Leveraging Large Language Models for Vulnerability Classification

Steven Slater

4th Year Project Report
Computer Science
School of Informatics
University of Edinburgh

2024

Abstract

Classifying customers as vulnerable is an inherently complex task that presents many exceptions and conditions for classification and demands common-sense reasoning. Typically, to address these tasks, when labelled data is available, smaller LLMs are fine-tuned on this data, but this does not fully leverage the robust reasoning capabilities of LLMs. Prompting has also been used with LLMs but struggles to improve over fine-tuned models.

This thesis investigates the potential of using GPT-4, without extensive training data, to surpass a fine-tuned RoBERTa model on labelled data in vulnerability classification, aiming to enable the creation of future models that can achieve reliable performance with minimal training data. We look to two new methods, Chain of Thought Prompting and Prompt Optimisation, as recent methods of improving LLMs performance to enhance GPT-4's capabilities.

We explore Chain of Thought Prompting's application to complex common-sense reasoning tasks and conduct novel research into how varying example types can enhance the technique's performance, looking at the effect of vulnerable vs non-vulnerable examples and guideline vs misclassified examples. For Prompt Optimisation, we propose EdiPrompt, a new framework that builds upon EvoPrompt's (Guo et al. 2023) use of Genetic Algorithms in Prompt Optimisation by harnessing the In-Context Learning abilities of LLMs.

We find that direct examples are more effective than the detailed explanations provided by Chain of Thought Prompting for tasks demanding complex common-sense reasoning, and demonstrate that the variability introduced by Genetic Algorithms in EdiPrompt leads to convincing performance improvements over Google DeepMind's OPRO framework. Furthermore, we illustrate that when labelled data is scarce, GPT-4 with effective Prompt Optimisation is a viable solution for complex classification tasks, achieving an F-1 score just 4.7% below that of our fine-tuned RoBERTa model. However, when labelled data is available, fine-tuning remains the optimal approach.

Research Ethics Approval

This project obtained approval from the Informatics Research Ethics committee.
Ethics application number: 331851
Date when approval was obtained: 2023-11-09

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Steven Slater*)

Acknowledgements

I would first like to thank my supervisor, Alexandra Birch-Mayne, as without her guidance and support I would not have been able to do this. I would also like to thank Iria del Río for her help throughout.

Thank you to my family and friends who listened to me chat nonstop about nothing but prompts for the past year, but gave constant support nonetheless.

Finally, to my girlfriend Emily who now understands that RoBERTa is not in fact a woman, but instead a Large Language Model.

Table of Contents

Chapter 1

Introduction

Large Language Models (LLMs) like OpenAI's GPT series, have significantly impacted various fields with their ability to generate contextually relevant responses. However, the effectiveness of these models in specific Natural Language Processing (NLP) tasks continues to be a subject of exploration. The quality of the responses provided by these models often relies on the input prompt's quality (P. Liu et al. 2023), leading to the development of "Prompt Engineering," a field focused on the careful and strategic crafting of prompts in order to optimise the model's output (White et al. 2023).

Furthermore, the financial services industry faces challenges in identifying and assisting vulnerable customers. The Financial Conduct Authority (FCA) has criticised the industry's efforts and introduced new consumer duty regulations to ensure better care for this demographic, highlighting the need for improved detection measures. As a result of this motivation, this thesis, conducted in collaboration with Aveni Labs, aims to leverage the capabilities of OpenAI's latest model, GPT-4, for enhancing vulnerability detection. The goal is to evaluate whether GPT-4, a large general-purpose model, can outperform the fine-tuned but smaller, state-of-the-art RoBERTa model in binary vulnerability classification on phone call transcriptions between customers and financial advisors.

