SpamResponder: Automatic Response System for Voice Phishing

Stanford CS224N Custom Project

Chae Young Lee
Stanford University
chae@stanford.edu

John Gunderson
Roblox
jgunderson@roblox.com

Abstract

We present SpamResponder that defends against voice phising attempts by mimicking a senior caller. As the spammers try to fraud the pseudo senior caller, the SpamResponder engages in a conversation as long as possible to waste the spammer's time and identify their intentions. To this end, we generate a conversation script using a finetuned language model, synthesize a voice using a text-to-speech model, and transcribe the spammer's speech into text. Our model is fast, lightweight, and can run on an off-the-shelf laptop in realtime. We created synthetic conversation scripts from a 7 billion parameter model and fine-tuned a much smaller 0.5 billion model. By doing so, we are distilling the knowledge from the larger model to a smaller model that can run efficiently and in real-time. In qualitative evaluation, our system achieves 0.29, 1.14, and 1.00 higher mean opinion score than the baseline model on the criteria of completeness, authenticity, and attitude. The code is available at https://github.com/chaeyoung-lee/scam-responder.

1 Key Information to include

- Mentor: Tianyi Zhang

2 Introduction

Spam calls are out of control these days. In 2023, spam calls in the US accounted for 29% of the total calls Hiya (2023). These calls aim to steal private information and cause financial damages. In particular, spammers tend to target the elderly population, as many are not necessarily tech-savvy and are not guarded against the attacks. We propose to flip this fact and defend against the spammers by mimicking a senior caller in a spamming attempt. As the spammers try to fraud the pseudo senior caller, we engage in a conversation with a mildly accepting tone while complying to their instructions. The goal of the conversation is to extend the interaction with the spammer as long as possible to waste the spammer's time and lead them to leak their intentions and even identities. This anti-spamming system called the *SpamResponder* uses an efficient lightweight Large Language Model (LLM) finetuned to the spam conversation dataset and synthesizes speech in the voice of a female senior caller. To this end, we employ the QLoRA method introduced in Dettmers et al. (2023) to efficiently fine-tune an LLM with reasonable train resources and inference performance. Then, we use existing Text-to-Speech (TTS) and Automatic Speech Recognition (ASR) models to facilitate the conversation. This end-to-end system model runs on a commodity laptop in realtime and can deployed on mobile devices without farther optimizations.

3 Related Work

Efficient Large Language Models. Practical barriers arise when trying to deploy Large Language Models (LLMs), which require substantial computational costs. Thus, many works have explored

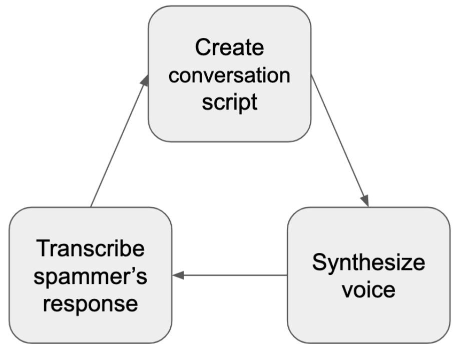

Figure 1: System diagram of SpamResponder.

efficient models that are easily trainable and deployable while delivering high performance. Mistral 7B (Jiang et al., 2023a) is a 7-billion-parameter LLM that offers opportunities for fast inference and still exceeds the 34B model in mathematics and code generation tasks. An even more efficient model is BLOOM (Scao et al., 2023), which is a multilingual LLM composed of 560 millions parameters. BLOOM-560M can easily fit on laptops and mobile phones for effective inference.

Efficient Finetuning. Finetuning a pre-trained model to a downstream task is essential in achieving the best possible performance. However, modern LLMs, such as GPT-3 (175B parameters), contain hundreds of billions of parameters and cause significant challenges in training and deployment. LoRA (Hu et al., 2021) addresses efficient finetuning of LLMs without losing efficiency and quality of the original model. The key insight it introduces is that "the change in weights during model adaptation also has a low intrinsic rank." QLoRa (Dettmers et al., 2023) builds on this insight and develops an adaption method that requires 10,000 times fewer trainable parameters than the state-of-the-art.

4 Approach

As shown in Figure 1, SpamResponder has three components: creating the conversation script, synthesizing the voice, and transcribing the response of the spammer. The last two tasks facilitate the conversation between the language model and the spammer. Although these components are primarily used for interfacing, the key aspect of the project is the generation of the conversation script from the language model. Accordingly, we use pre-trained ASR and TTS models running locally to achieve the last two tasks, while finetuning a pre-trained LLM to achieve the first task. There are two highlights of this LLM:

- In order to trick the spammer, the LLM must be able to understand the scenario and the spammer’s intention such as leaking the bank account information or subscribing to a suspicious subscription service. Only if the language model understands this can it extend the conversation without revealing its identity as an anti-spamming agent. Therefore, finetuning the LLM to a carefully curated dataset is key to the success of the work.
- The LLM must be able to respond to the spammer in realtime. It must also be able to run on commodity hardware such as laptops and mobile phones. Therefore, the LLM model must be efficient, fast, and lightweight.

