Spam Message Classification

Authors: Congkai Sun, Jiawei Tong, Huajuan Chen

Introduction and Motivations:

Spam emails are a common problem that affects individuals and organizations. These emails are usually sent to several recipients at once by hackers, and they may contain phishing scams that attempt to get you to click on a link or divulge personal details such as banking information, credit cards, address, and even ssn.

Therefore, the problem we are trying to solve is to create a classification model to distinguish between malicious emails and legitimate emails. Spam email classification is an active area of research in NLP. This report provides a potential solution for a spam text classification model based on machine learning techniques.

Common Solutions for Spam Classification:

There are several NLP techniques that people use to classify spam emails. These include Bag-of-Words, Term Frequency-Inverse Document Frequency (TF-IDF), Word Embeddings, and Sentiment Analysis. These NLP techniques can be used alone or in combination to develop effective spam email classification models.

Methods

1. **Data Sample**
   The dataset consists of 5572 rows, with 4825 rows representing legitimate messages, accounting for 86.59% of the dataset, and 474 rows representing spam messages, accounting for 13.41% of the dataset.
   The data was downloaded from Kaggle: spam-text-message-classification

2. **Text Preprocessing**
   As part of our effort to maintain clean text information, we remove punctuation, URLs, and stopwords while also utilizing steaming to standardize words to their base stem. We also create two numerical features: Total Words and Count Distinct Words. Total Words represents the number of words in each text and the Count Distinct Words represents the number of distinct words in each text. These two features have a clear and specific word range for spam text. So, adding these two variables as input features can improve the performance of the model.

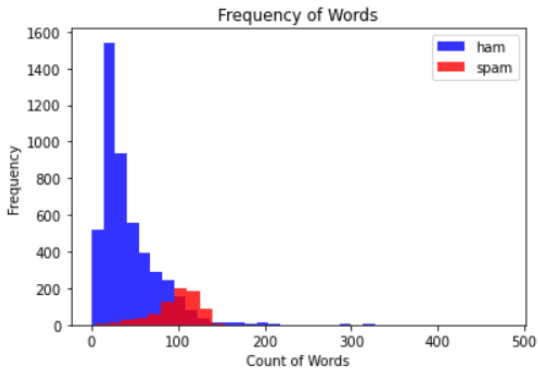


Figure 1: Distribution Plot of Total Words

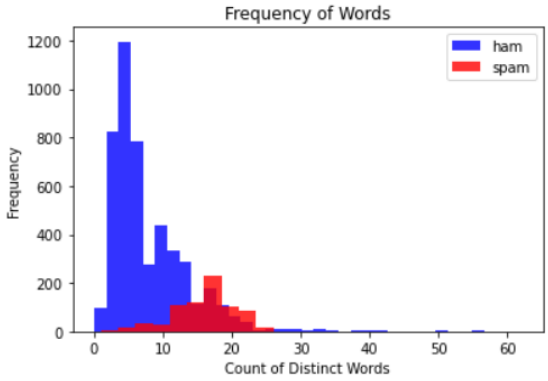


Figure 2: Distribution Plot of Count Distinct Words

Word cloud is used to visualize the most frequently occurring words in both spam and legitimate text. The Word cloud is a graphical representation of text data where the size of each word indicates its frequency or importance within the dataset. By generating separate word clouds for spam and legitimate text, we can easily identify and compare the most common terms and patterns in each category. This visual aid not only helps us understand the underlying characteristics of spam and legitimate emails, but it also guides our feature engineering process by highlighting potential keywords and phrases that could serve as informative features for our spam text classification model.

Figure 3: Common Words in Spam Text

Figure 4: Common Words in Legitimate Text

Furthermore, we employ different text vectorization and word embedding techniques, Latent Dirichlet Allocation (LDA), Term Frequency-Inverse Document Frequency (TF-IDF), and Doc2Vec to convert text to vectors.

LDA is a topic modeling technique that aims to discover hidden topics within a collection of documents by analyzing the co-occurrence patterns of words. By representing text as a mixture of topics, LDA allows us to capture the underlying thematic structure of the text data, which can be useful for distinguishing between spam and legitimate messages.

TF-IDF, on the other hand, is a widely-used technique that measures the importance of a term within a document relative to its frequency in the entire document collection. By emphasizing terms that are unique to specific message and downweighting common terms, TF-IDF generates a high-dimensional vector representation that captures the most informative aspects of the text for classification purposes.