To achieve this goal, we leverage two new research areas, Chain of Thought Prompting and Prompt Optimisation, to assess their ability to enhance GPT-4's vulnerability classification performance. We are motivated by prior research showing that Chain of Thought Prompting boosts LLMs performance in logical reasoning tasks (Suzgun et al. 2022, Jason Wei et al. 2022) and previous research demonstrating that using LLMs to optimise prompts significantly outperforms human-generated prompts (Zhou et al. 2023, Yang et al. 2023, Pryzant et al. 2023). We explore Chain of Thought Prompting's applicability to complex common-sense reasoning domains like vulnerability classification and conduct new research that explores various strategies to identify the types of examples that can improve the technique's effectiveness. For Prompt Optimisation, we propose a novel framework, named EdiPrompt, that builds upon EvoPrompt's (Guo et al. 2023) pioneering use of Genetic Algorithms in Prompt Optimisation with stronger selection and combination stages alongside meta-prompts that fully utilise the In-Context Learning abilities of LLMs.

1.1 Research Questions

We now define the research questions that this thesis aims to answer.

- How effectively do Chain of Thought Prompting and Prompt Optimisation enhance GPT-4's performance compared to fine-tuning smaller models like RoBERTa on labelled data, for real-world complex classification tasks requiring specifications and common-sense reasoning, such as vulnerability classification?
- Which Chain of Thought Prompting strategies, including the use of positive versus negative cases of vulnerability and examples from FCA guidelines versus previously misclassified real conversations, yield the best performance in GPT-4's vulnerability classification?
- Can Genetic Algorithms be effectively applied within the new EdiPrompt framework to produce prompts that are higher performing and more diverse compared to the baseline established by Google DeepMind's OPRO framework?
- Can we naively combine Prompt Optimisation and Chain of Thought Prompting to leverage GPT-4 as a vulnerability classifier and surpass RoBERTa?

We address these research questions by conducting experiments that query GPT-4 through the OpenAI API with tailored meta-prompts, utilising the Aveni dataset of customer call transcripts, and repeating key experiments three times for robustness. All code can be found **here**.

1.2 Contributions

- Fine-tuned a high-performing RoBERTa model for the upper baseline on our conversation data.
- Designed a dataset of 100 hand-crafted Chain of Thought examples to use in experimentation.
- Conducted thorough research and testing on existing Prompt Optimisation frameworks to identify shortcomings, inspiring the creation of our new framework, EdiPrompt.
- We demonstrate that the extra variability introduced by Genetic Algorithms in EdiPrompt leads to convincing performance improvements over our OPRO baseline.
- We conducted an in-depth analysis of Chain of Thought Prompting and showed that strictly using negative examples of vulnerability is the optimal strategy, and for intricate common-sense reasoning tasks, examples outperform explanations.
- We establish that for complex classification tasks like vulnerability detection, fine-tuning a model with labeled data is most effective. However, in the absence of labelled data, GPT-4 with Prompt Optimisation offers a viable solution.

Chapter 2

Background & Literature Review

2.1 Deep Learning Based Text Classification

Text classification is the process of assigning pre-defined labels to text and traditionally relied on Machine Learning methods such as Support Vector Machines (SVM) or Random Forests. However, the effectiveness of these techniques is largely restricted by feature extraction (Qiu et al. 2020, Li et al. 2021). Feature extraction is the process of selecting key attributes or characteristics of data, such as selecting sentiment-indicative words in sentiment analysis (Shah and Patel 2016). Deep Learning models are a significant advancement from traditional Machine Learning models as they integrate the feature engineering process into the model itself, significantly reducing the need for human intervention and enhancing model effectiveness (Alzubaidi et al. 2021). Deep Learning models consist of structures called neural networks, which work by processing data through multiple layers of interconnected nodes or "neurons", adjusting their internal parameters in response to the input data to make accurate predictions or classifications.

An extensive survey evaluated over 150 deep learning models, highlighting the performance levels these models can achieve (Minaee et al. 2021). A standout finding from this survey was the BERT (Devlin et al. 2019) model developed by Google, which achieved an impressive accuracy of 99.32% in classifying topics from the Wikipedia dataset. OpenAI achieved state-of-the-art performance on detecting undesired content with their GPT model, as they combined many advanced topics such as Domain Adversarial Training, a method that trains a model to not differentiate between different types of data, overcoming the challenge of adapting to new domains (Markov et al. 2023). Similar work was also carried out in classifying hate speech, where BERT achieved 75.5% accuracy on the task (Röttger et al. 2021). Prior research shows that deep learning models excel at text classification. This thesis aims to evaluate GPT-4, the latest Generative Pre-trained LLM from OpenAI, at classifying vulnerable customers from phone call transcripts.

