Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/48/20241337

Spam email classification based on SVM, Transformer and Naive Bayes

Yijun Qiao

School of Software, Harbin Institute of Technology (Weihai), Weihai, China

2021212063@stu.hit.edu.cn

Abstract. As a matter of fact, with the booming of information and big data, there are too many unwanted e-mails called "spam" sent to people's e-mail account in recent years. On this basis, it could lead to a lot of problems including occupying the public resources, causing financial loss and so on. With this in mind, spam filtering technique is in need to solve the problem and address the issues. In reality, based on previous analysis, machine learning methods are very effective in spam filtering. On this basis, this study carried out background research of machine learning algorithms in spam filtering, and find the spam e-mail dataset of Kaggle.com, and implement 3 algorithms on the dataset. According to the analysis, Transformer and SVM work better on the dataset, and SVM is the best. At the same time, the current limitations are discussed as well. In addition, the prospects are demonstrated in the meantime.

Keywords: Spam email, machine learning, Naive Bayes, Transformer, SVM.

1. Introduction

In recent years, with the development of Internet and e-mail systems, there were more and more unwanted emails called spam sent to every user. This has nowadays become a huge problem. Some spams are even in the disguise of reputable companies with the intention of cheating individuals to get sensitive personal information such as credit card numbers and passwords. This can lead to big financial losses if the users are deceived by the spam. Spam also prevents the user from making full and good use of time, storage capacity and network bandwidth. The memory space of email servers, communication bandwidth and CPU power are destroyed by a huge number of spams flowing through the network of personal computers. At the same time, spam is also an important way for viruses and hobbyhorses to spread. The threat of spam increases every year, accounting for more than 77% of global email traffic [1]. Users inadvertently disclose email addresses, which leaves chances to spam makers. In addition, RCPT, VRFY and other commands in SMTP protocol are used to compile scanning programs, and verify the addresses one by one from the mail server by means of dictionary attack, which is also the technique for some spammers to find email addresses. Advertising publishers make a list of user email addresses for purchases or collections and automatically send them in a mass loop over the Internet, causing spam to flood [2].

Since spam can cause so much harmful problems, automatic e-mail filtering systems are in need, and this seems to be the most effective methods to avoid spam. There are two general approaches used in e-mail filtering, i.e., knowledge engineering and machine learning. In knowledge engineering approach people set rules to categorize emails as spam or normal ones. A set of such rules should be created either

DOI: 10.54254/2755-2721/48/20241337

by the user of the filter, or by some other authority. By applying this method, people could hardly get the results they like because the rules must be constantly updated and maintained, which is not so practical because of wasting the time and lack of convenience for most users. Compared to knowledge engineering approach, machine learning approach is more efficient, because it does not require specifying rules. Instead, a set of training samples are collected. These samples are made of a set of pre-classified e-mail messages. A specific algorithm will be used to learn the classification rules from these e-mail messages [3].

There are lots of algorithms in machine learning can be used in e-mail filtering. Naïve bayes classifier is one of the most widely used algorithm in spam filtering. However, there are still so many other machine learning algorithm to solve the problem. naïve bayes classifier, LSTM, CNNs, support vector machine (SVM), Transformer. To find a valid subset of features for spam classification, an intelligent vector machine method was proposed by Hamed. In order to improve the binary differential algorithm (BDE), the fitness function of the selected feature subset is taken as the correlation coefficient. The selected feature subset was used for the calculation and evaluation of credibility in spam classification, and the global accuracy of the algorithm reached 94.55% [4]. When studying the random forest algorithm, Vinitha used minimum redundancy and maximum correlation as selecting a subset of features, which achieved 86% accuracy in sensitivity, correctness, and accuracy [5]. Sumathi proposed a model based on deep neural network and random forest, and the algorithm has high classification success and accuracy of 88.59% [6].

The motivation of this paper mainly focusses on making comparison among Naïve bayes, Transformer and SVM, analysing the results to get a model fitting better to the spam filter. The rest part will introduce these three algorithms concisely and compare them. The research process will be listed here. First, this study collects enough emails as dataset, then this study uses these three models and some other evaluation metrics to estimate. When the calculation results are obtained, the author would analyse and compare the fitting score of those models, and finally get the conclusion.

