CERN Zenodo - Adaptable Spam Filter Modelling

Luka Secilmis, Yanis De Busschere, Thomas Ecabert
Department of Computer Science, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

***Abstract*—Zenodo is a digital scientific research repository operated by CERN. The platform is widely used by researchers and scientists as it enables the sharing, preservation and citation of research papers, data, and software at a very large scale. As a result of its popularity and growing scale, Zenodo has been a target for spam content since its launch in 2013. Their current ML-based spam detection models, however, are not performant enough, often requiring a significant amount of resources and time for manual checking. In this paper, we present the development of a production-ready text-based spam classifier achieving 98.8% accuracy along with a 98.2 F1-score by fine-tuning BERT, the state-of-the-art NLP model by Google, for spam classification.**

I. Introduction

Zenodo is a digital scientific research repository operated by the European Organization for Nuclear Research (CERN). The platform is widely used by researchers and scientists as it enables the sharing, preservation and citation of research papers, data, and software at a very large scale. Since its launch in 2013, Zenodo houses over two million records and a petabyte of data, serving 15 million user visits from around the world annually [1]. It is further committed to collect and store 100s of petabytes of Large Hadron Collider data as it grows over the next 20 years [2].

As a result of its popularity and growing scale, Zenodo is a target for spam content such as advertisements and streaming content. Spam is highly undesirable for a digital repository as it wastes a significant amount of resources such as storage, bandwidth, and processing power. This can lead to decreased performance of the data centre. Furthermore, the integrity and reputation of Zenodo as a research-oriented repository can be compromised through holding a lot of spam content, which can particularly perturb user experience when querying the repository for scientific material.

Thus, the filtering of spam has been of paramount importance to Zenodo, which they have attempted to automize through the use of machine learning models trained on the large amounts of spam content they have filtered over the years. Their current ML models, however, are not performant enough, often requiring a significant amount of resources and time for manual checking.

In this paper, we present the development of a text-based spam classifier achieving 98.8% accuracy along with a 98.2 F1-score by fine-tuning BERT - the state-of-the-art NLP model by Google - for spam classification. First, an exploratory data analysis is conducted. We then introduce BERT and the technique of fine-tuning - a machine learning paradigm known as transfer learning, and outline our model's methodology. Then, feature engineering and the training process of our model are discussed before evaluating its performance on the test set and comparing it to Zenodo's current spam classifier model. Finally, the development of a production-ready adaptable spam filtering model which Zenodo will deploy as a service in the near future is presented.

II. Data Overview & Analysis

In this section, we provide an overview and analysis of the data used to train and evaluate our spam classifier. The data consists of Zenodo's published open access records metadata [3], including entries marked as spam and deleted by the Zenodo staff.

The dataset contains the metadata of 1,722,305 records, each containing a variety of fields, including information about the record's creators, contributors, and keywords, as well as a description and a boolean value indicating whether the record was marked as spam by Zenodo staff. While most of the records were hand checked, some were classified using Zenodo's current ML models, leading to a some false positives and false negatives in the given dataset.

Of these 1,722,305 records, 37,784 were marked as spam and 1,684,521 were marked as non-spam (also known as "ham"). Having a large proportion of ham records relative to spam records can be challenging when building a spam detection model. To address this issue and avoid overfitting on the ham class, we decided to sub-sample the ham class - which we will further explain in the following sections.

In this data analysis, from manually investigating many spam and ham records, we decided to focus mainly on the descriptions - as these very clearly indicated whether a record is spam or ham to the human eye.

In the context of this project, the distinction between ham and spam is very strong as spam records are simply any text that does not qualify as scientific research. The Zenodo staff made it clear to us that most spam came in the form of advertisements or uploaded movies. We postulated that with the recent advancements in Natural Language Processing (NLP) we would be able to build a robust model that is able to make a clear semantic and contextual difference between a scientific text and one that is not.

III. Background Knowledge & Methodology

In this section we present background knowledge on BERT and fine-tuning, tying it together with spam classification, in order to motivate our use of it and give a general overview of our methodology in the following sections. Furthermore, we will outline the three main constraints set to us by the CERN Zenodo team, and how these impacted our final model choice.