2.2 Large Language Models

Large Language Models (LLMs) are a type of deep learning model. They utilise a transformer architecture (Radford and Narasimhan 2018), which has revolutionised NLP and is the backbone of many state-of-the-art NLP models. A transformer model is similar to a neural model. However, they also use an attention mechanism (Vaswani et al. 2017), allowing the model to weigh the relevance of parts of an input sequence. This enables the models to capture contextual relationships and recognise long-range dependencies better than conventional neural models (Vaswani et al. 2017). GPT is short for Generative Pre-trained Transformer and was developed by OpenAI (Radford, Narasimhan, et al. 2018). It was the first of a new breed of models dubbed Generative Pre-trained Language Models. The 'Pre-trained' means it has been pre-trained on a large corpus of data. The 'Generative' part refers to the model's capability to generate new text sequences based on the input it receives.

In this thesis, we will be using **GPT-4** (OpenAI 2023) in all experiments to classify vulnerable customers. GPT-4 is the most recent and advanced model from OpenAI, and the main difference between it and its predecessors is its scale. The exact number of parameters in GPT-4 was not initially disclosed, but it is understood to be significantly larger than its predecessor GPT-3 with 175 billion parameters. GPT-4 has also incorporated more data into its pre-training, giving it a broader understanding of language and knowledge. Released in 2019, **RoBERTa** (Y. Liu et al. 2019) is an extension of Google's original transformer BERT (Devlin et al. 2019). RoBERTa improves on BERT by making significant modifications to the original training approach, as it was found that BERT was severely undertrained (Y. Liu et al. 2019). RoBERTa has outperformed BERT on several NLP tasks and achieved state-of-the-art text classification performance (Y. Liu et al. 2019).

The main differences between RoBERTa and GPT-4 can be characterised by two factors: directionality and their generative nature. Roberta processes text bi-directionally, simultaneously considering context from left to right and right to left to enhance language understanding for tasks like classification. GPT-4 generates text, predicting each subsequent word based on the previous ones to create coherent, contextually relevant responses across various topics and styles. These differences serve as a compelling motivation for this thesis, as we seek to assess if we can leverage GPT-4, which exists as a general model, using limited labelled data, to surpass a fine-tuned RoBERTa model on labelled data in the domain of vulnerability classification. We aim to enhance GPT-4 by leveraging its In-Context Learning ability.

2.3 In-Context Learning

LLMs, such as GPT-4, are task-agnostic, which means that they are not explicitly trained or fine-tuned for any downstream tasks, they exist as general models (Radford, Narasimhan, et al. 2018). In contrast, RoBERTa is usually fine-tuned and specialised to specific tasks (Y. Liu et al. 2019). As a result of GPT-4's nature, there is a paradigm called In-Context Learning (ICL). The term "In-Context Learning" describes a model's capacity to carry out tasks or comprehend instructions based on examples or context

given directly within the input prompt without the need for explicit retraining or fine-tuning on task-specific data (Dong et al. 2023). This method makes use of the vast amount of information and patterns the model has picked up during its initial pre-training, allowing it to adjust to new tasks or obey commands depending on the prompt's context. A prompt can be described as: "text fragments that are passed into language models that can encourage certain behaviours" (Brown et al. 2020).

Previous work in ICL has shown substantial gains, such as GPT-3 achieving a state-of-the-art F-1 score of 85.0% on the question-answering dataset, TriviaQA, when provided with examples in-context (Brown et al. 2020). Other research has also found that types of examples used in-context can produce varied effects, such as supplying examples the model has previously misclassified resulting in improved performance (Gao et al. 2023). On the other hand, Few-Shot Learning (FSL) is a broader term used in Machine Learning. FSL is a technique that aims to train models on a minimal amount of labelled data (Wang et al. 2020) We can think of ICL as a subset of FSL that specifically applies to LLMs, leveraging the examples provided within the prompt supplied to these models. In this thesis, when we mention FSL, we are describing the process of only supplying examples and their corresponding true label to the LLM (answer-only), we use this as a baseline.

