String Pattern Matching: Hybrid Ad-Specific Spam Classifier

By Anirudh Reddy

What models/algorithms are we using to accomplish string and pattern matching?

Multinomial Naive Bayes (NB) - Numerical/Algorithm

The Naive Bayes Algorithm tokenizes each word and calculates the frequency of how often it appears. The more it appears in a set that had already been classified earlier would allow it to determine for which class (spam or ham) that word would have a higher probability. The equation for the calculation of such a probability is as follows:

$$P(Class|Word) = [(P(Word|Class) \times P(Class)]/P(Word)$$

When predicting whether a message is spam or not in a group, it would be simplified and calculated as:

$$P(Class|Word) = [\Gamma^{n}_{i=1} (P(Word\_i|Class) \times (P(Class)]/P(Word\_n)$$

*Where $P(x \mid y) \rightarrow$ The probability that x is … given that y is …

[1][2][3][4]

Support Vector Machine (SVM) - Numerical/Algorithm

The Support Vector Machine algorithm plots data points as an n-dimensional vector, in n-dimensional space R^n. A hyperplane that is n-1 dimensions is then drawn between the data points. This can be adjusted to find a better fitting in accordance to the training data but must be done through the creation of margins.

To find this margin, 2 parallel hyperplanes are constructed, one on each side of the original hyperplane. These two planes are typically almost perfectly aligned with the nearest data point on both sides [4][5][6].

Support Vector Classifier (SVC) - Numerical/Algorithm

The specific type of SVM I have implemented into my code is a Support Vector Classifier (SVC). This version of an SVM is made specifically in regards to classifying features. Essentially, if a data point can can be classified by the current hyperplane, then the data point is not a support vector. If it is not classifiable, it becomes a support vector for all of the data points present (given that at least two already exist to form).

Then, using the new support vectors, the hyperplane is optimized by minimizing the following function [7]:

$$L_D = 12 \beta^T K \beta - \langle c, \beta \rangle,$$

How a hybrid model was formed – Application

I used the pandas and numpy python packages to create a dataframe that displays information similarly to an excel sheet and can be manipulated to fit our needs.

Finally, I used the sklearn python package to train the Multinomial Naive Bayes and the Support Vector Machine models and combine them into a “hard” voting classifier, which would compare the two model’s accuracy scores and employ that for the predictions [8][9]. Commands to measure the accuracy was also imported using sklearn.

Dataset

We used 3 datasets:

- Original Dataset used for Training/Test Split
  - Tokenize messages and has both models learn keywords to be applied on its own test data or another dataset entirely
  - This dataset was mostly already advertisements offering some kind of product or service.
  - Used a 75 : 25 training to test ratio as it provided the highest accuracy for the models in both my testing. Randomly shuffled messages.
- External Dataset
  - Used strictly for testing purposes given the original hybrid model was already trained.
  - Had a wide variety of spam messages from ads to phishing spam.
- External Dataset - Trimmed to just have Ad spam
  - Curated the original External Dataset to test how much more accurate it would be for the training model to test against something without keywords that might be unrelated.
  - Deleted about 20% of the original External Dataset due to containing unnecessary or unrelated info

Both the Original and External Dataset were found to be public datasets on Kaggle. The original contained about 4825 ham emails and 747 spam emails while the external set contained about 2170 ham emails and 433 spam emails.

Original Dataset: https://www.kaggle.com/datasets/phangud/spamcsv

External Dataset:
https://www.kaggle.com/datasets/nitishabharathi/email-spam-dataset?select=lingSpam.csv - Select lingSpam.csv for download

Trimmed External Dataset:
https://docs.google.com/spreadsheets/d/1rpTVJ7OWXoB_YJhCEk_mUGa5ilobkb_pDdprVdt4XaI/edit?usp=sharing - Download as csv before using

Results

SCORE COMPARISONS: SAME DATASET WITH TRAINING/TEST SPLIT

NB Accuracy Score: 0.9842067480258435

SVM Accuracy Score: 0.9827709978463748

NBSVM Accuracy Score: 0.9806173725771715

Results

SCORE COMPARISONS: ORIGINAL MODEL VS EXTERNAL DATASET

NB 2 Accuracy Score: 0.7930902111324376

SVM 2 Accuracy Score: 0.837236084452975

NBSVM 2 Accuracy Score: 0.837236084452975

SCORE COMPARISONS: ORIGINAL MODEL VS EXTERNAL DATASET (FILTERED FOR ADS)

NB 3 Accuracy Score: 0.8134715025906736

SVM 3 Accuracy Score: 0.8688720605819051

NBSVM 3 Accuracy Score: 0.8688720605819051

Problem and Contribution

The issue with a lot of spam filters now are that they are not very accurate in recognizing what is spam and ham. There are several trained machine learning models that on their own could prove to be useful and have yielded high accuracy rates. However, those tend to refer to a dataset that the original training set and test set has come from as opposed to an external dataset.

