Exploration of spams detection
in traditional machine learning algorithms and large language models

Yiyi Wang
LT2314 Language Technology Resources
University of Gothenburg
Email: guswanyie@student.gu.se

Abstract

This study explores the effectiveness of various machine learning algorithms for spam classification, comparing traditional models such as naive Bayes, K-nearest neighbor (KNN), and support vector machines (SVM) with advanced techniques such as deep learning models including multi-layer perceptrons (MLPS) and pre-trained language models such as RoBERTa and SPAM-T5. We used three publicly available datasets: TREC, SMS Spam Collection, and Enron email dataset, applied a Bag of Words (BoW) model for feature extraction, and then trained and evaluated using different classification algorithms. Our analysis shows that for small-scale datasets, naive Bayes methods (specifically multinomial naive Bayes classifiers) achieve the highest performance in terms of accuracy and F1 scores, while deep learning models, such as MLP and pre-trained language models, exhibit superior performance on large-scale and complex datasets. The results also highlight the importance of misclassification costs (TCR), showing that traditional models such as naive Bayes are more cost-effective when computational efficiency is prioritized. In contrast, deep learning models, while computationally expensive, offer the best trade-off between the accuracy and scalability of large data sets. This article provides valuable insights into choosing the right algorithm for the spam sorting task, emphasizing the need to balance performance with computational efficiency and economic cost in real-world applications.

1. Introduction

With the advent of the digital age, spam filtering has become an important task to protect users' mailbox security and improve information security. However, with the rapid development of the Internet, the number and types of spam have exploded, which has brought a lot of troubles to users and enterprises. Traditional spam filtering methods (such as naive Bayes algorithm) have achieved remarkable results in the past research because of their simplicity and efficiency. However, with the advancement of natural language processing technology, algorithms based on large language models (LLMs) are gradually emerging, showing stronger semantic understanding and context processing capabilities. These modern algorithms are able to capture more complex text features through deep learning techniques, thus improving the accuracy and robustness of spam classification.

However, different methods have their advantages and disadvantages in practical applications. Naive Bayes algorithm is still the first choice in many scenarios because of its high computational efficiency and simple implementation. Deep learning algorithms based on large language models, on the other hand, are considered to be more suitable for dealing with rapidly changing forms of spam due to their strong ability to learn complex text patterns. Therefore, comparing the performance of naive Bayes algorithm and large language model in spam filtering can not only provide a new perspective for the research of related fields, but also provide a useful reference for the selection of algorithms in practical applications.

For example, the daily spam received by users comes in many forms, including advertising information from merchants, promotional emails, financial promotion information, and so on. Although many mail clients block spam by setting keywords or sorting algorithms, there are

still fish that slip through the net. Currently commonly used spam classification algorithms include polynomial naive Bayes, Bernoulli naive Bayes, complementary naive Bayes, logistic regression, support vector machine (SVM), K-nearest neighbor (KNN), decision tree, random forest, gradient lifting, neural network (multi-layer perceptron), etc. Studying and comparing the performance of these algorithms in spam sorting tasks can effectively guide the selection of appropriate solutions in practical applications.

Some studies have shown that polynomial naive Bayes classifier is superior to Bernoulli naive Bayes classifier in terms of accuracy and accuracy when classifying English user comments [1]. On the recall rate parameter of positive reviews, polynomial naive Bayes classifiers also show higher scores [2]. Still, there are some unanswered questions in existing research: How well do these algorithms perform in the face of semantically complex spam? Can large language models provide significant benefits? This study aims to further explore these issues.

To this end, this paper will take the following steps: (1) Review the research progress of naive Bayes algorithm and large language model in spam filtering; (2) Construct a benchmark experiment to test and compare the classification performance of various algorithms; (3) Put forward the improvement scheme and analyze its applicability in the actual scenario.

The structure of this paper is as follows: The second part describes the experimental method and data set. The third part shows the experimental results. The fourth part discusses them in detail. The fifth part summarizes the full text and puts forward the future research direction.