ICL has created a new field, named "Prompt Engineering", as prompts allow for a more nuanced interaction with the model, they allow us to contextualise examples, specify desired outputs and even influence the style of response from the LLM. Prompt Engineering is defined as the process of designing prompts that clearly and effectively communicate the task to the model (White et al. 2023). Prompt Engineering has created many new prompting techniques, in this thesis, we investigate the new Chain of Thought Prompting technique (Jason Wei et al. 2022), with the aim of utilising it to enhance GPT-4 as a classifier on vulnerable customers.

2.3.1 Chain-of-Thought Prompting

A Chain of Thought (CoT) is a series of intermediate natural language reasoning steps that lead to the final output (Jason Wei et al. 2022). CoT Prompting uses this as a prompting method, where we show the LLM our reasoning process behind an answer by breaking down examples into a step-by-step process that leads to the final answer. This method encourages the model to emulate similar reasoning patterns, thereby enhancing its ability to generate more accurate and contextually relevant responses (Jason Wei et al. 2022). It is also believed that another reason CoT Prompting shows performance benefits is that such prompts allow the model to access the relevant information used in pre-training (Jason Wei et al. 2022).

Early works of CoT Prompting (Nye et al. 2021, Jason Wei et al. 2022) used In-Context Learning to get LLMs to display their work for mathematical problems. Utilising just eight CoT examples with PaLM 540B led to state-of-the-art accuracy on the GSM8K benchmark, a dataset of math word problems (Jason Wei et al. 2022). CoT Prompting was also used against 23 Big Bench Hard (BBH) tasks, which are tasks that human annotators outperformed language models on. When using CoT Prompting, the paper found significant gains over standard answer-only prompting. Specifically, for the Codex

model, CoT Prompting improved the performance from outperforming the average human-rater baseline on only 5 out of 23 tasks (for answer-only prompting) to 17 out of 23 tasks (for CoT Prompting)(Suzgun et al. 2022).

Although possessing strong results, all of these papers display a lack of in-depth research into the optimal design or structure of CoT prompts, and as a result may overlook potential performance enhancements. Furthermore, the CoT Prompting papers discussed (Jason Wei et al. 2022, Suzgun et al. 2022, Nye et al. 2021), all apply CoT Prompting to mathematical or logical reasoning problems, suggesting a narrow range of problem domains. This motivates our research into CoT Prompting, as we seek to explore whether its proven effectiveness extends beyond logical reasoning tasks to new areas like vulnerability classification, a task requiring complex common-sense reasoning to understand nuanced exceptions and conditions for classification.

2.4 Prompt Optimisation

Prompt Optimisation (PO) refers to the process of modifying or refining input prompts in order to obtain specific desired outputs or replies from the model. The primary goal of PO is to improve the model's performance and ensure that it generates accurate, relevant, and contextually appropriate responses. LLMs have been shown to be sensitive to the formatting of the prompts provided to them (Zhao et al. 2021, Jerry Wei et al. 2023). In addition to this, semantically similar prompts can produce an array of strikingly different results (Kojima et al. 2022). As a result of these findings, it makes the need to optimise our prompts a necessity.

Previous work in PO has explored using "soft prompts" where the prompt is represented as a task-specific continuous vector, meaning that the prompt is not fixed or discrete but instead can change based on examples and data. This work found significant performance increases compared to few-shot prompt design on GPT-3. (Lester, Al-Rfou, and Constant 2021, Shin et al. 2020). Despite these compelling results, the applicability of these methods is infeasible when we only have API access to the LLM (Yang et al. 2023). This results in this thesis using GPT-4 itself as an optimiser to find the best prompts for enhancing vulnerability classification performance.

