

Spam Classification Based on Machine Learning Algorithm

Yichun Huang*

Data Science, Beijing Normal University-Hong Kong Baptist University United International
College, Zhuhai Guangdong, 519000, China

* Corresponding Author Email: p930026050@mail.uic.edu.cn

Abstract. As technology continues to advance, email is used in almost every field. However, the dramatic increase in the number of spam emails has led to a growing need for accurate and powerful spam classifiers. Unfortunately, spam in its many forms is constantly being updated as the Internet evolves, and the challenge of fighting spam is enormous. With the emerging of deep learning methods, deep based algorithms are widely applied in most of the real-world scene and tasks. In this paper, we used CNN, RNN, LSTM and naïve Bayes models to implement spam filtering, and compared them based on information such as accuracy, model strengths and weaknesses. We test all the models on the public dataset and measure them by four metrics, including accuracy, precision, recall and F1-score. Finally, we conclude that the naïve bayes model achieves the best performance among all those methods, which can deal with the threat of spam efficiently.

Keywords: Machine Learning, Convolution Neural Network, Recurrent Neural Network, Long Short-Term Memory, Naïve Bayes.

1. Introduction

Email, as one of the communication media, is an efficient way for people to communicate with each other on a daily basis. In recent years, the growing number of spam emails has caused a lot of problems for users. The threat posed by spam to users is increasing year on year, with spam accounting for over 77% of global email traffic. Spam wastes unnecessary reading time, takes up storage space in mailboxes, and consumes communication bandwidth and CPU power [1]. This thesis provides a systematic analysis and study of the characteristics of spam. The common classifier used for filtering spam is the plain Bayesian classifier, a model that is a learning method that compares new data with known data and then selects the higher probability result to classify the new data. A probabilistic-based classification technique is Naïve Bayes. We employed Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN) models in this study. Undoubtedly, the convolutional neural network is a feed-forward neural network that does not account for data correlation; rather, the network's output is only connected to its input at the time of computation. Recurrent neural networks are added on top of this. The RNN model is suggested in order to adapt to the processing of time-series data since changes on time series cannot be modeled using the CNN model because many aspects in real life are interconnected. In order to solve the temporal gradient disappearance in RNN, the long- and short-term memory unit LSTM is proposed to achieve the temporal memory function through the opening and closing of gates and to prevent gradient disappearance.

After reading through the papers related to spam classification, I found that some of the papers were reviewing the most popular machine learning methods for this classification and examining the usefulness of these methods for the spam classification problem, from which I could learn that among the six methods, naïve bayes, k-nearest neighbor, support vector machine, Among the six methods, naïve bayes has the highest accuracy rate, k-means has the worst accuracy percentage and naïve bayes has the best performance in terms of recall rate [2]. The majority of studies also employed the naïve bayes approach. For example, when the performance of the naïve bayes algorithm for spam filtering was tested on two datasets, the results were evaluated using the Waikato Environment for Knowledge Analysis (WEKA) tool, and the results revealed that email instances and datasets with fewer attributes can perform better for naïve bayes and achieve higher accuracy [3]. Some studies have proposed simpler and more accurate filtering procedures, such as SpamCop, which treats an email message as

a multi-word set and uses the naive bayes algorithm to determine whether an email is spam or not. the probabilistic approach used in the SpamCop procedure not only improves the accuracy but also serves the keyword search method with its brittleness [4].

In this paper, a comprehensive comparison is made by comparing the accuracy of these four models for classification, model advantages and disadvantages and other information. The data for the experiments were downloaded from https://github.com/wruochao19/Spam_Filter. The experimental findings demonstrate that the main approach of spam categorization, Naïve Bayes, may effectively manage the spam threat.

2. Our Methods

2.1. Naïve Bayes

Naïve Bayes is a classification technique used in machine learning that is based on the Bayes theorem and makes the assumption that features are conditionally independent. The notion is that for a specific goal value, qualities are believed to be conditionally independent of one another. On the basis of the assumption of feature independence, the output joint probability distribution is first learned for a given data set. Based on this model, the predicted variables are then entered, the posterior probability is discovered using Bayes' theorem, and the class with the highest posterior probability is the class to which the variable belongs. Naïve Bayes, so-called because they only make the most basic and fundamental assumptions throughout the formalization, particularly the assumption that each characteristic is equally relevant and that they are all independent.