2. Materials and methods

2.1 Data Resource

2.1.1 TREC06C

TREC06C is a public spam corpus provided by the International Text Retrieval Conference (TREC), which is widely used in spam filtering research. The corpus is divided into English data set (TREC06P) and Chinese data set (TREC06C), which correspond to spam classification tasks in different language environments respectively. The mail data of TREC06C comes from the real mail communication, and retains the original format of the mail, including the title, body content and metadata (such as sender address, recipient address, etc.), so that it can more truly reflect the characteristics of spam in the actual scene. In addition, the messages in this dataset were rigorously labelled to clearly distinguish between spam and normal mail (ham), providing high-quality supervised learning data for the study. In this study, we selected TREC06P (English data set) as the experimental data source to evaluate the performance of different classification algorithms in English mail spam classification tasks. In this dataset, the total number of emails is 37313. Separately, the number of spam emails is 24542 and the number of normal emails 12771.

2.1.2 SMS Spam Collection

The SMS Spam Collection is a publicly available annotated SMS data set designed specifically for the study of SMS spam filtering on mobile devices. The dataset was collected from multiple sources, including user-contributed and publicly available SMS samples, and was designed to cover a wide range of SMS content formats and linguistic features. The data set contains 5,572 text message samples, in which each text message is clearly labelled as a category (spam or normal text message) to facilitate the training and testing of supervised learning algorithms. Separately, the number of spam SMS is 747 and of normal SMS is 4825.

The content of SMS covers many types of common daily communication, advertising promotion, scam SMS and so on, with good diversity and representation. In this study, the data set is used to verify the performance of the algorithm in short text garbage classification tasks, especially for the detection ability of spam SMS in the environment of mobile devices, and to provide data support for exploring the applicability of the algorithm in different scenarios.

2.1.3 Enron Email Dataset

The Enron Email Dataset (also known as the Enron Corpus) is a well-known public email dataset that was collected in 2002 as part of the investigation into the bankruptcy of Enron Corporation. [3] The dataset, generated by 158 employees and containing more than 600,000 emails, is one of the classic corpora for studying spam classification and natural language processing. A smaller version of this dataset was used in this study, containing a total of 33,716 emails, and each email was clearly labelled as spam and normal mail (ham). This version of the dataset is a balanced dataset with 17,171 spam messages and 16,545 normal messages, with a spam ratio of about 49%. This annotation balance makes it ideal for training and evaluating classification algorithms to more accurately measure the actual effectiveness of algorithms in spam filtering tasks. Through the use of Enron Email Dataset, this research can simulate the mail scene closer to the real business environment, and deeply explore the performance of different algorithms in mail spam classification.

2.2 Methods

This study uses three publicly available Spam datasets: TREC Dataset, SMS Spam Collection dataset and Enron Email Dataset for spam classification experiments under different scenarios. Here is a detailed description of the method:

2.2.1 Data preprocessing

The TREC dataset is provided in file form, where the spam or normal mail (ham) label of each message is indexed in the corresponding file. By writing code, extract the body content of each email, and output the body content and corresponding labels to the file, so as to prepare for the subsequent "bag of words model" conversion. The SMS Spam Collection dataset is a labelled text message dataset where each text message is clearly marked as either a spam message or a normal message. In order to be consistent with other datasets, the contents of this dataset were also preprocessed, including the removal of stop words and the transformation of bag of words model. In the pre-processing process, the mail body is also extracted and stop words are removed to build training and test data sets suitable for subsequent classification tasks.

2.2.2 Dataset partitioning

Each data set is divided into a training set and a test set using the 'train_test_split()' function provided by sklearn. The training set is used to train the model, while the test set is used to evaluate the model performance.

2.2.3 Feature extraction

All data sets were feature extracted using the "Bag of Words Model". The bag of words model is a simple and direct text vectorization method, which converts sentences into vector representations, does not consider the order of words, and only focuses on the frequency of words in the sentence. Set 'stop_words='english' and use the built-in English stop list (such as "a", "the" and other words that are high frequency but ineffective for analysis) to remove these words to reduce noise. Construct the word frequency matrix of the training set using

'fit_transform(x_train)' and generate the word frequency matrix of the test set using 'transform(x_test)' based on the word order of the matrix.

