Automated Contextual Spam Categorization Tool

LAKHANI Sunil Harsh
20910249
The Hong Kong University of Science and Technology
hslakhani@connect.ust.hk

LI Ka Ho
20922151
The Hong Kong University of Science and Technology
khlibg@connect.ust.hk

TSANG Kai Ho
20905476
The Hong Kong University of Science and Technology
khtsangak@connect.ust.hk

Abstract

In this project, we address the problem of spam detection in comments, messages, and emails, a critical issue prevalent across various social media platforms. We curated a dataset of 6000 labeled instances by combining multiple sources of spam messages and emails. We employ supervised learning using text classification models, including BERT-base, BERT-finetuned, RoBERTa-base and RoBERTa-finetuned, to classify messages as spam or not. Subsequently, we apply unsupervised clustering techniques, such as Hierarchical clustering, K-means, and DBSCAN, to group similar spam messages. We then leverage a generative model to assign labels to each cluster, identifying the type of scam. Our aim and approach demonstrate the potential to detect spam and categorize it into different scam types, facilitating better content moderation on various online platforms.

1 Introduction

Spam messages, comments, and emails have become a pervasive issue in the digital era, polluting online communication channels and causing inconvenience for users. This problem has been emphasized by key figures such as Elon Musk, highlighting the widespread prevalence of spam across various social media platforms. Beyond being a mere nuisance, spam often serves as a medium for scams, including identity theft, malware, financial fraud, and other potentially detrimental effects, thus posing significant risks to unsuspecting individuals. As these unwanted communications continue to evolve, the need for robust and efficient spam detection systems has become more pressing.

This report presents a novel approach to spam detection and categorization by combining supervised learning techniques, unsupervised clustering, and generative models, as illustrated in Figure 1. We first use text classification models, such as BERT and RoBERTa, to perform supervised learning on a curated dataset of 6000 instances. Afterward, we employ unsupervised clustering algorithms, including Hierarchical clustering, K-means, and DBSCAN, to group similar spam messages. Finally, we utilize a generative model to assign labels to each cluster, identifying the type of spam.

By implementing this approach, we aim to not only improve content moderation and provide a cleaner online environment for users but also to protect individuals from the harmful consequences of spam-related scams.

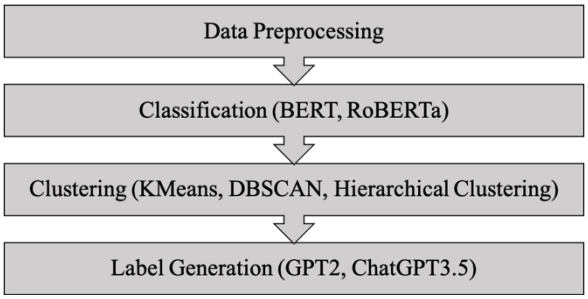

Figure 1: Project flow

2 Related Work

2.1 Text Classification Models

2.1.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a powerful pre-trained language model introduced by Devlin et al. in 2018 [1]. BERT is based on the Transformer architecture proposed by Vaswani et al. in 2017 [2]. It leverages a bidirectional mechanism to understand the context from both left and right sides of a given input, providing rich contextualized word representations. BERT has achieved state-of-the-art performance in various NLP tasks, including text classification, due to its ability to capture complex language patterns.

2.1.2 RoBERTa

RoBERTa (Robustly optimized BERT approach) is an optimized version of BERT, introduced by Liu

et al. in 2019 [3]. RoBERTa modifies BERT's pre-training process, using larger mini batches, more data, and dynamic masking, which allows it to learn deeper contextual representations. This results in better performance in various NLP tasks, including text classification, as demonstrated by its improved benchmark scores.

2.2 Clustering Models

2.2.1 Hierarchical Clustering

Hierarchical clustering [4] is an unsupervised learning method that constructs a tree-like structure (dendrogram) to represent data relationships. The method can be either agglomerative or divisive, depending on whether it follows a bottom-up or top-down approach, respectively. The key advantage of hierarchical clustering is its ability to provide insights into the data hierarchy, making it suitable for grouping similar instances in our spam detection problem.

