CS 549 : Computer and Network Security

Term Project

Detection of Spams using Language Models

Aayush Sanjay Agarwal 160101002
Harshit Srivastava 160101033

Github link - https://github.com/harshitsriv10/cns_term_project

Introduction

Spams have become a major problem in the emailing. People have been trying to evade the security of email providers to include unwanted advertisements, viruses etc. to the user. Most of the models implemented for this is mechanical detection of hints that help us detect spams. Use of machine learning in this field is fairly new. We have used language models like Term Frequency (TF) and Doc2Vec and various machine learning algorithms for prediction.

In this paper, we first talk about a few current problems currently being faced by email companies. We then talk about various non-ML and ML approaches for solving this problem. We then talk about the implementation for our project and it’s analysis.

Recent Problems in Spam filtering

The volume of spam emails have been increasing ever since emails have become a popular medium of communication. These emails can be meant for promotions, advertisements, spreading viruses and frauds trying to dupe people.

The number of spam emails received can be so high that if not filtered properly can lead to flooding of mail boxes and cause people to miss out on important emails. Thus spam filtering is an important problem which needs to be focussed on. At the same time, care should be taken that important emails are not regarded as spam and deleted. As the spam filtering algorithms have advanced, the spammers have also adopted new techniques to bypass the spam filter. Thus, spam filtration is a process which requires constant attention and adopting new techniques to counteract the methods of the spammers.

Non ML Methods

These methods are not too common nowadays and they usually work based on some heuristics.
There are usually 2 ways of spam classification - **Pre Send** and **Post Send**. Pre-Send filtration works before the email is sent to the recipient while Post-Send filtration classifies the email after it has reached the recipient. Pre Send Filtration helps to reduce the network usage as the email classified as spam is never sent. But if the false-positive rate of the filtration method is high, some important emails may never be sent causing problems to the users hence this method is not used much nowadays.

One of the most basic methods to classify spam emails that is used is to **blacklist** certain IP’s or email addresses that are known to send spam emails. This is not a robust method and can be easily avoided by the sender of spam emails by creating new email accounts at regular intervals thus avoiding the spam filter.

Hence looking just at the sender of the email is not enough. There is a need to analyze the message itself to identify whether the email is spam or not. This is known as **content-based spam filtering**.

One method of content-based spam filtering is as follows -

A dataset of already known spam and ham(non-spam) emails is used.
First, the HTML tags are removed from the emails and the message is divided into a list of words. The stopwords are removed from the list of words and only the meaningful words are kept. The frequency of each word is counted in spam and ham emails separately. The most common words in spam emails are regarded as spam keywords and the most common words in ham emails are regarded as ham keywords. If a word is common in both, it is disregarded.
On the arrival of a new email, the number of spam and ham keywords are counted in it. If the spam keywords cross a certain threshold, the email is classified as a spam email.
For Example - Let’s say the word “promotion” occurs more frequently in spam emails while it is not so common in general emails. It is regarded as a spam keyword, now if a new email arrives containing the word “promotion”, it is more likely a spam email.

This is much more effective than just blacklisting certain addresses. The major drawback of this method is that spammers try to obfuscate the words in the emails by introducing special characters like replacing “a” in a spam keyword with the symbol “@”. Hence to improve the filtration process it works better to replace these special characters with meaningful characters beforehand and taking the count of special chars into account while classifying emails

ML Based Methods

Most of the machine learning based spam filtering is done in three steps : pre-processing, feature selection and then classification. All of the steps play an important role in classification of mails into spam and ham. Most of the email providers use the latest machine learning algorithms and methods like optical character recognition.

Pre-processing is the first step performed when an email is received. The email consists of two main parts : header and the body. Header consists of the sender, subject and other information that can be used as a feature. The body contains text with images and attachments. The text needs to be transformed to convert the message into meaningful tokens that can be fed to the model in the later step. Various language models are used to convert the text into tokens.

Feature selection step is the step of extracting the relevant features from the email that can be helpful to determine whether the email is spam or ham. This is used to remove the irrelevant information and reduce the size of the data that is fed to models in the next step. It makes the filtering,both training and prediction, light and fast.

Classification step is the step that determines that email is spam or ham. Various machine learning models are being used for this classification, we will discuss some of them:

Clustering:Clustering means categorising similar patterns in a particular class. Clustering is extensively used nowadays as it can be used on both unsupervised and supervised learning. There are two major types of clustering that are used : Density based clustering and K-means clustering.

Density based clustering is often used in document based clustering. It can be used on encrypted documents, hence keeping the confidentiality. In this, various peaks in the density graph refer to various clusters.

K-means clustering is used when the number of classes are predefined, in our case ham and spam. This clustering divides the dataset into predefined k clusters with each cluster resembling similar features. This has a high success rate but requires the whole dataset to be stored at the time of testing resulting in huge overhead of space and time of execution.

Naive Bayes: This method is based on a probabilistic model based on Bayes theorem. It gives accurate likelihood and is robust to noises in the dataset. It is simple and effective and is successful in spam filtering scenarios. It can only be used on supervised datasets.

SVM:Support Vector Machines (SVM) are a set of algorithms that try to seperate the dataset by hyperplanes. These algorithms take time to train as we have to find complex multi dimensional hyperplanes but are highly efficient in the testing phase.

CNN: Convolutional Neural Network (CNN) are a set of neurons arranged in a graph-like fashion where there is an input layer that takes input features, hidden layers that do processing and output layer that gives the classification. Every layer passes it's output as the input to the next layer. Each neuron does a small computation which is a linear function followed by an activation function. Neural networks are highly scalable and can be extended to form deep neural networks and thus is extensively being used for spam filtering.