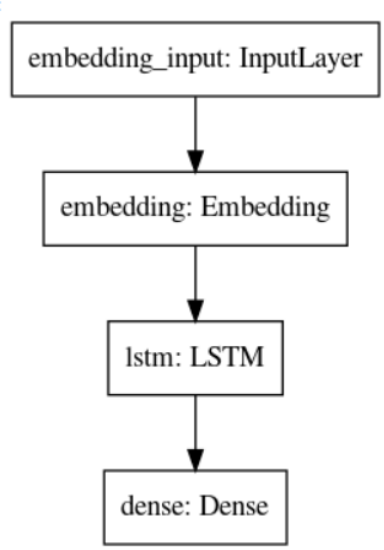
Lastly, Doc2Vec is an unsupervised neural network-based algorithm that learns to generate dense, fixed-size vector representations for variable-length text inputs. By considering the context in which words appear, Doc2Vec is capable of capturing semantic relationships between words and documents, resulting in a more expressive and meaningful representation for email content.

We also incorporate total_words and distinct_words and combine them with LDA, Tf-Idf, and Doc2Vec as input features to improve the performance of the model. By comparing and combining the strengths of LDA, TF-IDF, and Doc2Vec in this project, we aim to develop a robust and accurate spam text classification model that leverages the best aspects of each text vectorization technique.

3. **Model Implementation**

In this project, we utilize logistic regression as one of the modeling techniques to classify messages into spam and legitimate categories. Logistic regression is a powerful and widely-used statistical method for binary classification problems, making it well-suited for our task. It works by modeling the probability of an email being spam or legitimate as a function of its features, which in our case, are the numerical representations obtained from the text vectorization techniques (LDA and Tf-Idf). The logistic regression model estimates the weights for each feature such that the probability of correct classification is maximized.

In this project, we also explore the use of Long Short-Term Memory (LSTM) networks in conjunction with Doc2Vec embeddings for spam email classification. LSTM networks are a type of recurrent neural network (RNN) designed to handle sequential data and capture long-range dependencies within the text. By capturing the semantic relationships between words and documents, Doc2Vec produces expressive and meaningful representations for message content. The structure of LSTM model is shown below:



4. **Model Evaluation**
   The model is evaluated by precision, recall, f1-score and confusion matrix to compare to determine the best-performing models. Also, we will evaluate and compare the model using ROC curve. The model is adjusted to minimize false positives (legitimate text classified as spam) and false negatives (spam text classified as legitimate).This approach will enable us to make informed decisions about which models are most suitable for our purposes.

Results

1. **Model Input:**
   The input is the text that was being preprocessed and split to training and testing sets. 80% of the data will be used for training and 20% of the data will be used for testing. The input features are splited to five groups in order to compare the performance of each model. Four of them are applied in Logistic Regression and one group is applied to LSTM. The input features shown below:

   - Logistic Regression:
     1. LDA Vectors
     2. LDA Vectors + Numerical Features (Total_Words and Distinct_Words)
     3. Tf-Idf Vectors
     4. Tf-Idf Vectors + Numerical Features (Total_Words and Distinct_Words)
   - LSTM:
     1. Doc2Vec Vectors + Numerical Features (Total_Words and Distinct_Words)

2. **Model Output:**

   Classification Report: The classfication report showing the precision, recall, F1-score, and support for both the spam and ham classes as following:

	Precision	Recall	F1-Score	Suppoort
0	0.93	0.98	0.95	966
1	0.80	0.50	0.62	149
accuracy			0.92	1115
Macro avg	0.86	0.74	0.79	1115
Weighted avg	0.91	0.92	0.91	1115

Table 1: The classification report of logistic regression with LDA Vectors

	Precision	Recall	F1-Score	Suppoort
0	0.95	0.98	0.96	966
1	0.80	0.66	0.72	149
accuracy			0.93	1115
Macro avg	0.88	0.82	0.84	1115
Weighted avg	0.93	0.93	0.93	1115

Table 2: The classification report of logistic regression with LDA Vectors and numerical features

	Precision	Recall	F1-Score	Suppoort
0	0.95	1.00	0.97	966
1	1.00	0.65	0.79	149
accuracy			0.95	1115
Macro avg	0.97	0.83	0.88	1115
Weighted avg	0.96	0.95	0.95	1115

