

Naive Bayesian Spam Filtering

Yuhao Zhu*

Department of computer science, The university of Sheffield, Sheffield, United Kingdom

*Corresponding author: zhuyuhao1111@gmail.com

Abstract. The spam filtering system is used to identify which emails in the received emails are completely meaningless to the recipient and perform operations such as interception and deletion. Nowadays, with the rapid development of the Internet, while e-mail provides convenience for people, spam also comes along with it, which brings many troubles to users. According to statistics, 80% of the emails in the world are spam, and e-spam is really annoying. Therefore, how to solve the problem of filtering emails has important practical significance. Spam filtering using Bayesian theory is a statistical technique applied to email filtering. It essentially uses Bayesian classification to discriminate the attributes of emails, including spam and non-spam. Bayesian-based spam filtering is a very effective technique that can modify the model to meet the needs of specific users and give a lower spam detection rate that is acceptable to users. In this experiment, we use Naive Bayes for experiments, and we use Unigram and bigram methods to preprocess the data, respectively. Finally, it is concluded that the data processing accuracy of unigram and bigram is greater than 0.75, and bigram performs better in four different evaluation indicators.

Keywords: Spam filtering; E-mails; Naive Bayes.

1. Introduction

Spam filtering using Bayesian theory is a statistical technique applied to email filtering. It essentially uses Bayesian classification to discriminate the attributes of emails, including spam and non-spam. Bayesian-based spam filtering is a very effective technique that can modify the model to meet the needs of specific users and give a lower spam detection rate that is acceptable to users. In this experiment, we use Naive Bayes for experiments, and we use Unigram and bigram methods to preprocess the data, respectively. Finally, it is concluded that the data processing accuracy of unigram and bigram is greater than 0.75, and bigram performs better in four different evaluation indicators.

E-mail is a mode of transportation that uses electronic means to exchange information. It is the most widely used service on the Internet. Through e-mail, people can get in touch with people all over the world very quickly. However, this simple and cheap communication tool is very easy. As shown in Figure 1, [1] be the target of scammers. Spam is defined as any email that is forcibly sent into a user's mailbox without the user's permission,[4] in 2020, 306.4 billion emails were sent every day, and in 2023, more than 347.3 billion emails are expected to be sent every day.[2] According to Statista, in September 2020, 47.3% of emails were spam, up from 55% just a year ago. These staggering numbers show how potentially dangerous it is to use this seemingly simple but effective communication tool in Figure 2.[3]

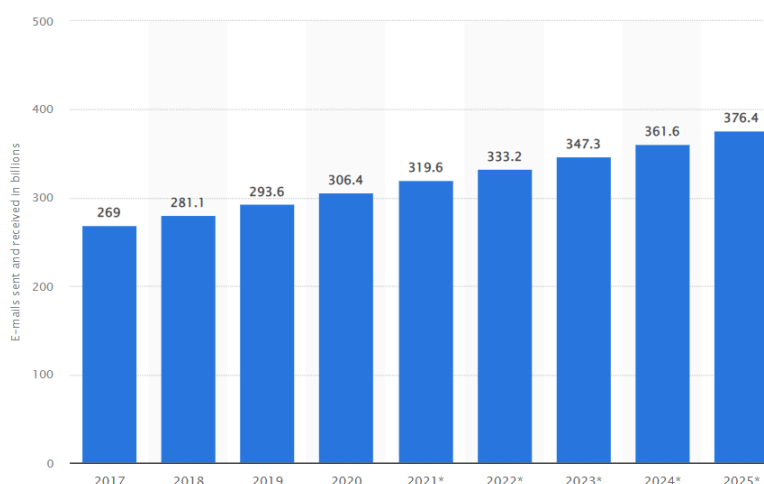


Fig. 1 Number of sent and received E-mails.