2.2.2 DBSCAN Clustering

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is an unsupervised clustering algorithm proposed by Ester et al. in 1996 [5]. DBSCAN identifies clusters based on the density of data points, grouping closely packed points together and treating sparse regions as noise. It is particularly effective in handling noisy datasets and discovering clusters with arbitrary shapes

2.2.3 K-means Clustering

K-means is a popular centroid-based clustering algorithm introduced by MacQueen in 1967 [6]. It aims to partition a dataset into K clusters by iteratively updating cluster centroids until convergence is achieved. K-means is simple, scalable, and efficient, making it suitable for various clustering tasks. However, it is sensitive to initial centroid positions and assumes spherical-shaped clusters, which can be limitations in some scenarios.

2.3 Generative Language Models

Generative language models, such as GPT-2 and GPT-3.5, introduced by OpenAI [7] [8] are pretrained models designed to generate coherent and contextually relevant text based on a given input. These models are built upon the Transformer architecture and trained on massive amounts of text data, allowing them to capture intricate language patterns. In our project, we leverage a generative language model to assign labels to spam clusters, identifying the type of scam and providing additional insights into the nature of the spam content.

3 Data

3.1 Datasets

Initially, we aimed to perform spam classification and categorization based on purely social media content. However, due to the scarcity of annotated datasets online, we could only find a single dataset for YouTube comments with about 350 instances, which was not sufficient for our purposes. As a result, we decided to train our model on text messages and email spam content instead. We collected and combined two publicly available datasets for text message spam and email spam content. However, the classes were highly imbalanced, leading us to curate a more balanced dataset using seven different datasets. Our final dataset comprises two columns: “text” and “pam”, where “text” represents the actual content, and “spam” indicates the class of the content, with 1 for spam and 0 for ham. The dataset contains a total of 6000 instances, evenly split between 3000 spam and 3000 ham instances. It is important to note that the YouTube comments dataset was not used in the training process but reserved for testing our model.

Dataset	URL	Type
SMS Spam Collection	https://www.kaggle.com/datasets/thedevastator/sms-spam-collection-a-more-diverse-dataset	SMS
Filtering mobile phone spam	https://www.kaggle.com/datasets/patil4444/filtering-mobile-phone-spam	SMS
Text Spam Dataset	https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset	SMS
Email Spam Dataset	https://www.kaggle.com/datasets/nitishabharathi/email-spam-dataset?select=enronSpamSubset.csv	Email
Spam Mails Dataset	https://www.kaggle.com/datasets/venky73/spam-mails-dataset	Email
Email Spam Dataset	https://www.kaggle.com/datasets/maharshipa	Email

(Extended)	ndya/email-spam-dataset-extended	
Spam Classification for Basic NLP	https://www.kaggle.com/datasets/chandramoulinaidu/spam-classification-for-basic-nlp	Email
YouTube Spam Comments	https://www.kaggle.com/datasets/lakshmi25npathi/images	Social Media

Table 1: Descriptions of dataset

3.2 Data Preprocessing

Before training our models, we performed several preprocessing steps to clean and normalize the data. Our custom script included the following text cleaning functions: removal of links, email addresses, special characters, HTML tags, hashtags, punctuation, non-ascii characters, and stop words. Moreover, it also sets the text to lowercase. These steps ensured that our dataset was free of noise, irrelevant content, and inconsistencies, allowing our models to focus on learning the relevant patterns for spam detection

4 Methodology

4.1 Supervised Learning

In the supervised learning part of the project, we train and evaluate four text classification models, including BERT, BERT fine-tuned, RoBERTa, and RoBERTa fine-tuned, on our dataset. We then select the best performing model for detecting spam content. To accomplish this, we used pre-trained BERT-base and RoBERTa-base models and experimented with fine-tuning them to better adapt to our specific task of binary spam classification.

