# SMS Spam Classification with DistilBERT

Huang Hankun
*Department of Economics*
*University of Michigan*
Ann Arbor, the United States
hankun@umich.edu

***Abstract*—This report presents a text classification pipeline leveraging the DistilBERT architecture on the SMS Spam dataset. The project demonstrates data preprocessing, fine-tuning of a pretrained language model, and evaluation of model performance. The model achieved a final accuracy of 98.2% on the test set, showcasing the effectiveness of transformer-based models in spam detection tasks.**

## I. Introduction

Spam detection is a critical task in text classification, with practical importance in messaging platforms and email systems. The SMS Spam dataset provides a collection of short messages labeled as "ham" or "spam". This project applies the transformer-based model DistilBERT to classify SMS messages and evaluates its performance on a sampled version of the dataset.

While traditional methods such as Naive Bayes or Support Vector Machines have been widely used for spam detection, the emergence of transformer models like BERT and DistilBERT has shown significant performance gains by leveraging deep contextual understanding of text.

## II. Method

### A. Dataset Description

The SMS Spam dataset was obtained from HuggingFace Datasets. It contains labeled SMS messages. We split the dataset into training and test sets using an 80/20 ratio, and sampled 1/5 of the data to speed up training.

- Original train size: 2230
- Original test size: 558
- Reduced train size: 446
- Reduced test size: 112

The data is relatively balanced and consists of short text messages with informal language, which makes it suitable for testing the robustness of NLP models on real-world text classification tasks.

### B. Data Preprocessing

The messages were tokenized using the distilbert-base-uncased tokenizer. Padding and truncation were applied to ensure a fixed-length input suitable for the model. The dataset was mapped using the tokenizer in batch mode.

All messages were converted into lowercase with attention masks applied. The maximum token length was restricted to the model's input limit (512 tokens), although most messages were significantly shorter.

### C. Model Architecture

We use the DistilBertForSequenceClassification model with two output labels. The model is initialized with pre-trained weights from distilbert-base-uncased, with a classification head added on top. The architecture is a lightweight variant of BERT, optimized for faster training and inference while preserving performance.

The classification head consists of a dropout layer followed by a dense layer projecting to the output logits.

### D. Training Setup

The model was trained for 3 epochs with the AdamW optimizer and a learning rate of $2 \times 10^{-5}$. The evaluation was performed in 50 steps. The table below summarizes the training hyperparameters:

TABLE I
Training Hyperparameters

| Hyperparameter | Value |
|---|---|
| Learning Rate | $2 \times 10^{-5}$ |
| Epochs | 3 |
| Training Batch Size | 16 |
| Evaluation Batch Size | 16 |
| Weight Decay | 0.01 |
| Logging Steps | 50 |
| Eval Steps | 50 |
| Save Steps | 50 |

The loss function used was cross-entropy loss, suitable for binary classification tasks. Optimization was performed using AdamW, which combines the benefits of Adam with weight decay for better generalization.

## III. Results

### A. Performance Metrics

Model performance was evaluated on the test set. The following table summarizes accuracy, precision, recall, and F1 score:

TABLE II
Performance Metrics of the DistilBERT Model on the SMS Spam Dataset

| Metric | Score |
|---|---|
| Accuracy | 0.9821 |
| Precision | 0.90 |
| Recall | 0.90 |
| F1 Score | 0.90 |

The progression of training loss during training is as follows:

TABLE III
TRAINING LOSS PROGRESSION ACROSS STEPS

| Step | Training Loss |
|---|---|
| 50 | 0.2031 |
| 100 | 0.0392 |
| 150 | 0.0185 |

These results indicate that the model was able to quickly converge with low loss and high evaluation performance within a small number of steps. The consistent decrease in loss also suggests effective learning with minimal overfitting.

### *B. Confusion Matrix*

The confusion matrix below illustrates the distribution of predictions:

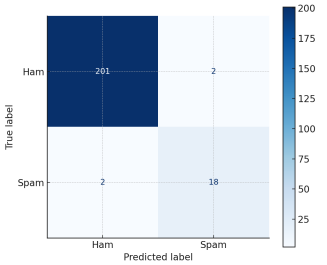


Fig. 1. Confusion matrix for SMS Spam classification

From this matrix:

- True Positives (TP): 18
- True Negatives (TN): 201
- False Positives (FP): 2
- False Negatives (FN): 2

This corresponds to a strong classification boundary with minimal misclassification. Precision and recall remain high, indicating robust performance across both classes. The low number of false positives is especially important in spam detection to avoid misclassifying legitimate messages.

## IV. CONCLUSION

This project demonstrates the application of DistilBERT to SMS spam classification. Despite training on only $20\%$ of the original dataset, the model achieves high performance with more precision than $98\%$. The combination of modern NLP architectures and simple preprocessing steps can yield excellent results on standard text classification benchmarks.

The model not only achieved high accuracy, but also maintained balanced precision and recall. This balance is essential in spam detection, where both false positives and false negatives have consequences.

Future work may include:

- Using the full dataset to further improve performance.
- Comparing with traditional models (e.g., Naive Bayes, SVM).
- Visualizing attention weights to interpret model predictions.
- Deploying the model in a live spam detection service.
- Exploring more advanced transformer architectures like RoBERTa or BERTweet for text classification.

This study confirms that transformer-based models like DistilBERT are highly effective even in low-resource training scenarios. The methodology outlined can serve as a template for similar classification tasks in natural language processing.

## ACKNOWLEDGMENT

The author would like to thank the instructor and course staff for their guidance throughout the project. The HuggingFace community and associated documentation were also invaluable in facilitating the use of pretrained models and datasets.