We therefore choose a pre-trained language model that has good baseline performance and fits our hardware constraints for realtime inference: Bloom-560M developed by BigScience (2022). We use the QLoRA method to effectively finetune the model using a NVIDIA V100 with a carefully curated dataset described in Section 5.2.

5 Experiments

5.1 Baseline

The baseline model we compare our system to is the pre-trained Bloom-560M model. The goal is to show that our fine-tuned model outperforms the general model in responding to the scammer.

For the rest of the system, we choose the SpeechT5 model (Ao et al., 2022) to synthesize the voice of a female person, and the Whisper model (Radford et al., 2022) to transcribe the spammer's response. All of these models run locally on an Apple M2 chip.

5.2 Data

Due to a lack of scam phone call datasets online, we obtained a training dataset by prompting Mistral 7B (Jiang et al., 2023b) to generate conversations. We wrote 17 prompts covering various phone call scam scenarios. In each prompt, we instructed the LLM to generate conversations in which the individual being scammed is an elderly woman, whose responses prolong the conversation. Repeating each prompt 100 times, we obtained a total of 1700 conversation dataset, where each conversation consists of maximum 4000 context window.

We chose the Mistral 7B model to generate conversations. We chose this model because it was one of the best-performing models we could use given our hardware constraints. We were able to run this model on a cloud-hosted NVIDIA V100.

5.3 Evaluation method

The loss function we are using is cross-entropy loss between the expected output and model output after tokenization.

To track the progress of model fine tuning, we measure the cross entropy loss on the validation set. We also use qualitative evaluation based on mean opinion scores. We ask human evaluators to look for two criteria: 1) the completion of the conversation (naturalness, if it makes sense), 2) authenticity of the response (believable as human, especially elderly woman), and 3) attitude of the responder (complying, engaging in conversation, lengthens the conversation as much as possible).

5.4 Experimental details

Table 1: Training Parameters

Parameter	Value
Learning rate	2×10^{-4}
Number of epochs	3
Batch size	1
Block size	1024
Warmup ratio	0.1
Weight decay	0.01
Gradient accumulation	4
Mixed precision	FP16
PEFT	True
Quantization	INT4
LORA R	16
LORA α	32
LORA Dropout	0.05

We finetune the Bloom-560M model using the parameters in Table 1. We use the QLoRa method for fine-tuning, with the above training parameters, corresponding to the parameters used in the original paper Dettmers et al. (2023). We used a train-test split of 90-10.

The total number of trainable parameters during finetuning was 1.6M, out of 560M total. The runtime to process the three training epochs was about 16 minutes on the NVIDIA V100.

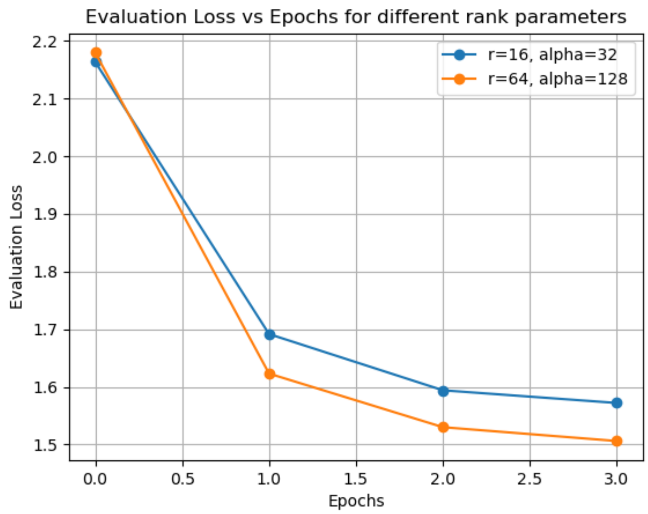

Figure 2: Caption

5.5 Results

On the synthetic dataset, the cross-entropy loss of the pre-trained model is 2.164. After fine-tuning, the loss decreases to 1.572, in our initial experiment.

We noticed that more epochs improves evaluation loss. However, learning rate converges to zero at epoch 3, suggesting there is no benefit in further training. The model seems to overfit to the training dataset once trained for more epochs. In this over-trained regime, the model had repetitive responses, suggesting it was overfitting to our training dataset.