4.1.1 BERT

BERT is a pre-trained Transformer-based model known for its capability to capture contextual information from both directions of a given text, making it highly suitable for various NLP tasks, including text classification. We used the 'bert-base-uncased' version of the model, which is trained on lower-cased text. We tokenized our text data using the BERT tokenizer, padding and truncating the input sequences to a fixed maximum length. The tokenization process also involved adding special tokens, returning attention masks, and token type IDs. Additionally, we specified the number of labels in order for the model to output only two labels, representing spam and non-spam instances.

For fine-tuning, we added several layers to the base model, including dense layers with varying units and activation functions, dropout layers for regularization, and a final dense layer with a sigmoid activation for binary classification. We trained the BERT fine-tuned model for five epochs, using the Adam optimizer with a learning rate of 5e-5 and binary cross-entropy loss.

4.1.2 RoBERTa

RoBERTa (Robustly Optimized BERT Pretraining Approach) is another pre-trained Transformer-based model, developed by Facebook. It is an optimized version of BERT that addresses some of its limitations. We used the 'roberta-base' version of the model for our experiments. Similar to BERT, we tokenized our text data using the RoBERTa tokenizer, applying padding, truncation, and special tokens. Additionally, we specified the number of labels in order for the model to output only two labels, representing spam and non-spam instances.

The fine-tuning process for RoBERTa was analogous to BERT fine-tuning, with the same additional layers and hyperparameters. We also trained the RoBERTa fine-tuned model for five epochs, using the Adam optimizer with a learning rate of 5e-5 and binary cross-entropy loss. We made sure the training and evaluation process is the same for all models so that it is a fair comparison.

4.2 Unsupervised Learning

In the unsupervised learning part of the project, we train three clustering models, including K-means clustering, Hierarchical clustering, and DBSCAN clustering, with different parameters on our spam contents to find the groups of similar contexts. However, as the number of types of the spam messages is unknown, it is difficult to determine the best model based on the clustering results. This leads to the use of generative models in Section 4.3.

4.2.1 Hierarchical Clustering

Hierarchical clustering is a clustering analysis method to build hierarchies of clusters using either agglomerative or divisive approach [9], which is known as “bottom-up” or “top-down” approach. Since we also use k-means, which has more efficient heuristics to split the data, in our

unsupervised learning, we experiment Hierarchical clustering with the "bottom-up" approach only. Initially, we used Word2Vec to find word embeddings for the corpus of our spam messages and form a vector for each message. After vectorization, we selected the pair of clusters to be merged based on the Ward's minimum variance method at each iteration. By minimizing the total variance of the merged clusters, i.e., the squared Euclidean distance between each point in the cluster, a hierarchy of clusters can be obtained.

4.2.2 DBSCAN Clustering

DBSCAN clustering is a density-based clustering algorithm to group the data points that are densely close to each other, defined by *epsilon* and *minPoints*. Data points that fall on the circle of radius epsilon are categorized into one group. Similar to Hierarchical clustering, we initially used Word2Vec to find the word embeddings and vectors for each spam message. After vectorization, we computed the average distance between each message and its k-nearest neighbours [10] and plotted the distance against the number of spam messages to find the optimal epsilon for training the DBSCAN model

4.2.3 K-means Clustering

K-means clustering is a well-known clustering algorithm to partition data into *k* clusters iteratively. Prior to applying the k-means algorithm to our spam content, we obtained the word embeddings and vectors of each spam message by TI-IDF. After finding the clusters, we then computed the inertia of different numbers of *k* and plotted the distance against the number of our spam messages to find the optimal *k*.