In an external dataset, there are several other issues that might factor in, such as dataset/concept shift and a large potential for keywords that the model might not recognize [10].

Problem and Contribution

With all of this in mind, I aimed to create a spam filter that utilizes the top two best performing machine learning models for classification to identify as much spam as it can and make up for the other's shortcomings if such a case was to show up.

Additionally, the intent with this spam filter is to focus only on one subject and its respective regular expressions. Creating a subject-specific model would eliminate many unrelated and misguiding tokens that could otherwise lead to misclassified spam, thereby increasing the accuracy of this hybrid model. This can be later utilized in a pool of subject specific models that could more accurately rid the user's inbox of malicious or unnecessary spam emails.

A hybrid model for spam classification is already not seen too frequently and the accuracy rates for when individual models are tested against external datasets drops significantly when compared to the original training dataset. However, having other models to make up for what the first may have missed is able to make up for the lost accuracy a considerable amount. Furthermore, through training and testing using mostly ad/commercial, pornographic, and lottery email messages, I was able to increase the accuracy rating of the hybrid model even more.

References

[1] V. Metsis, I. Androutsopoulos, and G. Paliouras, “Spam Filtering with Naive Bayes – Which Naive Bayes? .” Third conference on email and anti-spam (CEAS), 2006

[2] M. Abbas, K. A. Memon, A. A. Jamali, S. Memon, and A. Ahmed, “Multinomial Naive Bayes Classification Model for Sentiment Analysis.” IJCSNS International Journal of Computer Science and Network Security, Mar. 2019

[3] J. Kagstrom, “IMPROVING NAIVE BAYESIAN SPAM FILTERING.” Mid Sweden University Department for Information Technology and Media , 2005

[4] S. Gibson, B. Issac, L. Zhang and S. M. Jacob, "Detecting Spam Email With Machine Learning Optimized With Bio-Inspired Metaheuristic Algorithms," in IEEE Access, vol. 8, pp. 187914-187932, 2020, doi: 10.1109/ACCESS.2020.3030751.

[5] “Spam Filtering using SVM with different Kernel Functions.” International Journal of Computer Applications, 2016

[6] S. Amarappa and S. V. Sathyanarayana, “Data classification using Support vector Machine (SVM), a simplified approach.” International Journal of Electronics and Computer Science Engineering, 2014

References

[7] K. W. Lau and Q. H. Wu, “Online training of support vector classifier,” ScienceDirect, https://www.sciencedirect.com/science/article/abs/pii/S0031320303000384?casa_token=TQavjupBIAQAAAAA:YHip15551YXd-ieFzDFTjD2KsxeNiyGPJ0Bhs2z9IXM6WvCfGcwK6YOvxyV2bKyFRlvNEJVndw (accessed Oct. 27, 2023).

[8] H. E. Kiziloz, “Classifier Ensemble Methods in feature selection,” ScienceDirect, https://www.sciencedirect.com/science/article/pii/S0925231220313151?casa_token=KVEwWKhFEb4AAAAA:rrMDkTzbghqVWgPKwhr7ZKCXhyHQ3PokBoIUuHzOBeBnacpbrEJ20Tp9Pg6Kx5_led4OYKxujw (accessed Oct. 27, 2023).

[9] D. Ruta and B. Gabrys, “Classifier selection for majority voting,” ScienceDirect, https://www.sciencedirect.com/science/article/pii/S1566253504000417?casa_token=vqvBEB9RZ9kAAAAA:tUXHBkIpElsle8NskpS9VTKjhAPBxGOOHS-oyn_vt5u9tcU_PRm8z5RAzWMiTwucpq8Je1yyeg (accessed Oct. 27, 2023).

[10] F. Janez-Martino, R. Alaiz-Rodriguez, V. Gonzalez-Castro, E. Fidalgo, and E. Alegre, “A review of Spam Email Detection: Analysis of spammer strategies and the dataset shift problem - artificial intelligence review,” SpringerLink, https://link.springer.com/article/10.1007/s10462-022-10195-4 (accessed Oct. 27, 2023).