2. Data and method

The study utilized a spam email dataset obtained from the Kaggle website as the source of experimental data. The dataset, which contains 500 emails, was pre-processed to better suit the three algorithms. Naive Bayes is a probabilistic algorithm that calculates the probability of an email being spam based on the occurrence of words or features in the email's content. If the total of words probabilities goes over a threshold, the spam filter will classify the e-mail to the certain category of spam or ham. Processing spam with this algorithm can be understood as: Using Bayes theorem to predict a email composed of several words that one doesn't know whether is spam mail, it is the possibility of spam or normal mail, if the algorithm predicts the possibility of spam is higher, then this is spam e-mail, otherwise it is normal mail. If an attribute value does not occur simultaneously with a class in the training set, the training model will appear over-fitting. To avoid the information carried by the other attributes being erased by attribute values that did not appear in the training set, one usually does a correction called Laplace correction: N represents possible categories in the training set D, N_i means the number i attribute possible values, the Bayesian formula can be corrected to:

$$P(c) = \frac{|D_c|+1}{|D|+N} \tag{1}$$

$$P(x_i|c) = \frac{\left|D_{c,x_i}\right|+1}{|D|+N_i} \tag{2}$$

Support Vector Machine (SVM) is primarily employed in classification and regression problems. Its fundamental principle entails finding an optimal hyperplane in the feature space for classification purposes. The core concept of SVM involves mapping the data points to a high-dimensional feature space where they can be separated by a hyperplane. This hyperplane aims to maximize the margin between different classes of samples while accurately classifying them. The margin, known as the "maximum margin," represents the gap between the samples and the hyperplane. By learning the optimal

DOI: 10.54254/2755-2721/48/20241337

hyperplane, new samples can be mapped to the feature space and classified using a decision function. The decision function determines the category to which a sample belongs based on its relationship with the optimal hyperplane. The spam filter based on SVM uses the training model to extract the main features from the mail text, classify the known and unknown mail, and automatically identify and filter out the spam mail. This method is based on the text feature processing technology, by processing the text data of spam and normal mail, extracts the effective feature vector, obtains the training set and test set, and uses SVM algorithm to create the classification model.

As a neural network model based on the self-attention mechanism, the Transformer model is able to globally model each element in the sequence and make connections between the individual elements. The self-attention mechanism is a key component of the Transformer model. It calculates contextualized representations for each position in the input sequence by applying weighted combinations of elements at different positions. The computation of self-attention involves three steps: calculating attention scores, applying attention weights, and computing weighted sums. Calculating attention scores: Attention scores are computed by taking the dot product of queries, keys, and values to determine the importance of each position relative to others. Applying attention weights: Values are weighted by the attention scores to obtain weighted context vectors. The weighted context vectors are summed with the input sequence to produce the final output of self-attention.

Confusion matrix is a contingency analysis table in machine learning that summarizes the outcomes of a classification model's predictions. It organizes the records from a dataset into a matrix format, summarizing them based on two criteria: the actual categories and the categories predicted by the classification model [7]. The description is given in Table 1.

Table 1. Evaluation metrics.

		Predicted condition	
	Total population	Positive (PP)	Negative (PN)
Actual condition	Positive(P)	True positive (TP)	False negative (FN)
	Negative(N)	False positive (FP)	True negative (TN)

Accuracy refers to the proportion of samples classified correctly out of the total number of samples by a classifier, and it can then evaluate the overall classification performance of a classifier [8]:

$$Accuracy = \frac{TN+TP}{TN+TP+FN+FP} \tag{3}$$

Recall is the proportion of true positives (TP) out of all actual positives (TP+FN), and it can evaluate the classifier's ability to correctly classify positive samples [9]. The formula is as follows:

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

Precision is the proportion of true positives (TP) out of all predicted positives (TP+FP), and it can evaluate the correctness of positive predictions made by the classifier [10]:

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

F1-score is the harmonic mean of precision and recall, combining both metrics to comprehensively assess the classifier's performance [11]:

$$F1-score = \frac{2\times Recall\times Precision}{Recall+Precision} \tag{6}$$

3. Results and discussion

For the naive bayes the author does the feature engineering as these steps:

- Clear punctuation marks, mark a string as a word, calculate the number of occurrences of a word.

DOI: 10.54254/2755-2721/48/20241337

- Build a collection to store all appearing words, count the number of spam and ham messages, calculate the prior probability (the proportion of spam and normal messages in all sample messages), build a dictionary to store the words in a single message, and the number of the words.
- Go through all the test set, calculate P (content | spam) and P (content | normal mail) (all the words should do Laplace smooth), do the calculation of P (content | spam) pluses P (content | normal mail), and add the prior (P (spam) and P (normal mail)) to it. Finally predict, if spam _ score> ham_score is marked as 1, namely spam

For the SVM the author does the feature engineering as these steps:

- The dataset was loaded, and the words in the text data were split into lists. Subsequently, the data was prepared in a format suitable for SVM. The dataset was divided into training sets and testing sets, and the labels were converted into NumPy arrays.
- Next, the text data was transformed into numerical feature representations using TF-IDF vectorization.
- A default SVM classifier was created and trained. Predictions were made on the test set, and a classification report was generated for the default configuration. Following that, hyperparameter tuning was performed using GridSearchCV. A parameter grid was defined, and the best combination of hyperparameters was found through cross-validation. The model was then trained using this optimized configuration.
- Predictions were made on the test set, and a classification report was generated for the optimized model.
- Rcursive feature elimination (RFE) was employed for feature selection. The top 100 ranked features were selected. An SVM classifier was trained on the chosen features and used for prediction and evaluation.