A. Pre-trained BERT model

BERT (Bidirectional Encoder Representations from Transformers), introduced by Devlin and colleagues from Google in their 2018 paper [4], is one of the most revolutionary breakthroughs in the field of NLP. Given an input text, a pre-trained BERT model outputs embeddings which are dense fixed-length vector representations of words. These word embeddings capture the meaning and context of the words in the input text, and they can be used as input to downstream NLP tasks. BERT was trained to learn embeddings through Masked Language Modelling (MLM) and Next Sentence Prediction (NSP) on the entirety of Wikipedia and a very large corpus of literature. It is bidirectional and context-based allowing us to gain a rich understanding of language and its context - a task traditional NLP models often struggle with.

B. Motivation: BERT for spam classification

BERT's ability to capture rich contextual information about the words in a given text is what motivated us to develop our spam classifier based on it. We postulated that BERT would allow us to truly make a robust semantic difference between spam and ham, and thus lead to improved performance and a more effective spam filter. Rather than training a new ML model from scratch by feeding it BERT embeddings paired with a corresponding label - which would require more computational resources and time - we employ a technique called fine-tuning.

C. Transfer Learning: fine-tuning BERT

Transfer learning is a machine learning technique in which a model that has been trained on one task is fine-tuned for a different task. It is a powerful technique allowing us to leverage the pre-trained general BERT model, and specialize it in the task of binary spam classification.

We add an additional layer on top of the BERT transformer architecture, thus using the knowledge and features learned by the pre-trained model as a starting point, and adjusting the model's weights to optimize its performance on the specific task during fine-tuning.

D. Requirements & model choice

For this project, we were given three main constraints:

- Multilingual model - ability to classify spam in multiple languages.
- Adaptability - ability to re-train the model to reflect new trends in research and spam content.
- Fast & light-weight - ability to re-train the model quickly, make predictions quickly. Ability to store the model easily (not too large).

In order to respect these requirements, and after a lot of research, we chose to fine-tune the **DistilBERT base multilingual (cased)** pre-trained model.

The multilingual version of BERT (mBERT-base) was trained on the concatenation of Wikipedia in 104 different languages. Its distilled version (DistilmBERT) is composed of 6 layers, 768 dimension and 134M parameters (compared to 177M parameters for mBERT-base) and thus significantly lighter. This is achieved through what is called a distillation process, in which a small student model (DistilmBERT) is trained to imitate the outputs of a larger teacher model (mBERT). During training, the student model is optimized to minimize the difference between its outputs and the outputs of the teacher model on a given input. This process allows the student model to learn from the knowledge and capabilities of the teacher model, while being more efficient, 60% faster to train and retaining 97% of its language understanding capabilities [5].

Because mBERT is pre-trained on a large dataset of text in multiple languages, it can capture general linguistic patterns and knowledge that may be shared

across languages, which can improve the performance of the classification model.

IV. Feature Engineering

To prepare the data for training, we read-in the data in chunks of size 100 thousand. We first select only the "description" and "spam" columns from the dataset and drops any rows that contain missing values. Next, we separate the data into two subsets: one for spam descriptions and one for ham descriptions. The "description" column in each subset is then cleaned by removing any HTML tags using the BeautifulSoup library [6] and the "spam" column is mapped to a binary label.

Finally, we sub-sample the ham class such that there is a 2:1 ratio of ham:spam in the processed dataset. This is done to avoid significant bias and overfitting on the ham class, while still prioritising it in order to avoid discarding valuable information that will help our BERT model learn the nuances and context of what constitutes scientific research text.

V. Training

In order to fine-tune our DistilmBERT model for spam classification, we first load the processed data and split 80% of it into training and 20% into test data. We apply stratification in order to preserve our 2:1 ratio of ham to spam labels in both sets. We then use DistilmBERT's pre-trained tokenizer which splits text into smaller pieces called tokens that can be processed by BERT. This process splits words into sub-words, adds special tokens at the beginning and end of each input sequence, and helps BERT handle out-of-vocabulary words by processing them as combinations of sub-words. Finally, we set-up and use Hugging Face's Trainer class to follow the standard fine-tuning recipe which has been widely adopted by the ML research community and fine-tune the chosen pre-trained DistilmBERT model for text classification.