Naïve bayes is a widely used Bayesian learning method, and in some specific applications its performance can even be compared to, for example, neural network algorithms, but it is far less complex to compute [5].

When determining whether an email is spam, the presence of a word in the message is taken into consideration. The conditional independence of features states that since the presence of one word does not impact the occurrence of another, the likelihood of various words appearing in the email is independent of one another, as are the characteristics of the samples. By calculating the conditional probability of the word vector distribution corresponding to each category, the maximum probability of the category corresponding to that distribution can be calculated and thus the distribution can be estimated. The training data set is known to be:

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} \quad (1)$$

Naïve bayes learn joint probability distributions from training data sets, that means the learning prior probability distributions and conditional probability distributions.

Prior probability distribution:

$$P(Y = c_k), k = 1, 2 \quad (2)$$

Conditional probability distribution:

$$P(X = x|Y = c_k) = P(X^{(1)} = x^{(1)}, \dots, X^{(n)} = x^{(n)}|Y = c_k), k = 1, 2 \quad (3)$$

Since conditional independence is assumed by Naïve Bayes for conditional probability distributions, conditional independence is:

$$P(X = x|Y = c_k) = P(X^{(1)} = x^{(1)}, \dots, X^{(n)} = x^{(n)}|Y = c_k) = \prod_{j=1}^n P(X^{(j)} = x^{(j)}|Y = c_k) \quad (4)$$

The Bayes theorem reveals the following:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \quad (5)$$

The variable x represents the characteristics, whereas the variable y is the class variable.

$$X = (x_1, x_2, \dots, x_n)$$

$$P(y|x_1, \dots, x_n) = \frac{P(x_1|y)P(x_2|y)\dots P(x_n|y)P(y)}{P(x_1)P(x_2)\dots P(x_n)} \quad (6)$$

For the proportionality theorem:

$$P(y|x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i|y) \quad (7)$$

In spam classification, the class variable y is one of two outcomes, healthy mail or spam. Therefore, finding the class variable with the highest probability y :

$$y = \operatorname{argmax}_y P(y) \prod_{i=1}^n P(x_i|y) \quad (8)$$

2.2. Convolutional Neural Network

The CNN model uses convolutional kernels to extract contextual information from the text, as well as the words themselves [6]. The CNN model stems from the use of channels with convolutional steps and the application of pooling, bringing the advantages of a model with spatial invariance and a programmed high light age [7]. A desirable feature of this model is that he retains a two-dimensional spatial orientation in computer vision. Text has an orientation like pictures; text is not two-dimensional, but text has a one-dimensional structure in which the order of the words in the text is important. As a consequence, in this project, a spam filter dimension is selected to match the word vector and modify the region's size. The number of rows in the matrix of filtered sentences that represent words determines the size of the region [8].

In this paper, the model structure is separated into four layers: a word embedding layer, a multi-channel convolutional layer, a multi-channel global maximum pooling layer, and a fully connected layer. The vector associated with each word is changed into an embedding vector with a number of dimensions as part of one of the words embedding layers. Meaning, each sentence is treated as a picture. The multi-channel convolution layer is used to convolve the word embedding transformed sentences using different convolution kernels to generate suitable feature data. For the multichannel global maximum pooling layer, a global maximum pooling is performed on each channel feature output from the multichannel convolution layer. The fully-connected layer then takes the pooled result, i.e. the fully-connected layer, and outputs the number of classifications to obtain the final classification result.

In classification models, the convolution and pooling layers play a major role in feature extraction, wherein primary features are retrieved from lengthy input text sequences using local word order information, and the primary features are then concatenated into high-level features. Convolution and pooling operations can take the place of the standard feature engineering stage in machine learning.