For the Transformer the author does the feature engineering as these steps:

- Define the input layer, where the shape parameter specifies the length of the input sequence. Define the embedding layer, where vocab_size indicates the size of the vocabulary and embedding_dim represents the dimension of the word embeddings. The weights parameter can be used to pass pre-trained word embeddings, and trainable should be set to False if the embeddings are not trainable.
- Create a masking layer. It sets positions that are not equal to 0 in the input sequence to 1 and positions that are equal to 0 to 0. This is done to mask the padding positions and set their attention scores to 0.
- Create a class called 'EncoderLayer' that inherits from keras.layers.Layer.The constructor '__init__' takes num_heads, feed_forward_dim, and dropout_rate as parameters and saves them as instance variables. The `build` method calculates embedding_dim based on the shape of the input tensor and constructs the multi-head attention, residual connections, and feed-forward neural network layers. Using call to take inputs and a mask and performs the functionality of the Transformer encoder layer using multi-head attention, residual connections, and layer normalization.
- Create a class called ScaledDotProductAttention that inherits from keras.layers.Layer.
- The build method is left empty as no additional parameter initialization is required. The call method takes query, key, value, and mask as inputs and implements the computation process of self-attention, including calculating attention scores, applying the mask, computing attention weights, and generating the output.
- Use the transformer_model function to create the Transformer model. The function takes max_seq_length and word_vectors as parameters. Inside the function, set hyperparameters such as num_heads, feed_forward_dim, num_layers, and dropout_rate. Create the input layer and obtain word embeddings using an embedding layer. Create the masking layer for masking the sequence.Use a loop structure to build multiple EncoderLayer layers, where the output of the previous layer becomes the input of the next layer. Apply global average pooling to the outputs

DOI: 10.54254/2755-2721/48/20241337

of the encoder. Add a fully connected layer and use the sigmoid activation function to obtain the final classification result.

- Predicted.py file is to train word vectors using the Word2Vec model, instantiate the model, encode text, and predict sentiment polarity.
- 'train.py' file is to use the data2inx function to convert text into indexed sequences in a dictionary, use the train_lstm function to optimize the sentiment analysis model with the Adam optimizer, train the model, and evaluate its performance. Finally, output the results.

The Table 2 shows the test results using the SVM model. Since the decision boundary of SVM is determined by support vectors, the author has included code output listing 10 important samples to better explain which samples play a key role in classification. This facilitates a better understanding of the principles of SVM algorithm. Table 3 presents the test results using the transformer model. Due to the addition of word2vec model, nltk data in the overall architecture of the code, it is believed that after a period of algorithm optimization, the use of transformer can be more accurate for spam classification. The evaluation metrics for each model are shown in the Table 4.

Table 2. Performance of SVM

	Precision	Recall	F1-score	Support
Ham(0)	0.99	1.00	0.99	856
Spam(1)	0.99	0.96	0.97	290
Accuracy			0.99	1146
Macro avg	0.99	0.98	0.98	1146
Weighted avg	0.99	0.99	0.99	1146

Table 3. Performance of Transformer

	Precision	Recall	F1-score	Support
Ham(0)	0.98	0.98	0.98	851
Spam(1)	0.94	0.95	0.95	289
Accuracy			0.97	1146
Macro avg	0.96	0.97	0.96	1146
Weighted avg	0.97	0.97	0.97	1146

Table 4. Comparison of 3 models.

	Accuracy	Precision	Recall	F1-score
SVM	0.99	0.99	0.98	0.98
Naive Bayes	0.97	0.98	0.93	0.95
Transformer	0.97	0.96	0.97	0.96

Evidently, the SVM model achieves the highest scores across all three metrics, indicating its superior performance in classification. Considering that the F1-Score taking both Precision and Recall into account, which provides a balanced assessment of the model's accuracy, according to its significance. Consequently, considering the F1-Score, one deduces that Transformer yields more favorable outcomes compared to Naive Bayes. Theoretically, the naive Bayes model is the most suitable and widely used for spam classification systems. One of the important reasons is that the naive Bayes model can handle a large number of features and have good results in processing sparse data, which just fits with the characteristics of a large number of mail data sets [11]. However, the actual conclusion is not the same as the theory. Apart from the samples itself, the author's analysis gave the following reasons. In spam classification, data is usually high-dimensional and sparse, which may cause the curse of dimensionality [12]. There may be noise data, such as misspelling or incorrect labels. SVM utilizes the "kernel trick" method to avoid direct computation of inner products in high-dimensional feature spaces, thereby

DOI: 10.54254/2755-2721/48/20241337

improving computational efficiency. Additionally, by selecting an appropriate kernel function (such as the radial basis function or polynomial kernel) when writing the code, SVM can map data to a higher-dimensional space, thus enhancing classification accuracy. Additionally, to prevent overfitting, a regularization parameter (C) is introduced to control the weight of the margin and the penalty for misclassification.

