Spam Detection Project Documentation

Kevin Zhuo and Daniel Zhang

1 Introduction

Spam detection has become an increasingly important challenge in the modern digital age, as the proliferation of unsolicited and unwanted messages, emails, and online content threatens to overwhelm communication channels and disrupt productivity. To combat this challenge, various techniques and algorithms have been developed to identify and filter out spam. In this project, we aim to utilize and analyze the efficacy of Deep Learning models in sorting SMS messages as either "Spam" or "Not Spam". Specifically, we test three different models based on the BERT architecture and analyze their performance on numerous evaluation metrics to determine which model is the best for spam detection.

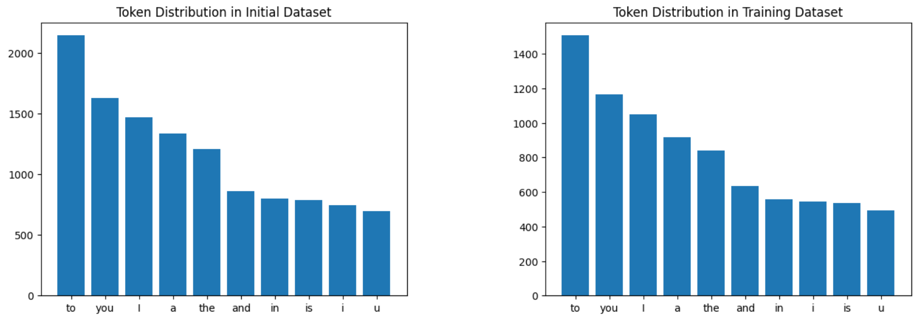
2 Previous Solutions

Previous solutions to spam detection have evolved significantly over time, with early approaches relying on more deterministic methods and evolving towards more sophisticated and effective methods. One common way to tackle this issue in the past was to include a rules-based approach, where certain keywords or other characteristics were set in place to determine spam messages. However, as machine learning techniques have grown more powerful and adaptable to the ever changing digital landscape, language models have began to become more prominent in this task of text classification.

3 Dataset

The dataset that we are using for the analysis is the "SMS Spam Collection" dataset (https://archive.ics.uci.edu/dataset/228/sms+spam+collection) which is a collection of 5574 messages, tagged according being ham (legitimate) or spam. The messages are collected from three main sources: the Grumbletext UK Online Web Forum, NUS SMS Corpus which consists of messages from Singaporean students, and messages from a PhD Thesis.

We performed a preliminary analysis of our dataset, primarily to explore the degree of (im)balance within the data. We found that in the intitial dataset, there were 747 spam messages and 4827 ham messages. After splitting our data



(a)

(b)

Figure 1: Token Distribution For Initial and Training Dataset

in training, validation, and testing, the training dataset had 522 spam messages and 3379 ham messages. This yields the valuable insight that our data is fairly imbalanced - something we will address with our evaluation methods.

We can see that from the distribution of tokens in the dataset, the most common tokens in both the initial and the training dataset are stop words, creating an exponential distribution for the tokens.

4 Proposed Method

We believe that using Transformers models, specifically models based on BERT, have the potential to better distinguish between spam and non-spam. In our project, we will first tokenize the sentences using the BERT embeddings, converting them into numerical inputs which can be fed into the BERT model. Then, we use a fine-tuned BERT model in a supervised learning approach on the training dataset. The hope is that eventually we can evaluate on the model on the held-out test set and achieve good results using the BERT model.

To this effect, we employ 3 different models to identify and filter out spam messages from emails, social media, or other types of communication. For each of these models, our notebook first loads in the dataset and splits it into training, validation, and testing sets. After a preliminary data analysis is performed, we will create a general pipeline streamline the process of training our models. While doing this, we will set the hyperparameters for the model as well as define the evaluation metrics that will be utilized. Lastly, import the three models, train them given the training arguments that we have set, and then test out the model on the validation and test set, a step we will discuss more in our Evaluation Method section.