Ensemble:This is a fairly new approach in which classifications from various classification methods are combined to give an improved model for classification. Bagging and boosting are two important features of ensembling.

Bagging (Bootstrap aggregating) is when we aggregate small learning models in parallel independent of others and then combine the results by some kind of averaging.

Boosting is when we join the small learning models in a sequential adaptive manner to give a better end result.

Random Forest:Random forest uses various decision trees for making the classification. Decision trees are tree-like structures of which each leaf is a classification. In a random forest, a poll is taken from the classifications of different trees and a winner is chosen. This is a specific type of ensemble classifier. Though the precision is low as a poll is taken for classification but it gives less error as compared to other models.All the trees need to be stored in the model and thus has memory overhead.

Firefly:This algorithm is a population based heuristic algorithm. This algorithm tries to find multiple probable solutions by means of population physiognomies.In this, like fireflies, information is shared between probable solutions leading to increased accuracy. This is a heuristic and hence gives good results most of the time. Since multiple probable solutions are developed, the memory requirement is high.

Rough Set:This method is based on the breakdown of categorisation of inexact, ambiguous or partial information from data based on training. It is based on the hypothesis that there is some amount of information with every data available. It is a mathematical model that concentrates on uncertainty. It tries to find the redundancy and dependencies between the features of the data. It is both time and memory efficient.

Implementation

We have implemented two types of two types of language models, namely Term Frequency and Doc2Vec, for preprocessing. We then tried various types of machine learning models on them to maximize the accuracy for prediction of spam and non-spam.We used the Spam Mails dataset for training the machine learning algorithms.

Term Frequency (TM)
Term frequency refers to the number of times a word appears in the document. This is a simple but effective method for converting text to vectors. For this, we used the observation that many spam emails have certain words occurring more frequently than others, like lottery, money, urgent etc. while some words are irrelevant and occur in all the mails. Stopwords and punctuations occur in all the mails. So, we removed all such stopwords and punctuations as they were irrelevant information for the model. Then we made a list of words that occur frequently in spams and made a list of it. We used these words as the feature as the features for training the machine learning algorithms. We also added the count of words that were not in the list. To normalise the data, we divided it by the total word count so that the size of the mail does not affect the model.

Doc2Vec
Doc2Vec is a model which converts a document into a vector of fixed length containing numbers. The major advantage of doc2vec over term frequency is that, in term frequency the context of the words and their order is lost while in doc2vec these properties are preserved which can be useful to find similarity among documents in a more accurate way. Moreover, in term frequency different words with similar meaning are regarded as different words with no similarity between them while in doc2vec the vectors for these words would be similar. Therefore, we can see that using doc2vec can lead to a better classification of spam emails. We trained the doc2vec model on all the emails in the training data to get a vector representation of all the emails. The naive method we tried first was computing the vector representation of spam and ham emails by taking the average of the vectors of the individual emails. For a new email, we first compute its vector and check its similarity to the spam and ham vectors, and classify it as spam or ham using the cosine similarity scores(whichever is higher). To improve on the accuracy of classification, instead of computing the average vector for spam emails we used the individual vectors of the emails as a feature for the classification models like in the term frequency implementation.

ML models
We have used the Spam Mails dataset for training the algorithms. This dataset had three features, the mail data (with subject,address etc.), spam/ham and label according to spam/ham. We used only two of these three features. We only used supervised ML models as the dataset already had labels. We used decision trees, SVM, random forest, logistic regression, naive Bayes and MLP classifiers. Details of the observation is given in the next section.

Analysis

We implemented two language models, Term Frequency and Doc2Vec. This gave us a vector format which could be fed to the machine learning algorithms. We used various machine learning algorithms whose accuracy score has been mentioned in the table below:

ML algo./Language model	Term Frequency	Doc2Vec
Logistic Regression	71.8%	94.4%
Decision Tree	73.7%	77.4%
SVM	72.0%	93.1%
Random Forest	77.6%	89.6%
Naive Bayes	71.8%	92.9%
MLP	71.4%	94.0%

Table : Accuracy score for different ML algorithms on language models

It can be seen clearly that Doc2Vec performed much better in predicting the spams than the term frequency. Term frequency, though a good way to detect spam, is naive and thus can be fooled easily. Doc2Vec converts text into vectors according to the context of the text. It produces a numerical representation of the text. Pre-processing of data for Doc2Vec however took more time than TF. While testing, Doc2Vec was slightly faster than TF. We can see that decision trees perform poorly in both the language models. This is because it forms a tree with decisions at the leaves. A single tree is not obvious to give predictions in complex cases. Random forest which has multiple such trees perform better. For this, we used trees with levels upto 10. Naive Bayes and SVM perform good as well. These models are good for predictions with two labels. For MLP, we used different sizes of the classifier. We concluded that the 5x5 MLP classifier performed better than similar models without making the model too bulky. After this, the trends started to saturate. Best performance was shown by Logistic regression which is an excellent model for predictions with two layers. Doc2Vec was much better to get the trend from the texts and hence performed well.

References

1. E-Mail Filtering
2. Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends
3. Security Evaluation of Pattern Classifiers under Attack
4. Machine learning for email spam filtering: review, approaches and open research problems
5. Content Based Spam Email Filtering
6. Spam Filtering Email Classification (SFECM) using Gain and Graph Mining Algorithm
7. MEADE: Towards a Malicious Email Attachment Detection Engine

Dataset :

https://www.kaggle.com/venky73/spam-mails-dataset?fbclid=IwAR2Gx4Ho5HCHN6k-oJiE3ayTE02itzCThwDxkk3Y6L1rQLTRbNr68iQuRio