums for formal communication in institutions, businesses, and universities due to its quickness and reliability. Due to its usefulness and spreading, spam emails were rapidly rising as well. Spam emails are considered a major threat in daily online communication. Spam emails account for generating around 57% of the total email traffic per year in 2020 [1]. This has resulted in the attacking the privacy of many users through, spreading of offensive material like pornographic content, unsolicited messages in the form of advertising and promotional materials and moreover, led to more financial strain and increased requirement of storage.

Spam Detection is an important research topic in the area of Natural Language Processing (NLP). Most of the work was carried out using traditional machine learning classifiers like SVM, Naïve Bayes, Decision trees, etc. In this paper, we will implement Deep Neural Network classifiers which are mainly Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM). In addition, attempts to extract features and integration with sentiment analysis model are experimented with.

Datasets

We used two open-source datasets from Kaggle. To solve unbalance, a new balanced dataset is created. The dataset is created by merging the spam emails from both of the datasets, and then randomly selecting emails for the ham class, with ensuring that there are no duplicate records.

	Spam	Ham	Total
Kaggle Spam Filter dataset	1368	4327	5695
Kaggle Spam Mails dataset	1462	3531	4993
Our Balanced Dataset	2830	3258	6088

Table 1 – Summary of datasets

Feature Analysis

In an attempt to extract relevant features for the spam-ham problem so that we can use non-sequential models to further solve the classification problem with higher accuracy, experiments and analysis were carried out on the training dataset to identify and explore those features.

1) Website Presence:

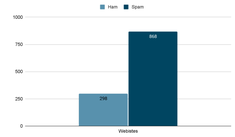


Fig 1. – Analysis results of website presence feature

The analysis was done by comparing the number of spam emails vs. ham emails containing websites.

2) Exclamation Count:

```
[ ] df["count_!"] = df['text'].str.count('!')

[ ] print(df.loc[(df['spam'] == 1)]["count_!"].mean())
    2.028975265017668

[ ] print(df.loc[(df['spam'] == 0)]["count_!"].mean())
    0.3968692449355432
```

Observation: the mean of exclaimation marks in spam emails is higher than in ham

Fig 2. – Analysis results of exclamation count feature

The analysis was done by calculating the arithmetic mean of the exclamation marks count in spam emails vs. ham emails.

3) Sentiment Analysis:

According to the literature, we found a close relation between detecting spam and sentiment analysis of the email, where past works exploited this as a feature in basic ML models.

Text Blob pretrained sentiment analysis model was used to combine its results as features to our model.

Data Preprocessing

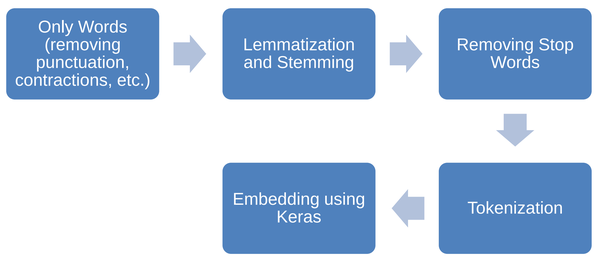


Diagram 1– Summary of Data Pre-processing Stages

BaseLine Model

The model implemented by Isra'a, et al. is used as our baseline model. The used dataset in this work is the same Spam Filter Dataset from Kaggle. The final results of the model was achieving 96.43% on the testing data.

The model used in this work was mainly using BiLSTM. Fig. 3 is a summary of the model used in this BaseLine model.

Fig 3 –BaseLine Model

Methodology

We decided to try both CNN models and LSTM models as they are both common solutions in the literature.

Since sentiment analysis has been proven to improve spam detection in basic machine learning models, we will integrate sentiment analysis in the CNN and LSTM models and record its results.

Moreover, we intend to extract more features to be added as inputs by analyzing the available datasets.

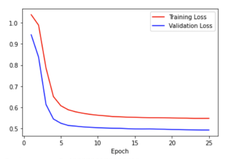
EXPERIMENTAL RESULTS

Fine Tuning Hyper Parameters:

- Loss Function
- Optimizer
- Embedding Depth
- Number of filters
- Number of convolutional / LSTM layers

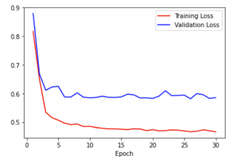
Phase 1(CNN & LSTM separately)

CNN:



Test score: 0.492070317683716
Test accuracy: 0.9629005193710327

LSTM:

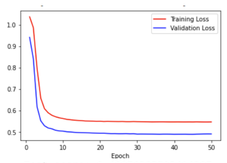


Test score: 0.5858830809593201
Test accuracy: 0.9696458578109741

Graph 1, 2 – Phase 1 Results

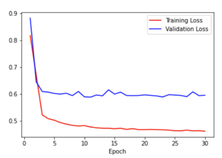
Phase 2(Adding Sentiment Analysis)

CNN:



Test score: 0.489763855934143O7
Test accuracy: 0.9629005193710327

LSTM:



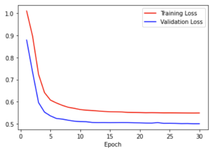
Test score: 0.594494640827179
Test accuracy: 0.9739933013916016

Graph 3, 4 – Phase 2 Results

Phase 3 (Adding Features websites - exclamation):

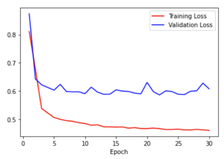
First Exclamation Count Feature:

CNN:



Test score: 0.500859797000885
Test accuracy: 0.9537426233291626

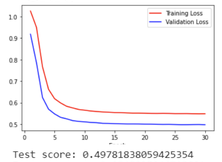
LSTM:



Test score: 0.6080095767974854
Test accuracy: 0.9714045524597168

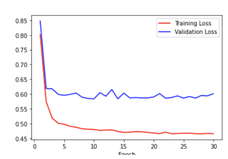
Second Website Presence Feature:

CNN:



Test score: 0.49781838059425354
Test accuracy: 0.9545837044715881

LSTM:



Test score: 0.6011539697647095
Test accuracy: 0.9722455739974976

Graph 5, 6, 7, 8 – Phase 3 Results

	LSTM	CNN
No extra features	96.9%	96.3%
Sentiment	97.4%	96.3%
Website Feature	97.2%	95.4%
Exclamation Count Feature	97.1%	95.3%
BaseLine BiLSTM	96.43%	

Table 2 – Summary of Results

CONCLUSION

CNN is not suitable for this problem as it overfits in the early epochs and it does not show any progress by adding extra features. Thus, the best architecture for training this combined data set is LSTM using the following hyper parameters in addition to adding the sentiment feature: In conclusion, the best model for training this data set is:

Loss Function: poisson

Optimizer: Adamax

Activation: softplus

Embedding Depth: 36

LSTM Layer Depth: 64

Dropout: 0.4

FUTURE WORK

People who are willing to work on the same problem in the future are recommended to do the following:

- Search for a bigger dataset or combine different datasets
- Make combinations of different models such as LSTM and CNN or add several features to one of them.
- Use Bert for the word embedding(state of art)

ACKNOWLEDGEMENTS

This work was supported by Dr. Mohamed Moustafa, AUC as well as AUC labs.