SMS Spam Detection

Samantha Allison, Connor Ruff, Ying Qiu, and Grace Kulin

Abstract

For our project, we will develop and test an algorithm that detects whether an SMS message is spam or not. Our project is socially good because spam messages often have malicious intent including stealing personal information so detecting whether or not the message is spam can protect phone users. Databases containing text messages along with labels telling whether or not they are spam already exist so our project goals can be accomplished within our timeline. Our proposed approach requires two steps: we will perform language processing on the text messages in the dataset and then we will utilize AI techniques (specifically, vectorizing the data and then using a classifier on it) to determine which messages are spam and which are not.

Introduction

Our project is to develop an algorithm that can determine whether or not a text message is spam. This is beneficial to society because these texts usually have malicious intent such as scamming the user for personal information (Credit cards, Social security, etc). Additionally, they usually use clickbait phrasing which tricks the user into believing that the text is regarding something important. One out of every six people will fall victim to these scams. In 2018, in total American's lost over $10 billion dollars from falling victim to spam text messages. Having a way to detect whether or not a text message is spam, will help users decipher which texts are legitimately important and which to disregard for their safety.

Related Work

Due to the limited length of SMS (140 bytes or characters [1]) and lack of headers, the feature space that can be used in SMS spam classifiers is much smaller as compared to e-mail. Another challenge is that SMS is usually written in informal language or symbols or abbreviations [2]. To tackle these challenges, researchers have come up with various spam detection techniques over the past decades. The most citable techniques include Bayesian [3], Support Vector Machine (SVM) [4], Decision Tree (DT) [5-7] etc. Other than detection techniques, the selection of feature space is important to achieve high detection accuracy. Mujtaba and Yasin [8] found that Naive Bayesian outperforms the other algorithms when they used the message size, class of messages, frequently occurring monograms and occurring diagram as the feature space. Sethi et al. [9] employed the information, such as the length of the messages, information gain matrix, and the raw text messages. Xu et al. [10] considered static features such as the number of messages, temporal features like the number of messages sent in a day and network features, for instance,

the number of recipients and clustering coefficients in their study and concluded that their method (SVM and K-Nearest Neighbors) provided a better performance when these three types of features were employed.

Proposed Approach

We will need large datasets containing real SMS messages that have been labeled as either spam or not spam ("ham"). This will allow our algorithm to adjust its parameters properly based on whether or not the message it just looked at was indeed spam. Kaggle.com has wide availability of such data, with messages that have been deemed as spam as well as legitimate SMS messages in the same datasets. The dataset found here will serve as the principle training dataset.

Training our algorithm will require two basic steps. First, we will need to do some language processing on the text message in the dataset. This will be a basic tokenization of the text into words, and will also filter out "stop words" that are very common, and can be disregarded when classifying the text. Then, we can apply AI techniques to analyze this processed data. Specifically, we will vectorize the data, and use a "Naive Bayes Classifier" (from sklearn) to train the algorithm to recognize certain tokens as associated with spam. With a large enough dataset, this should produce an accurate predictor of Spam vs "Ham".

Additionally, after the algorithm has been trained, we will need additional data to be able to test it, and verify its accuracy. We can follow the same approach to do this. We will feed the trained algorithm SMS messages that it has not seen before, that have pre-determined "spam" truth values, and check that our output matches this.

Below is a list of Python libraries we will be using, and what they generally will be used for:

- Pandas and Numpy: reading CSV file with data, doing some pre-processing on the training data sets
- NLTK: performing tokenization and stemming the dataset to prepare it to be trained
- SKLearn: perform the machine learning to train the algorithm

Expected Deliverables

Our first set of deliverables, for October 17th, will include a clear set of objectives for our project, properly organize data, and complete exploration of our data. Success will be determined by whether or not we are able to successfully identify a pattern to determine which SMS messages are spam and which are not.

Our next milestone, on October 31st, will include having the tools to train our model prepared (by finding and exploiting patterns in our data) and having validation and testing of our model complete. At this stage we will determine success by whether or not the validation of our model is successful (i.e., if it can correctly identify which SMS are spam and which are not based upon its training).

The deliverables for our presentation, on November 11th, will be a clear overview of our work and how we met our project goals.

Our final project deliverable, for November 12th will be our completed AI model that successfully identifies spam or non-spam SMS messages.

References

[1] Nagwani, Naresh Kumar. "A Bi-Level Text Classification Approach for SMS Spam Filtering and Identifying Priority Messages." *International Arab Journal of Information Technology (IAJIT)* 14.4 (2017).

[2] Sjarif, Nilam Nur Amir, et al. "SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm." *Procedia Computer Science* 161 (2019): 509-515.

[3] SHUKLA, ANJALI. "SMS SPAM CLASSIFICATION BASED ON NAÏVE BAYES CLASSIFIER." *Journal of the Gujarat Research Society* 21.14s (2019): 48-59.

[4] Sjarif, Nilam Nur Amir, et al. "Support Vector Machine Algorithm for SMS Spam Classification in The Telecommunication Industry."

[5] Ghodke, Tejashri, and Vijay Khadse. *Effective Text Comment Classification Using Novel ML Algorithm-Modified Lazy Random Forest*. No. 3987. EasyChair, 2020.

[6] Goswami, Vasudha, Vijay Malviya, and Pratyush Sharma. "Detecting Spam Emails/SMS Using Naive Bayes, Support Vector Machine and Random Forest." *International Conference on Innovative Data Communication Technologies and Application*. Springer, Cham, 2019.

[7] Abayomi-Alli, Olusola, et al. "A review of soft techniques for SMS spam classification: Methods, approaches and applications." *Engineering Applications of Artificial Intelligence* 86 (2019): 197-212.

[8] Mujtaba, Ghulam, and Majid Yasin. "SMS spam detection using simple message content features." *J. Basic Appl. Sci. Res* 4.4 (2014): 275-279.

[9] Sethi, Paras, Vaibhav Bhandari, and Bhavna Kohli. "SMS spam detection and comparison of various machine learning algorithms." *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*. IEEE, 2017.

[10] Xu, Qian, et al. "Sms spam detection using noncontent features." *IEEE Intelligent Systems* 27.6 (2012): 44-51.