4.3 Generative Models for Labelling

To determine the most suitable clustering model for our spam dataset, we employed two generative models and designed a prompt to generate a label for each cluster. As the k-means algorithm was insufficient in identifying the optimal number of clusters, together with hierarchical clustering, we experimented with four clustering models. While we initially tested GPT-2 on various numbers of k-means clusters, the results were unsatisfactory, prompting us to exclusively use ChatGPT-3.5 for comparison with hierarchical clustering. Ultimately, we chose the optimal clustering model for our spam dataset through a voting system based on whether the labels accurately represented the clusters. Moreover, after experimenting with all of the clustering algorithms, we came to a conclusion that the K-means clustering results were more balanced, resulting in us selecting it to be a part of the final pipeline.

5 Experiments

5.1 Classification Models

We employ a 70-30 train-test split for evaluating the performance of our models. The evaluation metrics used include precision, recall, F1-score, and accuracy, and the results are shown in Table 2. These metrics helped us compare the performance of the four models and select the best one for our spam detection task.

	BERT (base)	**BERT (fine-tuned)**	**RoBERTa (base)**	**RoBERTa (fine-tuned)**
Train Loss	0.0802	0.0598	0.0158	0.0208
Validation Loss	0.0999	0.1108	0.1462	0.1357
Validation Accuracy	0.9703	0.9661	0.9703	0.9686
F-1 Score	0.9703	0.9661	0.9700	0.9661
Recall	0.9703	0.9661	0.9700	0.9661
Precision	0.9704	0.9663	0.9703	0.9663

Table 2: Empirical results of classification models

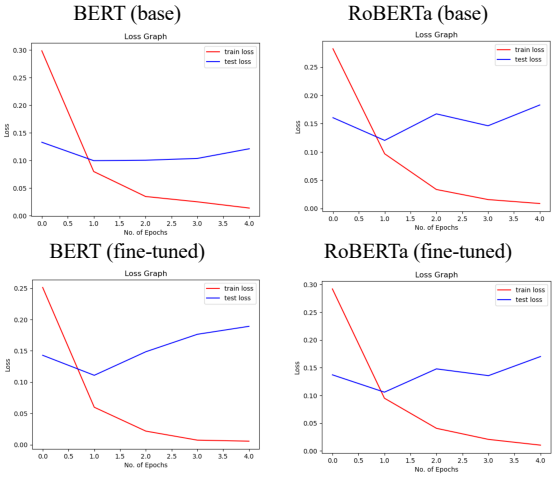

Figure 2: Loss graphs of classification models

The results from our experiments on training spam classifiers show that each model we trained experienced a drastic drop in training loss during

the first epoch, followed by a steady decrease in loss until it reached a plateau. We trained each model for five epochs to evaluate their performance and compare them against each other.

From Table 2 and Figure 2, it is clear that the best performing model is the BERT (base) model, with an impressive accuracy of almost 97%. Although the other models we trained were not far behind, we ultimately decided to use BERT (base) as our spam classifier in the final pipeline.

The BERT (base) model outperformed other models and fine-tuning experiments we conducted, suggesting that it is highly effective at detecting spam content. Its success can be attributed to its advanced natural language processing capabilities, which allows it to understand the meaning and context of the content accurately. Overall, we are confident that the BERT (base) model will be an effective spam classifier for the final pipeline.

To test the efficacy of our tool, we created customized spam inputs and manually tested the model. We found that the model performed exceptionally well, correctly categorizing spam content and clustering them into the appropriate category with high accuracy.

To further validate our model, we tested it on the YouTube comments dataset. The dataset is challenging to classify since it contains a wide range of content, some of which are not strictly spam but could be considered as such. Nevertheless, our tool achieved an accuracy of approximately 77% of correctly categorizing the content as spam. We consider this a significant achievement, given the complexity of the dataset and the challenges it poses.