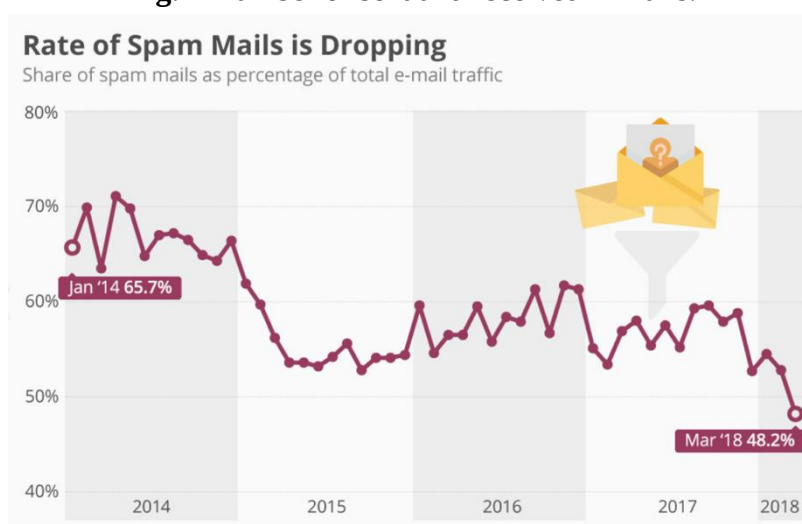


Fig. 2 Rate of Spam Mails.

So far, there are many spam filtering techniques. Decision tree is one of them, [7] a method of supervised learning in which the main idea is top-down divide and conquer. First recurse upwards from the root position, find an attribute that can be divided at the intermediate node, all subsets continue to be recursively divided according to its internal nodes, if these subsets can be correctly classified, then the leaf nodes can be constructed, these subsets also need Classification to the corresponding leaf node, when each subset is classified to the leaf node, the decision tree is formed. Chandrasekar used statistical or data mining methods to formulate precise spam rules in 2018, [5] using a decision tree classifier to classify emails as spam or normal, and finally got a high 99 percent accuracy rate.

Support vector machines are also by far a good method for spam filtering, [6] a supervised learning method that performs binary classification of data, where the decision bound is the maximum margin hyperplane for the sample solution, this hyperplane is used to distinguish whether it is spam or not. The choice of kernels is a key issue when using SVMs because they directly affect the separation of emails in the feature space, this method also has high precision and accuracy.

Naive Bayesian filter is a common method in the field of spam detection and has been widely used in spam filtering. In this paper, we will focus on the techniques and methods of naive Bayesian spam filtering.

2. Method

2.1. Naive bayes

Let's introduce the algorithm principle and implementation of Naive Bayes. [8] For each classified feature and other features, these features must be independent of each other. Bayesian theory is based on a probability that has occurred to infer the probability of another event occurring, formula: $P(A|B) = P(B|A) P(A)/P(B)$.

Among them, A and B are two events, and P(B) cannot be equal to zero. In general, when event B is given to be true, the probability of event A occurrence can be calculated, and P(A) is called as the prior probability of event A, P(A|B) is called the posterior probability of B. When event A and event B are independent of each other: $P(AB)=P(A)P(B)$, The following formula can be obtained by conversion: $P(y|x_1, \dots, x_n) = P(x_1|y) P(x_2|y) \dots P(x_n|y) P(y)/P(x_1) P(x_2) \dots P(x_n)$. where $(x_1, \dots, x_n)$ are dependent features, y is a class variable, and because the denominator is constant related to the input data, the denominator can be removed to get: $P(y|x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i|y)$.

Next, we will introduce how Naive Bayes is applied in spam filtering. Through the above understanding of Bayes, the formula can be transformed into: $P(\text{Spam} | w_1, \dots, w_n) \propto P(\text{Spam}) \cdot \prod_{i=1}^n P(w_i|\text{Spam})$, and $P(w_i|\text{Spam}) = (N_{w_i|\text{Spam}} + \alpha) / (N_{\text{spam}} + \alpha \cdot N_{\text{Vocabulary}})$.

2.2. Implementation details

Machine-learning algorithms use statistical models to classify data, and the trained models determine which categories the emails are likely to belong to by giving them the vocabulary of the message.

2.2.1. Data collection

The data set uses the data in Kaggle, which has two columns, one of which is the text of the email, and the other is the label of the spam email (0 or 1), with a total of 5728 samples. As shown in Figure 3.