Table 3: The classification report of logistic regression with Tf-Idf Vectors

	Precision	Recall	F1-Score	Suppoort
0	0.97	0.99	0.98	966

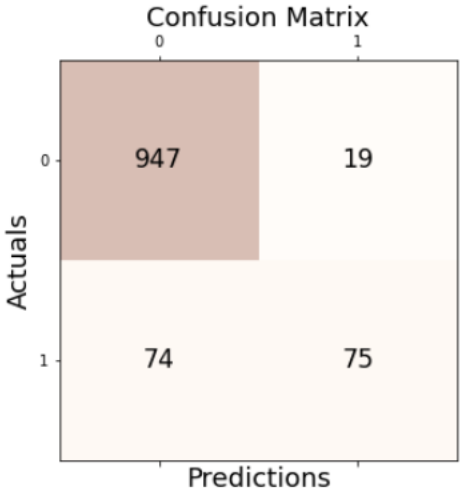
1	0.89	0.79	0.84	149
accuracy			0.96	1115
Macro avg	0.93	0.89	0.91	1115
Weighted avg	0.96	0.96	0.96	1115

Table 4: The classification report of logistic regression with Tf-Idf Vectors and numerical features

	Precision	Recall	F1-Score	Suppoort
0	0.98	0.99	0.99	966
1	0.93	0.88	0.90	149
accuracy			0.97	1115
Macro avg	0.96	0.93	0.94	1115
Weighted avg	0.97	0.97	0.97	1115

Table 5: The classification report of LSTM with doc2vec

Confusion Matrix: The confusion matrix showing the number of true positives, false positives, true negatives, and false negatives for the spam and ham classes:



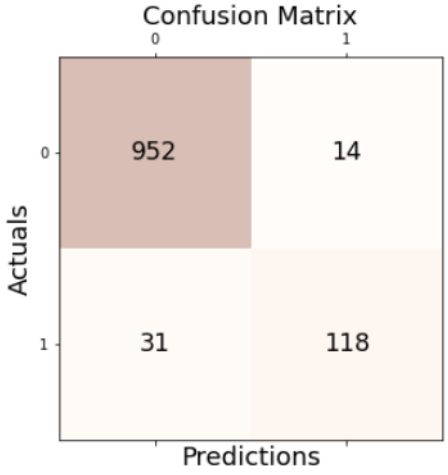


Figure 5:Confusion matrix of LDA vectors *Figure 6:Confusion matrix of LDA vectors and numerical features*

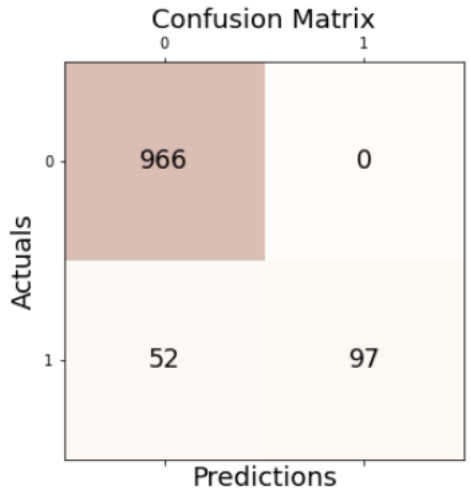


Figure 7:Confusion matrix of Tf-Idf vectors

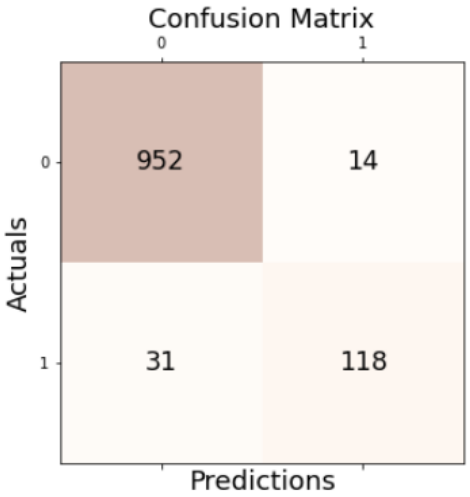


Figure 8:Confusion matrix of Tf-Idf vectors and numerical features

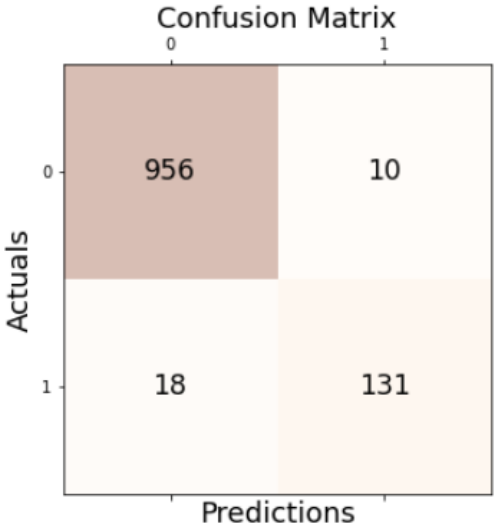


Figure 9:Confusion matrix of LSTM

Topic Coefficient: The topic coefficients of the logistic regression model for each LDA topic, sorted in descending order by coefficient value, which gives insight about which topics are most prevalent for distinguishing spam and ham emails:

	Coefficient	Topic
2	5.175083	3
4	2.419161	5
0	1.258472	1
9	0.754803	10

7	-0.252663	8
8	-0.887010	9
3	-1.334249	4
5	-1.597581	6
1	-2.368378	2
6	-3.167800	7

Table 6: Topic coefficient

Topic Words: Top words associated with each LDA topic, which can help with interpreting the topics and understanding why certain topics may be more important for classification.

Topic #1:	pleas girl cash collect repli went minut need holiday man
Topic #2:	ill later sorri im like come yeah ur night tomorrow
Topic #3:	free mobil txt claim phone prize messag award contact ur
Topic #4:	ok said finish lor lunch alreadi ask come told yup
Topic #5:	ur stop tone week repli free mobil txt send rington
Topic #6:	lor got da home wat ok dun im ur thk
Topic #7:	good day love time work happi got come im oh
Topic #8:	im text na think love miss need like gon wan
Topic #9:	ltgt dont know want pl like tell anyth sent send
Topic #10:	ur im want day new today best friend smoke come

Table 7: Topic words

ROC curve plot: ROC curve shows the trade-off between true positive rate and false positive rate for different classification thresholds. The area under the curve (AUC) is also displayed, which is a measure of the overall performance of the model.

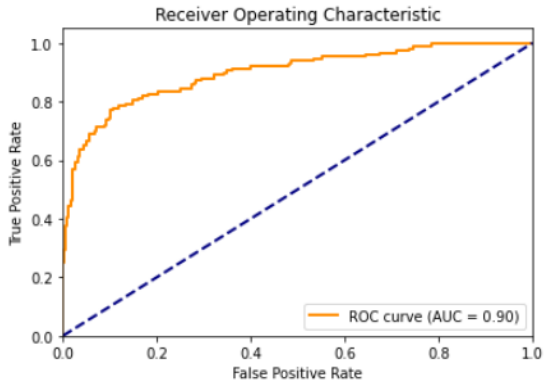


Figure 10: ROC curve of LDA vectors

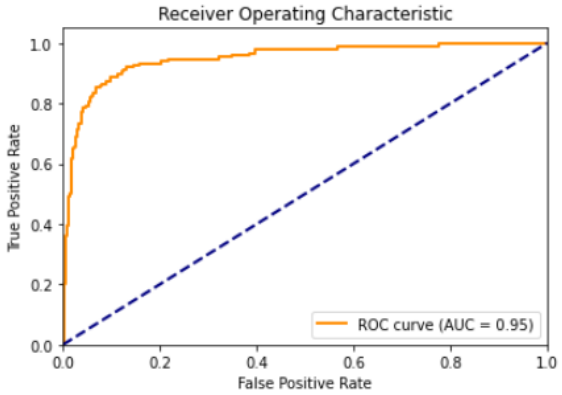


Figure 11: ROC curve of LDA vectors and numerical features

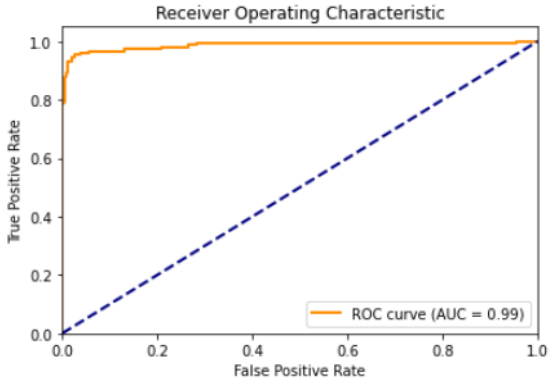


Figure 12: ROC curve of Tf-Idf vectors

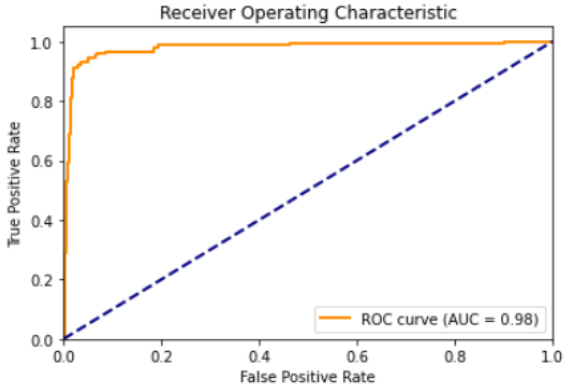


Figure 13: ROC curve of Tf-Idf vectors and numerical features

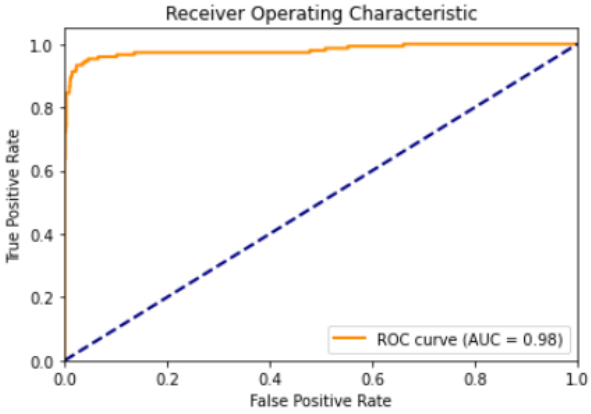


Figure 14: ROC curve of LSTM

3. Model Evaluation

Each model was evaluated using precision, recall, F1-score, confusion matrix, and ROC-AUC score. The results indicate that the combination of Tf-Idf Vectors

with numerical features (Total_Words and Distinct_Words) provide the best performance for logistic regression. Also, LSTM has the best performance among all tested models.

Discussion

The results show that the logistic regression model with tf-idf vectors has better performance in predicting spam messages than the logistic regression model with LDA vectors. The confusion matrix for the tf-idf model shows that it correctly predicted 966 non-spam messages and 97 spam messages, while incorrectly predicting 0 non-spam messages and 52 spam messages. On the other hand, the confusion matrix for the LDA model shows that it correctly predicted 947 non-spam messages and 75 spam messages, while incorrectly predicting 19 non-spam messages and 74 spam messages.

The difference in performance between the two models can be attributed to the nature of the dataset and the text vectorization methods used. Tf-idf vectorization calculates the frequency of words in each document and downweights words that are common across all documents, while LDA vectorization considers the topics that are present in the documents and assigns probabilities to each topic. Since the spam message dataset has a clear distinction between spam and non-spam messages based on the frequency of certain words, tf-idf vectorization might be a better approach than LDA vectorization for this specific dataset.

