Artificial intelligence-based spam filtering using a neuro-linguistic approach with PyTorch framework

Balázs Tóth
John von Neumann Faculty of Informatics
Óbuda University
Budapest, Hungary
mwzx0d@stud.uni-obuda.hu

Valéria Póser
John von Neumann Faculty of Informatics
Óbuda University
Budapest, Hungary
poser.valeria@nik.uni-obuda.hu

Szandra Anna Laczi
John von Neumann Faculty of Informatics
Óbuda University
Budapest, Hungary
laczi.szandra@nik.uni-obuda.hu

***Abstract*—The paper presents the development of a PyTorch-based artificial intelligence spam filter based on neuro-linguistic approaches, i.e. natural language processing (NLP). A model has been developed to easily filter out messages that appear suspicious.**

***Keywords*—PyTorch, model, development, NLP, spam**

I. Introduction

Digital communication has become almost indispensable in people's daily lives. Unfortunately, spam is growing exponentially at the same time, challenging the systems that filter out unwanted content. It is critical that the software we use to send and receive message filters them reliably.

This paper shows how the PyTorch framework and natural language processing approaches can be used in concert to design an intelligent spam filtering system. It is able to recognise if the given data is general or suspicious message.

A. Typical patterns in email spam

- Phishing emails are designed to impersonate a trusted organisation and lure recipients into revealing sensitive information such as usernames, passwords or any valuable data.
- Malware emails send attachments or links to malicious software designed to infect the recipient's device with viruses, ransomware or other malicious programs.
- The advance payment scam, these emails promise large sums of money in exchange for a small advance.
- Fake lottery or prize scams, which falsely claim that recipients have won a lottery prize and often ask for personal details or payment to claim the alleged prize.
- Survey emails designed to collect personal data for fraudulent purposes.

These are the most common types of email spam but the list could be endless.

II. Target of the project

There can be found various of spam types, especially e-mail, SMS spamming, social media spamming and others. The project focuses on e-mail messages spam precisely in terms of data.

III. Why PyTorch?

In 2016, Facebook's AI research group, now known as Meta, took the lead in creating the PyTorch framework and generously shared it with the global community as an open-source resource. PyTorch has gained recognition for its outstanding qualities, being praised for its exceptional simplicity, impressive flexibility, and inherent efficiency. These remarkable features have solidified PyTorch's position as a fundamental and highly regarded tool in the fields of artificial intelligence and machine learning.

TABLE I: Comparing PyTorch with Keras

Category	PyTorch	Keras
API Level	Low	High
Datasets	Large datasets, high-performance	Smaller datasets
Debugging	Good debugging capabilities	Challenging
Pretrained models	Yes	Yes
Speed	Fast, high-performance	Slow, low-performance
Written in	Lua,	Python
Visualization	Limited	Depends on backend

Table 1 provides a detailed comparison between the PyTorch and Keras frameworks. [1] The key factor influencing the choice of PyTorch is its impressive performance and the ability to handle large datasets seamlessly. This pivotal decision is grounded in the framework's robust capabilities, making it a reliable choice for our study.

IV. Overview of spam statistics

As discussed previously about what are the common patterns in email spams, it is crucial to know about the statistics too. These statistical values shows how the companies and end users are not having the good amount of knowledge how to handle potential harmful messages in their inbox if that is not handled by the spam filter nor how their personal data is valuable for the hackers.

TABLE II: Spam statistics [2]

Threat actors	In 2023, 32% of threat actors used email as a pathway to disrupt organisations.
Grow	The total number of business and consumer emails sent and received daily will exceed US$333 billion in 2022 and is forecast to grow to over US$392 billion by year-end 2026.
Global value	In 2022, nearly 49% of all e-mails globally were identified as spam, up from 46% in 2021.
Origin	The United States of America currently leads as the country of origin of spam emails with 8,765 spam emails sent.
Report	In 2023, 18 million emails were reported by the State of the Phish organisations over 12 months.
Most common form	Phishing is the most common form of cybercrime, with an estimated 3.4 billion spam emails sent every day.
Accounts	Spam accounts for 14.5 million messages globally per day. This makes up 45% of all emails according to research.
Compromise	Scams and fraud comprise only 2.5% of all spam emails, however, identity theft makes up 73% of this figure.
Malware threats	In 2022, the number of unknown malware threats spiked to 3.8 million, indicating a substantial 46% surge according to Trend.
Spam blocks	According to Google, Gmail blocks more than 100 million spam emails per day.

V. Methodology

A. Hyperparameters

B. Data collection and preprocessing

C. Model architecture

$$input\_dim = \sum_{j=1}^{m} \begin{cases} 1 & \text{if feature}_j \text{ is in the vocabulary} \\ 0 & \text{otherwise} \end{cases}$$