0	Subject: naturally irresistible your corporate...	1
1	Subject: the stock trading gunslinger fanny i...	1
2	Subject: unbelievable new homes made easy im ...	1
3	Subject: 4 color printing special request add...	1
4	Subject: do not have money , get software cds ...	1
...	...	...
5723	Subject: re : research and development charges...	0
5724	Subject: re : receipts from visit jim , than...	0
5725	Subject: re : enron case study update wow ! a...	0
5726	Subject: re : interest david , please , call...	0
5727	Subject: news : aurora 5 . 2 update aurora ve...	0

Fig. 3 Example of data

2.2.2. Prepare and analyze data:

Through `emails.groupby('spam').describe()`, we can know that there are 4327 hams and 1368 spam in this dataset in Figure 4.

	count		unique	text	top	freq
spam						
0	4327	4327	Subject: hello guys , i ' m " bugging you " f...		1	
1	1368	1368	Subject: naturally irresistible your corporate...		1	

Fig. 4 Data Preparison

We wanted to understand the relationship between word count and spam, and it turned out that some spam emails had much larger word counts than regular emails. Shown in Figure 5.

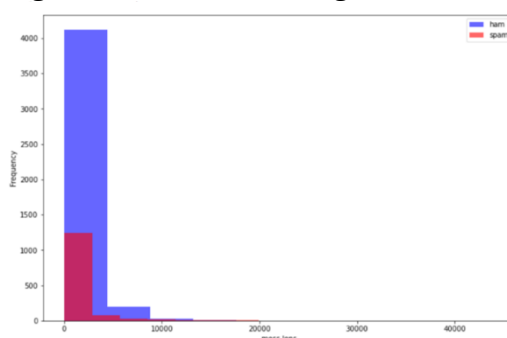


Fig. 5 The correlation between the length and type of E-mail.

Next, the original dataset is divided into training set, test set, training labels and test labels.

Before using the Naive Bayes algorithm, we need to remove punctuation marks, etc., convert all letters to lowercase, and finally split into separate words, parse the text file into term vectors. Where `ngram_range()` represents the upper or lower boundary of different words or characters, text n-grams are often used in natural language processing. It's essentially a string of words appearing in the same window at the same time. When computing n-grams, a word will usually be advanced, and when creating language models, n-grams are used to create not only unary models, but also bigrams and trigrams. we use two forms of unigram and bigram. There are generally three conversion forms, as shown in Figure 6

If the occurrence of a word is independent of its surrounding words, it is called a unigram language model.

If the occurrence of a word depends only on the occurrence of a word before it, it is called a bigram language model.

If the occurrence of a word depends on the two words that appear before it, it is called a trigram language model.

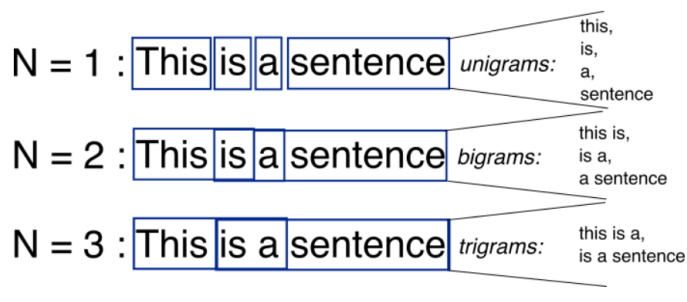


Fig. 6 Example of different variants of Naïve Bayes, such as unigram, bigram and trigram.

2.2.3. Analyze the data:

Calculate the probability that the email is spam and normal:

$$P_{\text{spam}} = \text{spam} / \text{training_set}$$

$$P_{\text{ham}} = \text{ham} / \text{train_set}$$

Create a function to calculate the probability that a word belongs to spam and a word belongs to normal mail, we assume that $\alpha=1$.