The Transformer model is capable of handling the dependencies and contextual information in long texts for the task of spam text classification. It utilizes attention mechanisms, including multi-head attention, to effectively capture text features and semantic information. The Word2Vec model is employed to train word embeddings, which convert text into numerical representations for learning and prediction by the model. Preprocessing techniques such as lemmatization, tokenization, and stop word filtering are applied to the input text, further improving the accuracy and reliability of the model. Compared to those two models, the Naive Bayes doesn't get multiple parameters to avoid curse of dimensionality, nor can protect itself from noise data such as misspelling. So, it indicates to the result that performing worse than other two features.

4. Limitations and prospects

In this paper, the author only compares the most general model, i.e., naive Bayes with VM and Transformer. More models for example, CNN, LSTM, random forest is not studied. Hence, the number of the algorithms compared is limited. The conclusions are based on the nature of spam, such as misspelling and homophonic words, and there is still a lack of research on the mathematical reasons for the index differences among these three models. This article can provide a reference for others when studying the problem of spam classification, providing a more accurate model, and solve the spam problem that disturbs the public more effectively.

5. Conclusion

To sum up, this study introduces spam email classification based on SVM, Transformer and Naive Bayes. The study uses these three models to analyse a dataset containing 500 emails. The result shows that SVM has superior performance, and the Transformer trails behind it. They both do better than the Naive Bayes. This paper leaves limitations that algorithms compared is limited and being lack of research on the mathematical reasons among three models. For the future outlooks, this paper may provide a reference to others studying on the algorithm about spam filtering.

In addition to the performance comparison of the three models, this study also provides insights into the features that contribute to spam classification. The most important features include the presence of certain keywords, the frequency of certain characters or symbols, and the length of the email. These findings can be useful for further research on feature engineering for spam classification.

Furthermore, this study also highlights the importance of dataset quality for model performance. The dataset used in this study is relatively small and may not fully represent the diversity of spam emails in the real world. Therefore, future studies may benefit from using larger and more diverse datasets.

Overall, this study contributes to the field of spam email classification by comparing the performance of three popular models and providing insights into feature importance. It also suggests potential areas for future research, such as feature engineering and dataset quality improvement.

References

[1] Emmanuel G D, Joseph S B, Haruna C, et al. 2019 Machine learning for email spam filtering: review, approaches and open research problems.Heliyon vol 5 p e01802.

[2] Wu J 2008 Reduce the harm of network use, improve the efficiency of network application —— On the characteristics of spam and anti-spam technology.Huazhong Architecture vol 26(5) 48-4952,

[3] Awad W A and Elseuofi S M 2011 Machine learning methods for spam e-mail classification.International Journal of Computer Science & Information Technology (IJCSIT) vol 3 p 1.

DOI: 10.54254/2755-2721/48/20241337

[4] Hamed S and Ahmad C 2021 Cloud E- mail Security:An Accurate E- mail Spam Classification Based on EnhancedBinary Differential Evolution Algorithm[J].Computing vol 59(3) pp 5634-5648.
[5] Vinitha H 2019 MapReduce mRMR:RandomForests- Based Email Spam Classification in Distributed Envi-ronment Computing[J].vol 735(3) pp 241-253.
[6] Sumathi P 2021 Cognition based spam mailtext analysis using combined approach of deep neural networkclassifier and random forest[J].Computing vol 12(6) pp 5721-5731
[7] Stehman S V 1997 Selecting and interpreting measures of thematic classification accuracy[J].Remote Sensing of Environment Remote Sensing of Environment vol 62(1) pp 77-89.
[8] Metz C E 1978 Basic principles of ROC analysis[J].Seminars in Nuclear Medicine.vol 8(4) pp 283-298.
[9] Olson D L and Delen D 2008 Advanced data mining techniques, Springer Science & Business Media, Berlin.
[10] Taha A A and Hanbury A 2015 Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool[J].BMC Medical Imaging vol 15 p 29.
[11] Kaiying Z 2021 Study of Chinese spam filtering Based on Improved Naive Bayesian Classification Algorithm[J].Journal of Physics: Conference Series vol 4 p 2083.
[12] Ursula L, Dianne C, Stuart L and Burning S 2022 Reversing the Curse of Dimensionality in the Visualization of High-Dimensional Data[J].Journal of Computational and Graphical Statistics vol 31 p 1