Fine-tuning required a significant amount of computational resources and time - which we facilitated by renting an NVIDIA RTX A5000 GPU. For training we chose a learning rate (ADAM) of 5e-05, number of epochs equal to 3, and a batch-size of 8. These choices were in-line with the authors' recommendations [4] and also the default choices deployed by Hugging Face [7]. Indeed, the authors themselves observed that large data sets with more than 100 thousand training entries are far less sensitive to hyper-parameter choice [4]. This was encouraging as tuning hyper-parameters and conducting cross-validation was not feasible within the scope of this project's computational limitations.

VI. Evaluation on test data

Below we present the performance of DistilmBERT on the test set. We then compare it to Zenodo's most recent spam classifier based on the Extra Trees model. Note positive labels refer to spam, negative to ham.

Fine-tuned DistilmBERT
Time: 1h09 training - 0.005s prediction

Accuracy	F1	TP	TN
98.8	98.2	99.0	98.3

We clearly see that indeed, as postulated, the fine-tuned DistilmBERT model is a robust spam classifier with excellent evaluation metrics. The model is very accurate, paired with high metrics in terms of F1 score, true positives and true negatives, which clearly indicates that we are not overfitting the ham labels.

2020 Zenodo classifier [8]

Classifier	Accuracy	TP	TN
Extra Trees Classifier	99.8	84.9	99.9

Zenodo's current spam classifier has a higher accuracy of 99.8% compared to our 98.8%. However, we find that their classifier has a true positive rate of 84.9%, which is remarkably smaller than the 99.0% of our model. Their true negative rate of 99.9% suggests that the ham class is significantly overfitted, which is not the case with the spam classifier developed in this paper.

VII. Developing a production-ready adaptable spam filter model

Now that we have trained and evaluated a highly performant spam classifier, the next step is to develop a production-ready model that can be easily integrated into Zenodo's existing infrastructure and deployed as a service.

In the research phase, we used common tools such as Python notebooks to quickly prototype and test ideas. However, research code in a Python notebook is not always suitable for production use, as it may not be optimized for efficiency and scalability, or may not be structured in a way that is easy to integrate into an existing infrastructure. To develop a production-ready model, we refactored and optimized the code, and implemented appropriate testing and deployment processes. We divided the code into steps, automated the process of finding relevant files, implemented security checks

between steps, and added a logging system that records each step of the process to a file.

One of the other key challenges in developing and maintaining an effective spam classifier, as outlined to us by the Zenodo team, is the need to retrain the model on a regular basis. This is because spam content is constantly evolving, and the model needs to be updated in order to continue accurately detecting and filtering out spam. To make the process of retraining the model as easy and efficient as possible, we used the GNU Make system. This allowed us to define a series of dependencies and rules for building the model. For training the model, a single command is needed: "make train". This makes it easy for the Zenodo team to retrain the model on a regular basis and ensure that the model is always up-to-date and performing at its best.

The code we developed is now ready to be integrated into a Docker container, and deployed as a service by the Zenodo team.

VIII. Conclusion

In conclusion, this paper presents the development of a highly accurate and robust text-based spam classifier using state-of-the-art NLP techniques. By fine-tuning the pre-trained multilingual DistilBERT model, we were able to achieve excellent performance, with an accuracy of 98.8% and an F1-score of 98.2. This marks a significant improvement over Zenodo's current spam classifier, which was prone to overfitting the "ham" class. Our work concluded with the development of a production-ready adaptable spam filtering model that will be deployed as a service by Zenodo in the near future.

References

[1] Zenodo, "Faq," https://help.zenodo.org, 2022.

[2] ——, "Blog," https://blog.zenodo.org, 2022.

[3] Z. team, "Zenodo Open Metadata snapshot - Training dataset for records and communities classifier building," Dec. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7438358

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2019. [Online]. Available: https://arxiv.org/abs/1910.01108

[6] L. Richardson, "Beautiful soup documentation," *April*, 2007.

[7] H. Face, "Trainer." [Online]. Available: https://huggingface.co/docs/transformers/main_classes/trainer

[8] A. Ioannidis, "Zenodo classifier model spam detection record." [Online]. Available: https://github.com/zenodo/zenodo-classifier/blob/7d8acf80b4868e5fe03e6cf6ee8d809002fd08f7/model_spam_detection_record.ipynb