$P_word_spam = (\text{len}(\text{spam}[\text{spam}[\text{word}]! = 0]) + 1) / (\text{len}(\text{spam}) + \text{len}(\text{vocabulary}))$

$P_word_ham = (\text{len}(\text{ham}[\text{spam}[\text{word}]! = 0]) + 1) / (\text{len}(\text{ham}) + \text{len}(\text{vocabulary}))$

2.2.4. Training algorithm and testing algorithm

Import the training set, training labels, test set, convert the probabilities P_spam and P_ham calculated above into logarithmic form, traverse the words in each line to calculate the probability of spam and the probability of normal mail, when the probability of spam is greater than the probability of ordinary mail, it is regarded as spam.

3. Results

Table1. The performance of the proposed two variants of Naïve Bayes.

	Accuracy	Precision	Recall	F1-score
unigram	0.75856	1.0	0.07095	0.13249
bigram	0.79368	1.0	0.20608	0.34174

Unigram accuracy is 0.759, bigram accuracy is 0.794. unigram recall is 0.07095, bigram recall is 0.20608 and the value of bigram_f1 is greater than that of unigram_f1. Naive Bayes is a simple yet powerful algorithm ideal for spam filtering. It can be seen that converting sentences into bigram format is more effective and more accurate than converting into unigram format.

It can be seen that the Naive Bayes algorithm is very effective in dealing with small data, and can also deal with multi-class problems. However, it is sensitive to the preparation data of the input method, because it is necessary to ensure that the feature conditions are independent when using Naive Bayes, so it will cause a certain loss of accuracy. Therefore, in order to solve this problem, you can use Laplace smoothing to solve the problem of zero probability, take the logarithm of the result of the product to avoid problems when underflow occurs, and use the natural logarithm to process without data loss.

Like other machine learning algorithms, Naive Bayes cannot understand the language context, and can only judge whether the text belongs to one of the categories by counting the relationship between words. When some terms of spam are replaced by other synonyms, Naive Bayesian algorithm is likely to judge spam as normal mail.

4. Conclusion

This paper mainly studies how to filter spam, and adopts Naive Bayesian algorithm to explain. For both bigram and unigram word segmentation forms, their accuracy rates are greater than 0.75, but bigram performs better in four different evaluation metrics. It can be speculated that for spam filtering, training bigram word vectors has a greater advantage, which can Better reflect the relationship to the previous words, thereby improving accuracy. From this research, we can conclude that Naive Bayes is a very simple but very practical technique for filtering spam well, ensuring that users can quickly find important emails of real value.

References

- [1] Resource from <https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/>
- [2] D. Lynkova, How Many Emails are Sent Per Day: The Startling Truth [2021] TechJury, Sep. 2021, [online] Available: <https://techjury.net/blog/how-many-emails-are-sent-per-day/>.
- [3] Resource from <https://onedrive.live.com/edit.aspx?resid=5F55E6440CEF19A5!2436&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU>

- [4] Kosta, E., Valcke, P. and Stevens, D. (2009) "Spam, spam, spam, spam ... Lovely spam!" Why is Bluespam different?', *International review of law, computers & technology*, 23(1-2), pp. 89–97. doi:10.1080/13600860902742513.
- [5] Chandrasekar, D.C. (2018) 'CLASSIFICATION TECHNIQUES USING SPAM FILTERING EMAIL', *International journal of advanced research in computer science*, 9(2), pp. 402–410. doi:10.26483/ijarcs.v9i2.5571.
- [6] Amayri, O. and Bouguila, N. (2010) 'A study of spam filtering using support vector machines', *The Artificial intelligence review*, 34(1), pp. 73–108. doi:10.1007/s10462-010-9166-x.
- [7] Saifuldeen H Abdulrahman and Mohammad Salim (2022) 'Using Decision Tree Algorithms in Detecting Spam Emails Written in Malay: A Comparison Study', *ITM Web of Conferences*, 42, p. 01001. doi:10.1051/itmconf/20224201001.
- [8] Xu, S. (2018) 'Bayesian Naïve Bayes classifiers to text classification', *Journal of information science*, 44(1), pp. 48–59. doi:10.1177/0165551516677946.
- [9] Resource from N-Grams Definition | DeepAI
- [10] Mishbah Bilgili (2022). Available at:<https://www.kaggle.com/datasets/karthickveerakumar/spam-filter>