5.2 Unsupervised Learning

5.2.1 Hierarchical Clustering

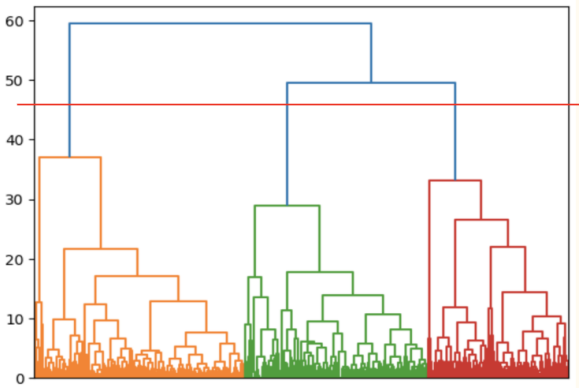

Figure 3: Dendrogram of Hierarchical clustering

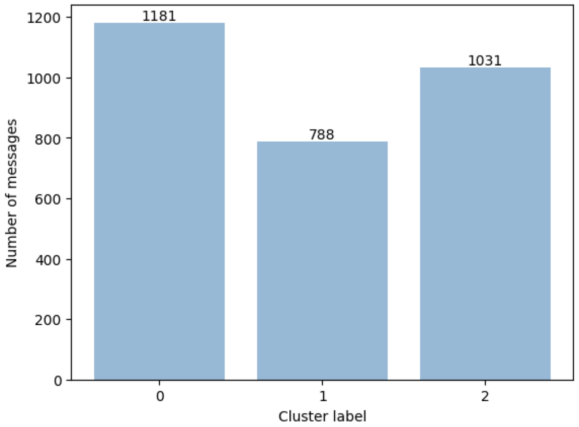

Figure 4: Distribution of clusters by Hierarchical clustering

Using the Hierarchical clustering algorithm, we generated a cluster hierarchy and represented the results using a dendrogram as depicted in Figure 3. The dendrogram indicates that the spam dataset can be divided into three clusters based on the maximum distance observed among them. Figure 4 also illustrates that the number of spam content is evenly distributed across the cluster. Nevertheless, it is crucial to note that the evenness of the distribution does not offer us any significant insights into the algorithm's effectiveness.

5.2.2 DBSCAN Clustering

The DBSCAN clustering algorithm has the ability to automatically determine the appropriate number of clusters based on the values assigned to its two parameters: *minPoints* and *epsilon*. In our case, we set *minPoints* to be 200 and calculated the average distance between each message and its 200 nearest neighbours to generate the Elbow curve, shown in Figure 5. The curve reaches its maximum curvature at a distance of 40, indicating that an optimal epsilon value of 40 should be used. Despite training the DBSCAN algorithm, we only obtained a single cluster in our spam dataset as shown in Figure 6. However, our goal was to classify spam messages into different categories, such as financial scams and malware warnings. As a result, we concluded that v did not meet the objectives of our project and, therefore, we decided to exclude it from our further analysis.