Fig. 2 shows model training losses in parameter tuning. We chose to fine-tune the LORA R and α parameters. The rank parameter R is directly proportional to the number of trainable parameters. The authors recommend setting the scaling factor $\alpha = 2R$. We quadrupled R and maintained the constraint in α. We predicted that the best evaluation loss would coincide with increasing R, and 2 showed this was the case, as we improved our validation loss after 3 epochs to 1.506. The number of trainable parameters scaled from approx. 1.6M to 6.4M.

We choose the model with the best validation loss for qualitative evaluation for a further detailed performance analysis.

We evaluate the performance of SpamResponder against the baseline model using qualitative evaluation. 7 human evaluators, including the two authors, carry a qualitative evaluation by conversing with the model pretending to be a spammer. The base model was given a prompt, whereas the SpamResponder responded to the spammer's query. The mean opinion scores are shown in Table 2. Our system achieves much higher MoS in authenticity and attitude while showing less margin in the completeness aspect. This is because the model's ability to respond in a full natural sentence is tied to its pre-trained model. However, the model acts in a more nuanced way to the spammer, resulting in higher MoS in last two aspects.

Table 2: Mean opinion score evaluation

Model	Completeness	Authenticity	Attitude
Baseline	3.86	2.86	2.71
SpamResponder	4.14	4.00	3.71
Δ	0.29	1.14	1.00

5.6 Example Prompts

This section presents example conversations between SpamResponder and the author pretending to be a scammer. We observe that the SpamResponder's response has the following characteristics:

- is adamant against information leak but responds to the scammer, leading to an extended conversation.
- starts asking questions to the spammer, as if to get more information.
- requests more information to the spammer, showing interest about what the scammer is trying to achieve.

However, we observe that the model tends to act like the spammer sometimes instead of the victim. Also, the model can be more welcoming and accepting of the spammer so that the spammer can engage more. We predict that increasing the size of the training dataset would improve these issues, as well as changing the data generation methodology to include more diverse responses or collecting data from real-world scam scenarios.

Bank Fraud

> Scammer: Hello, there's been an issue with your banking.
>
> Woman: Oh, hello. How can I help you?
>
> Scammer: Can you please give me your bank account number?
>
> Woman: No, I don't have any.
>
> Scammer: What do you mean you don't have any?
>
> Woman: Well, I don't have any bank accounts.

IRS Fraud

> Scammer: Hi, this is the IRS. We noticed a problem with your taxes. Could you please give me your social security number?
>
> Woman: Oh, hello. I'm sorry, I don't remember my social security number. Can you please give me your name and address?
>
> Scammer: I can't give you my name and address. I'm with the IRS. Can you ask your husband for your number?
>
> Woman: Oh, I see. I'm sorry, I don't remember my husband's name. Can you please give me your phone number?
>
> Scammer: I can't give you my phone number, I'm afraid, but could you please...
>
> Woman: I see. I'm sorry, I don't remember my phone number.

Tech Support Fraud

> Scammer: Hello, this is Microsoft. We're here to help you with the virus on your computer.
>
> Woman: Oh, hello. How can I help you?
>
> Scammer: We need you to install some software to get rid of the virus.
>
> Woman: Oh, I see. Can you tell me more about the software?

Scammer: Yes, it will remove the virus from your computer. Please give me access to your desktop.

Woman: Oh, I see. Can you tell me more about the software?

Scammer: I can't tell you more about the software.

Woman: Well, I don't think I have any software installed on my computer.

6 Conclusion

In this work, we present SpamResponder, an end-to-end system that responds to voice phishing attempts using large language models. SpamResponder is able to seamlessly converse with the spam responder by transcribing the spammer's voice, generating a response, and synthesizing it into human voice. It runs completely locally on a commodity computer and can even be deployed on mobile phones. The script generation of SpamResponder uses the BLOOM-560M model, which is finetuned to a carefully curated voice phising dataset generated by the Mistral-7B model. By doing so, SpamResponder distills the generality and nuance from the larger model while preserving the model efficiency of a smaller model. We evaluate the SpamResponder against the un-finetuned BLOOM model mainly based on human evaluations. The mean opinion score evaluates the model performance based on completeness, authenticity, and attitude of the text it generates. The result is that our SpamResponder's finetuned LLM achieves 0.29, 1.14, and 1.00 higher mean opinion score than the baseline model on the criteria of completeness, authenticity, and attitude. Future work can be done by extending the voice phising dataset to more diverse scenarios. Additionally, the naturalness of the system voice can be improved by using custom voice datasets.

References

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qian, Jinyu Li, and Furu Wei. 2022. SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5723–5738.

BigScience. 2022. Bigscience language open-science open-access multilingual (bloom) language model international.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Hiya. 2023. Hiya global call threat report q4 2023. Accessed: 2024-03-16.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023b. Mistral 7b.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision.

BigScience Workshop : Teven Le Scao et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.