In additionally, combining LDA vectors and Tf-idf with Total_Words and Distinct_Words as input features can potentially improve the performance of the model by providing additional information about the text data.

The LDA vectors capture the latent topics in the text data, and the Tf-idf vectors capture the importance of each word in the text data. By combining these two types of vectors with Total_Words and Distinct_Words as input features, we can potentially provide a more complete representation of the text data. The Total_Words and Distinct_Words features provide information about the length and complexity of the text, which can also be useful in predicting the category of the text data.

The combination of these input features can help the model capture more complex relationships between the text data and the category variable, potentially improving the accuracy of the model. For example, the model may be able to identify specific combinations of words that are strongly associated with the spam or ham category, or it may be able to identify topics that are particularly relevant to the spam or ham category.

Also, LSTM with Doc2Vec has the best performance among all models. Doc2Vec captures the semantic information of entire documents by considering the context of words within the document. By doing so, Doc2Vec creates a more comprehensive representation of the text, enabling the LSTM model to better understand the relationships between words, phrases, and the overall meaning of the message. This enhanced semantic representation allows the LSTM with Doc2Vec more accurately differentiate between spam and non-spam messages.

Conclusion

Based on our experiments, we conclude that the LSTM model with Doc2Vec embeddings offers the best performance for spam messages detection among the models tested. This finding suggests that the combination of LSTM networks and Doc2Vec is a powerful and efficient approach for tackling the spam messages detection problem. The superior performance of the LSTM-Doc2Vec model for spam detection can be attributed to the effective representation of semantic information and context, the preservation of sequential information, robustness to noisy data, and reduced dimensionality. These factors together enable the model to better understand and classify text data, making it a powerful solution for detecting spam messages.

Improvement

Future work may involve exploring additional improvements, such as optimizing hyperparameters, incorporating attention mechanisms, or testing the model's performance on different datasets and languages.

References

SMS text Dataset:
https://www.kaggle.com/datasets/team-ai/spam-text-message-classification