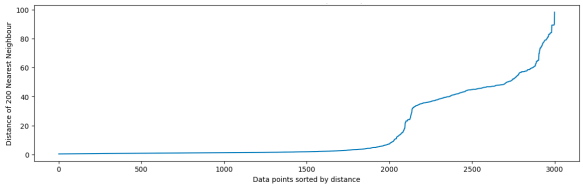

Figure 5: Elbow method for optimal eps for DBSCAN clustering

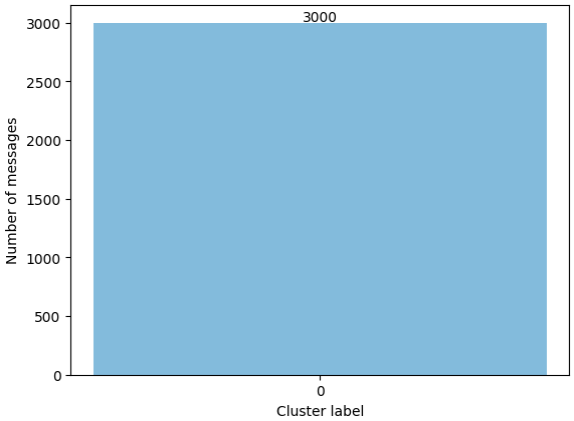

Figure 6: Distribution of cluster by DBSCAN clustering

5.2.3 K-means Clustering

K-means clustering differs from DBSCAN in that it requires the number of clusters k to be predefined prior to training the model. To determine the optimal number of clusters, we generated an Elbow curve, plotting the decrease in the sum of the squared distance between each spam message and its centroid (inertia) for a range of k values, as illustrated in Figure 7. The curve shows diminishing returns when k equals 7, indicating that the cost of splitting our dataset into more than seven clusters is no longer justified. However, we considered seven clusters to be a bit excessive, and instead opted to set the optimal number of clusters to be between three, four, and five. As shown in Figure 8, each number of clusters has a dominant cluster containing approximately two-thirds of the spam messages. This emphasizes the importance of identifying the labels of the small clusters as they may be easily overlooked due to their lower occurrence rate, which leads us to our third task in this project - label generation.

5.3 Label Generation

To generate labels for each cluster, we utilized GPT-2 and ChatGPT-3.5 and designed a prompt for the task.

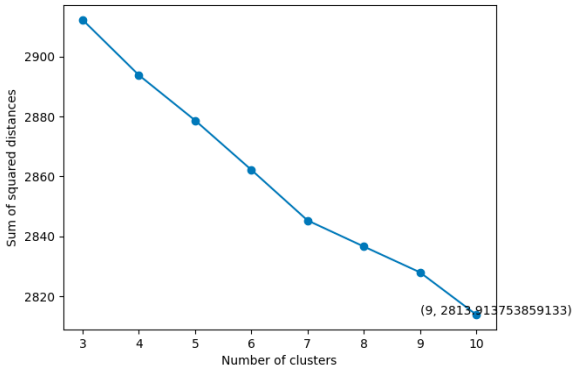

Figure 7: Elbow curve for K-means clustering

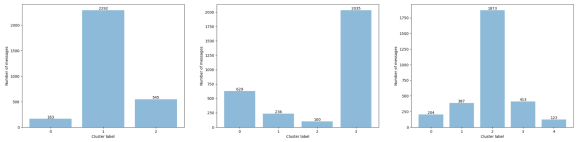

Figure 8: Distributions of clusters by K-means clustering

5.3.1 GPT-2

When experimenting with GPT-2 on the clusters generated by K-means, we discovered that it has a limited maximum text length that allows us to include only up to 50 words per cluster in the prompt, which may limit the generalizability of the clusters. In addition, as illustrated in Table 3, we encountered issues with GPT-2's interpretation of the prompt as it did not generate labels for the clusters but instead continued writing the prompt (k=3) or provided a summary of the entire prompt (k=4 or 5). For k=5, it even produced an incomplete response. Given GPT-2's unsatisfactory performance, we did not use it to test the clusters obtained through Hierarchical clustering.

Number of clusters = 3
todays money 10 day call to date phone to day phone call call call call
Number of clusters = 4
The above list of clusters can also help to determine which words or sentences
Number of clusters = 5
clusters 1-4 are available in a single page format, with the

Table 3: Responses generated by GPT2 for K-means clusters

5.3.2 ChatGPT-3.5

We conducted experiments with ChatGPT-3.5 on the clusters generated by K-means and Hierarchical algorithms using the same prompt. The generated labels are shown in Table 4 and Table 5, and all are reasonable and specific.

Compared to GPT-2, ChatGPT-3.5 exhibits a more powerful comprehension ability and provides precise responses according to the prompt.