2.2.4 Model training and comparison

On TREC Dataset, SMS Spam Collection dataset, and Enron Email Dataset, the following common classification algorithms were tested and compared:

- Naive Bayes family: Polynomial Naive Bayes, Bernoulli Naive Bayes, complementary naive Bayes.
- Traditional machine learning algorithms: logistic regression, support vector machine (SVM), K-nearest neighbor (KNN), decision tree, random forest.
- Ensemble Learning Method: Gradient Boosting.
- Deep Learning Method: Neural network (Multi-layer Perceptron, MLP).

On data sets of different sizes and characteristics, the performance of these algorithms is compared by experiments, and the difference of their application scenarios and effects is analyzed.

2.2.5Performance evaluation indicators

In order to comprehensively evaluate the spam classification performance of different algorithms, the following five main indicators are selected in this study:

- Precision (P): indicates the classifier's ability to correctly identify spam. The higher the accuracy rate, the lower the probability that normal mail will be mistakenly judged as spam.
- Recall (R): reflects the ability of the classifier to detect spam. The higher the recall rate, the less likely the system will miss spam.
- Accuracy: A comprehensive measure of the overall classification accuracy of spam and normal mail. The formula is Accuracy = (TP + TN)/N, where TP and TN represent the number of correctly classified positive and negative samples respectively, and N is the total number of samples.
- F1 Metric: A harmonic average of accuracy and recall, used to comprehensively evaluate model performance. The higher the F1 value, the better the classification effect of the model.
- Total Cost Ratio (TCR): In practice, the cost of misrepresenting normal mail as spam is usually much higher than the reverse. For this reason, the misjudgment cost ratio λ is defined, assuming that the loss of misjudgment of normal mail is λ times that of misjudgment of spam, for example, λ=9 means that the cost of misjudgment of normal mail is 9 times that of misjudgment of spam. TCR reflects the economic cost performance of classifier in different scenarios.

2.2.6 Experimental purpose and hypothesis

Through the comparison of the above algorithms, the performance of each algorithm in the spam classification task on the data sets of different sizes and characteristics is verified, and the best classification method suitable for different scenarios is explored.

Experimental hypothesis

- Naive Bayes algorithm performs better on small-scale data sets, especially in terms of computational efficiency.
- Neural network method based on deep learning can show higher classification accuracy and robustness when processing large-scale and complex text data (such as Enron Email Dataset).

- The performance of different algorithms is significantly different under the constraint of TCR (misjudgment cost). These hypotheses are verified by experiments, which provides reference for the selection of spam filtering algorithms.
-

3. Results

Trec06c	F1 Score	Precision	Recall	Accuracy	TCR Score
Multinomial Naive Bayes	0.98	0.99	0.96	0.97	12.2
Bernoulli Bayes	0.93	0.88	0.99	0.9	7.2
Complement Bayes	0.98	0.99	0.95	0.97	11.2
KNN Algorithm	0.96	0.92	0.99	0.94	11.2
SVN Algorithm	0.99	0.99	0.99	0.99	48.74
Decision Tree	0.98	0.97	0.99	0.97	24.28
Random Forest	0.99	0.99	0.99	0.99	45.03
Gradient Boosting	0.96	0.93	0.99	0.94	12.49
Netural Network	0.99	0.99	0.99	0.99	75.09

Figure 1 Trec06c dataset

SMS Spam	F1 Score	Precision	Recall	Accuracy	TCR Score
Multinomial Naive Bayes	0.95	0.96	0.93	0.98	6.03
Bernoulli Bayes	0.88	1	0.78	0.97	1.83
Complement Bayes	0.89	0.84	0.95	0.97	4.39
KNN Algorithm	0.78	1	0.65	0.95	0.93
SVN Algorithm	0.92	0.99	0.87	0.98	3.34
Decision Tree	0.88	0.91	0.85	0.96	2.45
Random Forest	0.9	1	0.82	0.97	2.42
Gradient Boosting	0.83	0.99	0.72	0.96	1.29
Netural Network	0.93	0.97	0.9	0.98	4.32