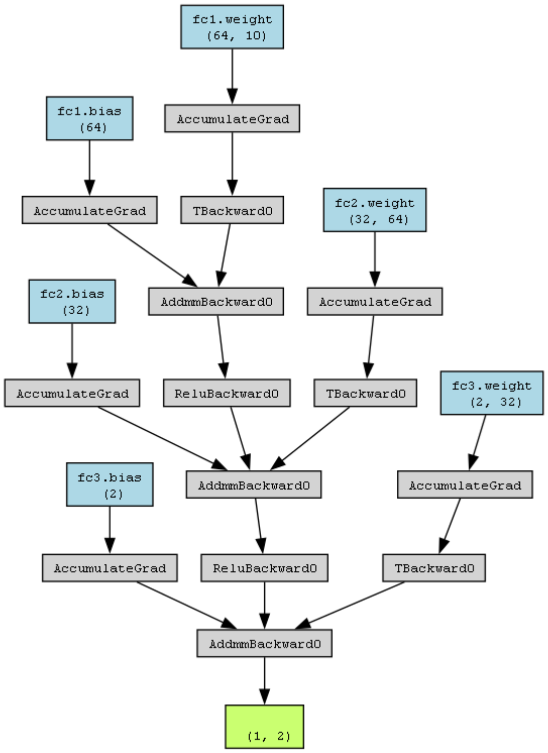
Sum of ones and zeros, where each 1 indicates the presence of unique feature in the vocabulary.

```
    class TextClassifier(nn.Module):
        def __init__(self, input_dim):
            super(TextClassifier, self).
                __init__()
            self.fc1 = nn.Linear(input_dim,
                64)
            self.fc2 = nn.Linear(64, 32)
            self.fc3 = nn.Linear(32, 2)

        def forward(self, x):
            x = torch.relu(self.fc1(x))
            x = torch.relu(self.fc2(x))
            x = self.fc3(x)
            return x
```

Listing 1: Modell Python kód tartalma

Fig. 1: Visualization of the model



D. Training process

True Positives (TP):

- Cases that were actually positive (spam) and correctly predicted by the filter to be positive (spam).
- Example: An email containing known spam keywords and characteristics is correctly classified as spam by the filter.

True Negatives (TN):

- Cases that were indeed negative (not spam) and the filter correctly flagged them as negative (not spam).
- Example: A regular non-spam email, without any spam-like attributes, is correctly identified as not spam by the filter.

False Positives (FP):

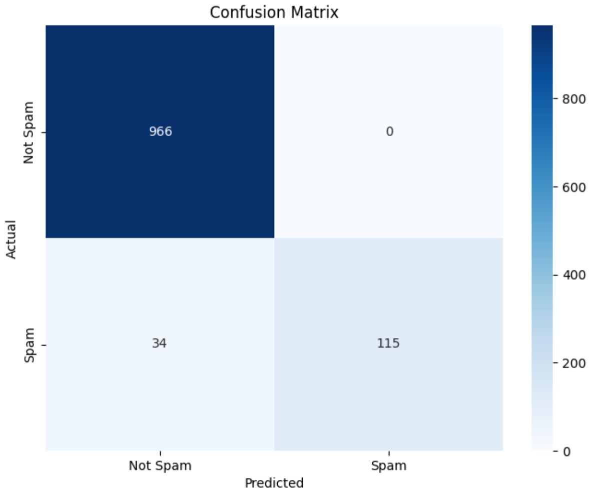
- Cases that were in fact negative (not spam) but were wrongly flagged by the filter as positive (spam).

- Example: A legitimate email from a friend contains certain words or patterns that the spam filter misclassifies as spam.

False Negatives (FN):

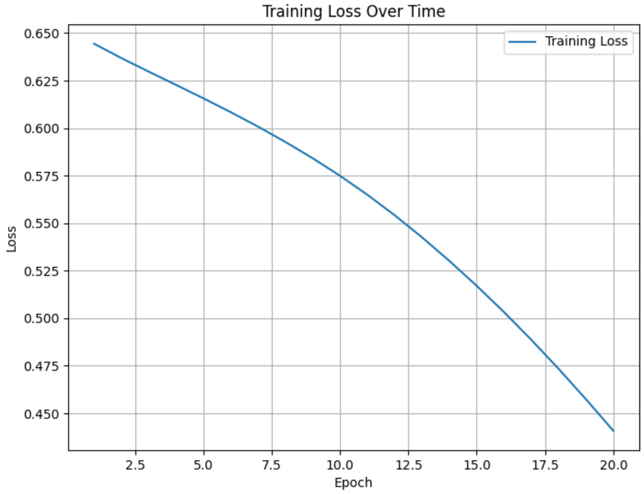
- Cases that were actually positive (spam) but the filter incorrectly flagged them as negative (not spam).
- Example: An actual spam email manages to evade detection and is incorrectly classified as a non-spam email.

Fig. 2: Confusion Matrix



The confusion matrix provides a visual representation of these variables, making the representation of these values clearer.

Fig. 3: Training loss over time



E. Evaluation metrics

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

Accuracy measures the overall correctness of the model by calculating the proportion of correctly predicted cases (true positives and true negatives) relative to all cases.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision is the ratio of correctly predicted positive observations (true positives) to the total number of predicted positives. It focuses on how many of the predicted positive cases were actually positive.

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN}$$

The recall calculates the ratio of correctly predicted positive observations (true positives) to the total number of true positives. This shows how well the model captures all positive cases.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 score is the harmonic mean of accuracy and recall. It balances accuracy and recall and provides a single metric for evaluating model performance. It is particularly useful when the distribution of classes is uneven.

References

[1] *PyTorch vs Tensorflow vs Keras*, https://www.datacamp.com/tutorial/pytorch-vs-tensorflow-vs-keras, Last viewed; 20:59, 6th of December, 2023

[2] *Spam statistics: a deep dive into unwanted emails*, https://eftsure.com/statistics/spam-statistics/, Last viewed; 18:34, 6th of December, 2023

[3] Basemah Alshemali and Jugal Kalita, *Improving the Reliability of Deep Neural Networks in NLP: A Review*, Knowledge-Based Systems, 2020.

[4] Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, Opeyemi Emmanuel Ajibuwa, *Machine learning for email spam filtering: review, approaches and open research problems*, Heliyon, 2019.