Number of clusters = 3
Cluster 0: Prize/Winning Claims and Offers Cluster 1: Marketing Lists and Communication Cluster 2: Mobile Phone Promotions and Services
Number of clusters = 4
Cluster 0: Mobile phone promotions and services Cluster 1: Winning prizes and offers Cluster 2: Information, numbers, and codes Cluster 3: Email and online marketing and services
Number of clusters = 5
Cluster 0: Prize Claim and Winning Notification Cluster 1: Mobile Text and Tone Services Cluster 2: Email Marketing and Information Lists Cluster 3: Holiday and Travel Offers Cluster 4: Award and Complimentary Collection

Table 4: Label generated by GPT3.5 for K-means clusters

Number of clusters = 3
Cluster 0: Business and Financial Information Cluster 1: Email Marketing and Communication Cluster 2: Sales and Promotions

Table 5: Label generated by GPT3.5 for Hierarchical clusters

To compare the performance of K-means and Hierarchical clustering, we examined the labels for the three clusters obtained by the two algorithms. We observed that the labels for each cluster are similar in nature; however, the labels for K-means clusters are more precise and informative. For example, while the label for the first cluster obtained by Hierarchical clustering is "Business and Financial Information", the label for the first cluster obtained by K-means is "Prize/Winning Claims and Offers", which is more specific and informative. This precise labeling can help individuals become more aware of the potential dangers of the spam content. Considering the preciseness, informativeness, and comprehensiveness of the labels, we believe that K-means clustering with five clusters is the best among the three algorithms.

6 Future Work

In the future, we plan to extend our automated contextual spam categorization tool to effectively handle spam and scam content in social media comments. Since there is a lack of annotated datasets for social media spam and scam content, we intend to explore the creation of synthetic data generated using generative models to augment our existing training data. Training and experimenting with fine-tuning different models specifically for social media comments will allow us to better understand the nuances of spam in these platforms and improve the overall performance of our spam classifier.

Another avenue for future work is to adapt our tool to various contexts and industries, making it a versatile solution that can be tailored to suit specific spam content types and use cases. This would further enhance its applicability and impact on reducing the amount of unwanted spam content that users encounter in their daily online activities.

By continuously improving and expanding the scope of our automated contextual spam categorization tool, we aim to make a substantial contribution to spam detection and filtering, ultimately providing users with a safer and more enjoyable online experience.

7 Conclusion

We are pleased to announce that we have successfully integrated our trained spam classifier, clustering algorithm, and generative model to create an automated contextual spam categorization tool. This tool takes an input, classifies it as either spam or legitimate, and if it is spam, categorizes it into the appropriate category.

Our spam categorization tool is a significant step forward in automated spam detection and filtering. It has enormous potential in reducing the number of unwanted spam content that users come across daily, particularly in environments where spam content can be particularly prevalent. Additionally, our tool can be easily customized to suit specific contexts and spam content types, making it a versatile solution that can be applied to a wide range of use cases.

In conclusion, we are confident that our automated contextual spam categorization tool will have a significant impact on reducing the amount of spam content users come across, improving their experience and reducing the risk of phishing attacks. We look forward to seeing its impact in various contexts and industries.

Acknowledgments

We collaborated throughout the entire project from data collection to data preprocessing, from model

implementation to model training, and from presentation preparation to report writing. We, therefore, view each team member's contribution as equal and significant.

References

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30, 5998-6008.

[3] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.

[4] Zhang Z, Murtagh F, Van Poucke S, Lin S, Lan P. Hierarchical cluster analysis in clinical research with heterogeneous study population: highlighting its visualization with R. Ann Transl Med. 2017 Feb;5(4):75. doi: 10.21037/atm.2017.02.05. PMID: 28275620; PMCID: PMC5337204.

[5] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), 226-231.

[6] MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1(14), 281-297.

[7] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. OpenAI Blog, 1(8).

[8] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Sutskever, I. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33.

[9] Teng, L., Amin, R., ElSayed, M. (2022). "An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement," *Journal of King Saud University - Computer and Information Sciences*, 22-Apr-2022.

[10] A. Sharma and A. Sharma, "KNN-DBSCAN: Using k-nearest neighbor information for parameter-free density-based clustering," *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT),* Kerala, India, 2017, pp. 787-792, doi: 10.1109/ICICICT1.2017.8342664.