Capstone Proposal

Esther Liao
December 28, 2017

Domain Background

As the world continues to move forward in an age of ever advancing technology, calls to action maintaining a free-flowing internet have become increasingly challenging with the presence of spam messages. Many email service providers have rigorously worked on updating spam filters that correctly identify spam messages. However, the same has not been nearly as true for text messages even as text messages have become more and more widespread. While SMS spam certainly inherits features from email spam, it is still challenging because text messages are inherently shorter than their counterpart.

As text messages are easier to read in full and at the same time potentially carrying important information, it is absolutely critical that SMS spam is classified correctly. At the same time, SMS spam can cause a disruptive experience and should be pr operly identified as email spam is filtered. As text messaging becomes an increasingly popular mode of communication, there should be developments made to properly classify SMS spam to allow people an optimal experience with text messaging.

Problem Statement

The more urgent nature of text messages makes it vulnerable to spam attacks which could potentially prevent or delay the reception of important information. A potential solution to this problem could be to create a spam filter for texts likened to that of an email spam folder. To begin creating such a filter, text messages must first be classified into spam and non-spam. For every message, the proposed solution should be able to correctly classify whether or not it is spam. The nature of this problem is that of a binary classification.

Datasets and Inputs

For this project, the SMS Spam Collection Dataset from the University of California in Irvine will be used[1]. The dataset has been made publicly available on Kaggle. It contains a total of 5,572 messages that are tagged as ham (legitimate) or spam. The dataset itself is split between two columns, one representing the label (ham or spam) and another containing the raw text. The distribution of the dataset is not very balanced, with 4,825 of these messages tagged as being "ham" while the remaining 647 messages being tagged "spam."

The simplistic nature of this dataset will make it easier to parse through, and as the project comes together, decisions may be made to use another tool such as WordNet or other tree visualization to help the classifier identify what words are indicators of SMS spam. If time permits, we will also attempt to use more advanced NLP techniques to analyze text.

Solution Statement

A solution to the problem would be to create a spam detector that can either hide SMS spam or raise alerts when spam is detected. In order to achieve this, this project will focus on creating a

[1] http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/

classifier that reaches above 90% accuracy in classifying messages as spam or ham. We will split the dataset into a 80:20 ratio for training and testing respectively. I plan on encoding the text by using Word Vectors.

Benchmark Model

In this project, I will use K Nearest Neighbors as my benchmark.

Evaluation Metrics

As mentioned above, the success of the classifier will be evaluated on its performance in correctly classifying the testing set. After meeting the 90% accuracy threshold, the classifier may be tested on a new test set to see its performance and to ensure that the training set did not bias the classifier.

Project Design

I will begin by preprocessing the data and cleaning extraneous data included with the dataset so that we are left with the text and spam label. While doing text analysis on the words, I will either filter through the words in each message or provide some sort of stripping of the message when applying machine learning (e.g. upper/lower case, punctuation, etc.). I will convert the text into word vectors, doing multiple trials to try and achieve optimal results. Different visualizations and graphs will be used to depict the results of applying the classifier in question. For testing purposes, depending on time complexity, I might use a smaller segment of the data. After getting the results of the highest achieving classifier, I will attempt to generate a word map for both spam as well as ham messages to see which words the classifier prioritized when identifying if a message was spam or not.