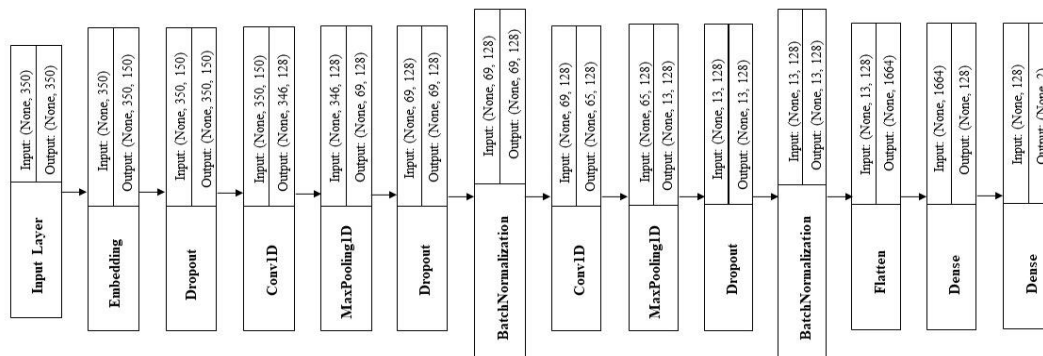


Figure 1. Illustration of our CNN

In this paper, I build a convolutional neural network that first feeds input text data into an embedding layer, extracts feature and downscales the mapping to a word vector. The output is then 'flattened' by one-dimensional convolution and maximum pooling, i.e. the multidimensional input is dimensionless. The output is then passed through a fully connected layer, a Dropout layer to prevent

over-fitting of the model and finally a softmax layer to classify and output the tagged values. Figure 1 above depicts the model's network structure. The model's parameters are as follows: the maximum step size of the pooling layer is set to 5, the convolution kernel of the convolution layer is set to 5*5, the coefficient of the Dropout layer is set to 0.5, the input sequence length is 350, the embedding dimension is 150, and the maximum step size of the convolution layer is 5.

2.3. Recurrent Neural Network

Recurrent Neural Networks (RNN) are networks with a powerful ability to process sequential data. It is an artificial neural network with a "memory" that holds the previous information content needed [9]. RNN perform the same iterative task for each element of the input sequence and store the previous information through hidden states, and then compute the output based on the results of the previous computations. In the principle of the model, RNN can exploit information from hidden states in an infinite sequence, but in practice, moving up from many time steps is more complex [10].

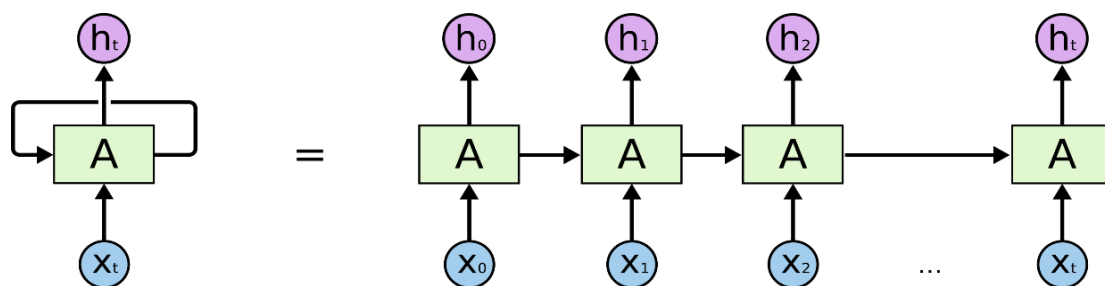


Figure 2. Illustration of RNN

A standard RNN network structure consists of an input layer, a hidden layer, and an output layer, as seen in Figure 2, which shows the structural layout of an RNN. RNNs are different from standard feed-forward neural networks in that they have recurrent neural network units, which, in addition to being connected to the input and output layers, allow data to be passed from one stage of the network to the next. In other words, the RNN model can use earlier data to remember it and affect the output of later nodes because of the network's structure.