The first type of model we use is DistilBERT: a version of BERT that is computationally lighter yet preserves much of the original model's power. As our first model, we want to gain a baseline sense of how effectively the BERT architecture can perform in relation to spam detection, so DistilBERT (a smaller, faster, cheaper and lighter version of BERT) will be a robust starting choice.

The second model we use is RoBERTa, another BERT-inspired model that shares the same architecture as its namesake. RoBERTa builds on BERT by changing key hyperparameters, training with much larger mini-batches and learning rates, and tweaking the pretraining process. While the model is different, the process in which we assess the model remains the same.

Lastly, we execute our task with ALBERT. This model distinguishes itself by splitting the embedding matrix into two smaller matrices and leveraging repeating layers split among groups to lower memory consumption and increase the training speed of BERT. Once again, we will tokenize the inputs using this model, define, train, and evaluate.

After employing these different methods, we will assess their relative performances.

5 Evaluation Method

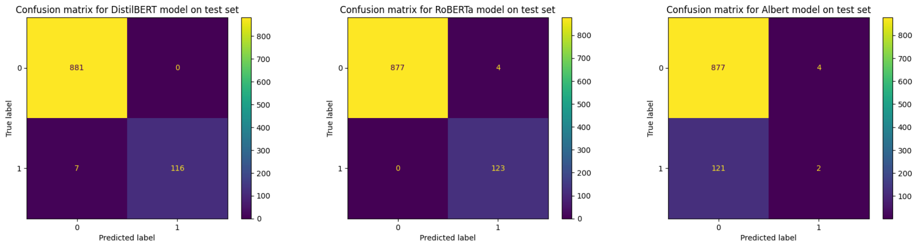
To evaluate the models that we have chosen, we use four main metrics: accuracy, f1 score, weighted accuracy, and a confusion matrix. Accuracy measures the overall proportion of correctly classified instances, while the F1 score provides a balanced evaluation by considering both precision (correctness of positive predictions) and recall (ability to find all positive instances). However, when dealing with imbalanced datasets like our own, weighted accuracy offers a more representative assessment by accounting for class distribution. Complementing these quantitative metrics, the confusion matrix provides a comprehensive overview of the model's performance, detailing the number of true positives, false positives, false negatives, and true negatives.

6 Results and Discussion

1. Let us consider the first model we attempted: DistilBERT. It yielded a test accuracy of 0.993028, a f1 score of 0.970711, and a balanced accuracy of 0.971545. Below is a confusion matrix describing DistilBERT's performance.

2. Now the second model: RoBERTa. It yielded a test accuracy of 0.0.996016, a f1 score of 0.984, and a balanced accuracy of 0.997730. Below is a confusion matrix describing DistilBERT's performance.

3. Let us consider the third model we attempted: ALBERT. It yielded a



(a) DistilBERT confusion matrix

(b) RoBERTa confusion matrix

(c) ALBERT confusion matrix

Figure 2: Confusion Matrices for the Models

test accuracy of 0.875498, a f1 score of 0.031007, and a balanced accuracy of 0.505860. Below is a confusion matrix describing DistilBERT's performance.

Let us analyze these results. Based on the numbers, it appears the RoBERTa ranks first, followed closely by DistilBERT, and ALBERT falling far behind. RoBERTa's accuracy, f1, and balanced accuracy all landed above 0.98, and it scored higher than DistilBERT across these three metrics, too. Still, DistilBERT represents a robust choice, as its balanced accuracy score of 0.970711 still performs well.

ALBERT lags far behind these two models. While its test accuracy is above 0.87, its f1 is a mere 0.031. Given that our dataset is imbalanced, this disparity makes sense. It suggests that our model is biased towards the 'ham' classification, leading to high precision for that class but low recall. In other words, ALBERT is good at correctly identifying instances of the ham but performs poorly in capturing instances of the spam. If we examine the confusion matrix, it reveals that ALBERT assigns nearly every value the same predicted label: that of the majority class ham. Thus, it yields a high accuracy but poor f1 and balanced accuracy.