Figure 2 SMS Spam dataset

Enron Corpus	F1 Score	Precision	Recall	Accuracy	TCR Score
Multinomial Naive Bayes	0.98	0.98	0.98	0.98	22.95
Bernoulli Bayes	0.97	0.99	0.94	0.97	8.38
Complement Bayes	0.98	0.98	0.98	0.98	22.95
KNN Algorithm	0.89	0.84	0.95	0.89	4.03
SVN Algorithm	0.98	0.97	0.98	0.98	17.76
Decision Tree	0.94	0.94	0.95	0.95	6.49
Random Forest	0.98	0.98	0.98	0.98	18.27
Gradient Boosting	0.93	0.89	0.98	0.93	7.47
Neural Network	0.98	0.98	0.99	0.98	24.53

Figure 3 Enron Corpus dataset

LLM Model	SMS Spam			Enron Corpus		
	F1 Score	Precision	Recall	F1 Score	Precision	Recall
RoBERTa	0.95	0.97	0.92	0.99	0.99	0.99
SetFit	0.96	0.97	0.95	_	_	_
Spam-T5	0.95	0.87	0.94	0.99	0.99	1.0

Figure 4 Test F1 score, precision, and recall performance of three LLMs for each dataset [4]

By comparing the performance of multiple classification algorithms on different datasets (such as TREC, SMS Spam Collection and Enron Corpus), this study draws the following key findings,

3.1 Performance analysis of Naive Bayes algorithm

Polynomial Naive Bayes algorithm shows excellent classification results in multiple data sets, especially in processing high-dimensional text data and long documents, showing high F1 values and TCR scores, which further verifies the role of its full consideration of word frequency in improving classification performance. The stability of complementary Naive Bayes on unbalanced data sets is also demonstrated, and the robustness of its parameter estimation makes the algorithm close to polynomial Naive Bayes in classification accuracy, but slightly inferior in TCR score. In contrast, Bernoulli Naive Bayes has certain advantages in short text processing and small-scale feature Spaces, but its classification performance is relatively limited as the dataset size increases.

3.2 Tradeoffs of traditional machine learning methods

Support vector machine (SVM) algorithm shows high accuracy and recall rate in multiple data sets, especially when the feature dimension is moderate. However, when the feature dimension is much larger than the sample size, SVM performance begins to be limited. KNN algorithm can reach the level of Naive Bayes in classification accuracy, but its slow classification speed makes it not suitable for tasks requiring high real-time requirements. Decision trees and random forests show strong competitiveness in classification performance, especially reflecting high cost efficiency in TCR scores.

3.3 Advantages and challenges of deep learning methods

Neural network models have the best classification performance on large-scale datasets such as Enron Corpus, with F1 values and TCR scores significantly higher than traditional machine learning methods. This shows that neural networks have strong learning ability and adaptability to complex data. At the same time, algorithms based on pre-trained language models such as RoBERTa and Spam-T5 further improve the classification effect, especially when processing emails with high semantic complexity. However, the shortcomings of deep learning models are also obvious, including reliance on hyperparameter tuning, high training costs, and sensitivity to feature scales.

3.4 Overall comparison and economic analysis

Combined with the analysis results of TCR scores, it can be seen that the actual choice of different algorithms should be weighed according to the specific application scenario. In small data sets that require fast classification, polynomial Naive Bayes is still the preferred solution due to its simplicity and efficiency. In large, unbalanced data sets, complementary Naive Bayes, random forests, and neural networks can provide higher classification accuracy and economic benefits. In addition, the introduction of pre-trained language models provides new possibilities for spam classification tasks, but its high computational overhead may limit its application in resource-constrained environments

4. Discussion

4.1 Comparison of Naive Bayes classifiers

Among Naive Bayes methods, the best performing is polynomial Naive Bayes classifier, followed by complementary Naive Bayes classifier, and the worst performing is Bernoulli Naive Bayes classifier. When using TREC datasets (which contain more samples), the difference in classification effect between Bernoulli Naive Bayes and complementary Naive Bayes is reduced, and the increase in data volume significantly improves the classification performance of Bernoulli Naive Bayes.

Specifically, complementary Naive Bayes is an improvement on the standard polynomial Naive Bayes algorithm, especially for non-balanced data sets. Complementary Naive Bayes

use supplementary statistics for each category to calculate the weight of the model, and their parameter estimates are more stable than polynomial Naive Bayes, and tend to perform better in text classification tasks.