The RNN model does, however, have the drawback of gradient disappearance, which implies that the gradient approaches to zero and prevents the network's parameters from being updated. However, the RNN model struggles to learn lengthy text sequences, which is the nature of the issue. Since then, the LSTM model has been suggested as a remedy for this issue.

For the RNN model, I set the number of neuron nodes to 64, the size of the input data to 28, and the size of a sequence of data, so the step size is also 28. Then the output layer is added, the excitation function is softmax, and finally the optimization method is set up, using the categorical_crossentropy loss function and using accuracy as the evaluation metric.

2.4. Long Short-Term Memory

This research uses the Long Short-Term Memory (LSTM) model to enhance the classification of spam's accuracy and feature engineering. It is a modified recurrent neural network that performs various tasks more effectively than conventional networks. LSTM (Long Short-Term Memory) is able to store information in storage units over a huge space of time [3]. In theory, an iterative LSTM system tries to "store" all the information about the network from the beginning to the present by adding layers of activation functions with different purposes, while "dropping" the irrelevant numbers [9].

As shown in Figure 3, the network structure of the LSTM consists mainly of an input gate, an oblivion gate, an output gate and finally an autocycle connection with the neurons. Unlike the RNN model structure where the repeating module contains one layer, the repeating module of the LSTM model contains four interacting layers that are interacting in a very specific way.

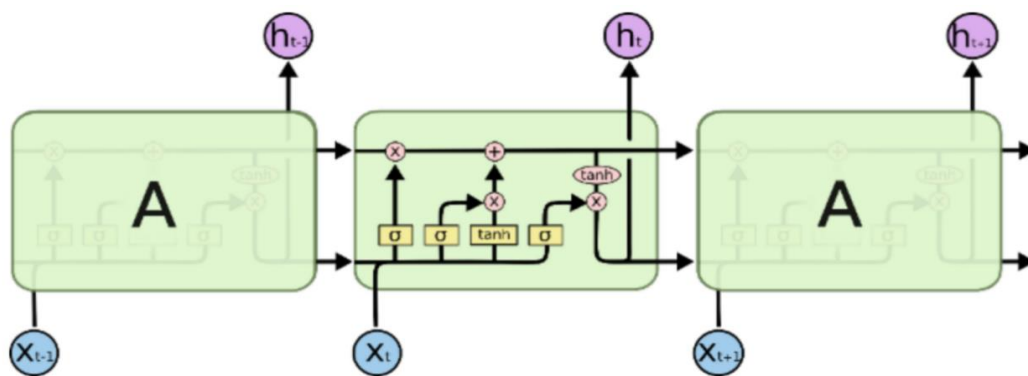


Figure 3. Illustration of LSTM

In the LSTM model the cell states are vectors transmitted at the top of the repetition cell. The cell states only need to undergo some elementary linear transformations and therefore remain invariant easily, while the three gates in the network structure are a structure that allows selective passage of information, namely forgetting, input and output gates, which are used to protect and control the cell states. Thus, the LSTM avoids the problem of long-term dependence by deliberate design, remembering that long-term information is in practice the default behavior of the LSTM.

For the LSTM model, the model uses an LSTM layer connected to a Dropout layer with a hidden layer of dimension 128 and the shape of the input word vector, then a Dense fully connected layer, and finally a softmax layer for classification to output the label values.

3. Results

3.1. Dataset Description

Once the dataset is obtained, it cannot be opened directly due to the encoding format of the original data, so the source data is first converted to utf-8 encoding format when the data is read, and then the characters are filtered. The dataset is then filtered by removing punctuation and more than two spaces, and converting all words to lower case. Furthermore, all stop words such as a, an, the, high frequency prepositions, conjunctions, pronouns etc. were removed. Finally, the content of the emails was split to reduce the prototypes of the English words as far as possible. Our method was implemented in the python language.

The dataset mainly contains the specific content of the emails and the corresponding labels for each email. The details of the dataset are shown in Figure 4.

	v1	v2
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will Ì_b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name
5572 rows × 2 columns		

Figure 4. Illustration of dataset

There are 5572 emails in this dataset, of which 747 are spam, accounting for about 13.41% of the total emails, and the rest are normal emails, as shown in Table 1.

Table 1. Statistic of experimental dataset.

Dataset	Training set	Testing set	Total
Ham	4327	498	4825
Spam	673	74	747

3.2. Overall results

Four alternative models were trained on the same dataset and evaluated on the same test set in this study, and the top performing model was chosen by comparing the accuracy and other assessment metrics of the various models. When compared to other machine learning classifiers like CNNs, RNNs, and LSTMs that are utilized for spam filtering techniques, Naïve Bayes offered better accuracy and performance. Table 2 displays the findings of the comparison of the aforementioned algorithms.

Table 2. Experimental Result

	Accuracy	Precision score	Recall score	F1 score
Naïve Bayes	0.9790	0.9589	0.8860	0.9211
CNN	0.9738	0.9745	0.9738	0.9725
RNN	0.8706	0.7580	0.8706	0.8104
LSTM	0.9073	0.8978	0.9073	0.8965

The Naïve Bayes filter offers the best classification outcomes for the user with the least amount of work and the least amount of work time when it comes to spam detection. The accuracy percentage we were able to obtain was 97.9%, which is a promising outcome. However, it can be inferred from the relatively good performance of Naïve Bayes that our dataset is rather straightforward, and subsequent work might be tested using a larger-scale.

4. Conclusion

This paper examines the main problems with spam classification methods and how to choose a better classifier by comparing various models. In order to categorize the same dataset and compare the outcomes, four different models (Naïve Bayes, CNN, RNN, and LSTM) were used in this study. According on the experimental data, the Naïve Bayes technique appears to be a better option for spam filtering. The trials revealed that the accuracy of Naïve Bayes was 97.90%, that of CNN was 97.38%, and that of the LSTM model was 90.73%, which was significantly greater than the accuracy of the RNN model, which was 87.06%. Based on the findings, it was determined that among the four algorithms, the Naïve Bayes model produced the best outcomes. Although excellent, the findings of this classifier require training on a new dataset. Therefore, the authors' future work will involve using various datasets and testing with more activation functions.

References

- [1] Christina V, Karpagavalli S, Suganya G. Email spam filtering using supervised machine learning techniques [J]. International Journal on Computer Science and Engineering (IJCSSE), 2010, 2(09): 3126-3129.
- [2] Awad W A, ELseuofi S M. Machine learning methods for spam e-mail classification [J]. International Journal of Computer Science & Information Technology (IJCSIT), 2011, 3(1): 173-184.
- [3] Rusland N F, Wahid N, Kasim S, et al. Analysis of Naïve Bayes algorithm for email spam filtering across multiple datasets [C]//IOP conference series: materials science and engineering. IOP Publishing, 2017, 226(1): 012091.

- [4] Pantel P, Lin D. Spamcop: A spam classification & organization program [C]//Proceedings of AAAI-98 Workshop on Learning for Text Categorization. 1998: 95-98.
- [5] Ning B, Junwei W, Feng H. Spam message classification based on the Naïve Bayes classification algorithm [J]. IAENG International Journal of Computer Science, 2019, 46(1): 46-53.
- [6] Zhang W. Spam filter through deep learning and information retrieval [D]. Dissertation, Johns Hopkins University, 2018.
- [7] Metlapalli A C, Muthusamy T, Battula B P. Classification of Social Media Text Spam Using VAE-CNN and LSTM Model [J]. Ingénierie des Systèmes d Inf., 2020, 25(6): 747-753.
- [8] Joshua Kim. Understanding how Convolutional Neural Network (CNN) perform text classification with word embeddings. Tech Report, 2019.
- [9] Gomaa W H. The impact of Deep Learning Techniques on SMS spam filtering [J]. International Journal of Advanced Computer Science and Applications, 2020, 11(1).
- [10] Raj H, Weihong Y, Banbhrani S K, et al. Lstm based short message service (sms) modeling for spam classification [C]//Proceedings of the 2018 International Conference on Machine Learning Technologies. 2018: 76-80.