In the comparison of polynomial model and Bernoulli model:

- Bernoulli model: regardless of the frequency of words, it is more suitable to deal with short texts; When the number of features is small, the performance is better and the implementation is simpler, which can reduce the impact of extreme data (such as certain words appearing frequently in some documents and missing completely in others).
- Polynomial model: fully considering the influence of word frequency, it is more suitable for processing long text and high-dimensional feature data. When the frequency of a word is higher, that word is also more representative, so the classification effect of the polynomial model is superior on large-scale data sets, but may be interfered with by extreme data.

4.2 Performance analysis of SVM and KNN

In the spam filtering task, a large number of experiments show that linear kernel SVM shows an excellent balance between performance and computational complexity. Linear kernel is suitable for proper feature representation and can achieve better classification performance. However, SVM performance is limited when the feature dimension is much larger than the sample size. In addition, when the sample size is large and the kernel mapping dimension is high, the applicability of SVM is low.

KNN is one of the most commonly used case-based classification methods, and its intuitiveness makes it often achieve good results in text classification tasks. Since KNN does not need training process, its classification process directly compares the text to be classified with all the texts in the training set, and judges the classification result by the category of the most similar k texts. However, KNN classification is slow and is not suitable for tasks such as spam filtering, which requires high classification speed. Nevertheless, in experiments for research purposes, KNN is still used in the field of spam filtering. The results show that when the sample set is large, the performance of KNN is similar to that of Naive Bayes, but the speed disadvantage is significant.

4.3 Performance of neural networks

The classification effect of neural network on large-scale data sets is better than that of traditional methods. For example, on the Enron dataset, the multi-layer perceptron (MLP) updates the model through incremental learning, allowing users to label new samples to further improve classification accuracy. This capability makes neural networks suitable for dynamic spam filtering scenarios.

However, MLP also has the following shortcomings:

- Non-convex loss function problem: the existence of hidden layers makes multiple local minima unavoidable, and different random weight initializations may lead to different classification accuracy.
- Hyperparameter tuning: There are many hyperparameters that need to be adjusted, including the number of hidden neurons, the number of layers and the number of iterations.
- Feature scaling sensitivity: The neural network is very sensitive to the scaling of input features, and different scaling strategies can significantly affect the result.

5. Conclusions and further work

Combining the performance of different algorithms, we can draw the following conclusions:

- Small data sets: On small-scale data sets, polynomial Naive Bayes has faster classification speed and higher classification accuracy, so it is a better choice.
- Large data sets: On large and complex data sets, such as the Enron dataset, neural networks can achieve higher classification accuracy and robustness thanks to their nonlinear modeling capabilities and incremental learning properties.
- Cost-sensitive scenarios: In scenarios with high misjudgment costs (such as TCR = 9), proper adjustment of feature selection and algorithm hyperparameters can significantly improve classification effects and reduce actual application costs.
- These findings suggest that LLMs, and specifically Spam-T5, could be a valuable tool for addressing the ongoing challenges in spam detection [4].

The above results confirm the research hypothesis:

- Naive Bayes algorithms are suitable for small data sets, especially for scenarios that require fast training.
- Neural networks have obvious advantages in processing large and complex text data.
- The performance difference of different algorithms under TCR constraints provides a reference for practical application.

For future work, to deploy LLMs in spam detection applications and improve the efficiency of models, future work will need to focus on reducing the computational requirements of these models.

References

[1] N. L. Octaviani, E. Hari Rachmawanto, C. A. Sari and I. M. S. De Rosal, "Comparison of Multinomial Naïve Bayes Classifier, Support Vector Machine, and Recurrent Neural Network to Classify Email Spams," 2020

[2] S. Rastogi, R. Sambyal, P. Tyagi and R. Kushwaha, "Multinomial Naive Bayes Classification

[3] Metsis, V., Androutsopoulos, I., Paliouras, G.: Spam filtering with naive bayes – which naive bayes? (2006), http://citeseer.ist.psu.edu/757874.html

[4] Labonne, M., & Moran, S.J. (2023). Spam-T5: Benchmarking Large Language Models for Few-Shot Email Spam Detection. ArXiv, abs/2304.01238.