Previous research has shown that using LLMs as optimisers can produce strong results. Google DeepMind developed a framework called OPRO (Optimisation by PROmpting). This framework iteratively refines prompts by testing newly generated prompts from the LLM on a fitness function and uses the result to improve future iterations. OPRO adds instruction-score pairs to the meta-prompt to help guide the model to generate better prompts in future iterations (Yang et al. 2023). OPRO showed strong improvements by outperforming human-generated prompts on Big-Bench Hard tasks by up to 50%. However, its limitations include generating only one prompt per iteration and not exploring variations of the prompt. OPRO also lacks guidance in its meta-prompts as it does not explicitly instruct the model to use the strong or diverse language within its highest instruction-score pairs. OPRO is used as a baseline in this thesis.

The Automatic Prompt Engineer (APE) framework (Zhou et al. 2023) optimises prompts by generating semantically similar variants of prompts and outperformed human-

generated prompts on 19 out of 24 of its NLP tasks. However, APE can be seen to be overlooking the creative capacity of LLMs by not generating entirely new prompts. Automatic Prompt Optimisation (APO) (Pryzant et al. 2023) saw boosts in prompt effectiveness by up to 31% over human efforts by seeking LLM feedback for refinement. However, it does not fully tap into the LLM's prompt creation capability. It is clear that PO is powerful and massively improves upon human-generated prompts, motivating its use in this research to enhance GPT-4 as a vulnerability classifier whilst also suggesting there is room for improvement upon currently available frameworks.

2.4.1 Genetic Algorithms in Prompt Optimisation

Genetic Algorithms (GAs) are algorithms inspired by evolution and have become a common method for solving optimisation problems since their creation in the 1960s (Mitchell 1998). GAs operate through selection, crossover, and mutation stages: selecting strong solutions, combining them, and then introducing random mutations to expand the search space and evolve solutions. GAs enhance variability by generating diverse solutions and exploring many possibilities through solution combinations and mutations (Mitchell 1998). EvoPrompt (Guo et al. 2023), the first to incorporate GAs into PO, achieved a 25% performance increase over human-generated prompts but left a lot of room for improvement in its approach.

EvoPrompt inadequately adapts GAs to leverage the creative capabilities and ICL potential of LLMs. EvoPrompt does not use any examples in its meta-prompts, such as instruction-score pairs (as adopted by OPRO) of previously high-performing prompts, when carrying out Prompt Optimisation. This neglects the LLM's ICL ability and gives the LLM no guidance when combining or mutating prompts, allowing for the language being combined or injected into the prompts to be sub-optimal. Additionally, EvoPrompt lacks clear task instructions, such as problem examples, leaving the LLM without a clear understanding of the task it is optimising, these critiques are also mentioned in OPRO's original paper (Yang et al. 2023).

A further drawback of EvoPrompt comes from its traditional usage of roulette wheel selection (Lipowski and Lipowska 2012). This selection method, common in GAs, means that the chance of a prompt being selected depends on its fitness score relative to the rest of the population; the higher the score, the higher the chance of being selected. However, it is still a random selection phase that permits low-performing prompts to be selected. This selection strategy does not guarantee that either of the prompts being combined are well-performing. EvoPrompt also heavily relies on good-quality and task-specific initial prompts to optimise from, as it does not provide any iterative updates to the LLM. It begins from a large domain of candidate prompts and evolves from this initial set.

We can confirm that these drawbacks limit EvoPrompt's efficacy, as in OPRO's original paper (Yang et al. 2023), it compares itself to EvoPrompt and shows that it convincingly outperforms it on two math-word datasets. These drawbacks highlight the need for a new Prompt Optimisation framework that better utilises the variability of Genetic Algorithms while fully leveraging the abilities of LLMs.

Chapter 3

Methodology

In this chapter, we develop our methodology, which is focused on two complimentary areas of research: Chain of Thought (CoT) Prompting and Prompt Optimisation. Our research is motivated by the goal of leveraging these techniques to enhance the performance of GPT-4 as a vulnerability classifier.

As a reminder, we research CoT Prompting as it has shown to significantly improve LLMs performance in logical reasoning tasks (Suzgun et al. 2022, Jason Wei et al. 2022, Nye et al. 2021). We aim to explore if this technique can also enhance GPT-4's performance in different domains, such as vulnerability classification, a task that requires complex common-sense reasoning to understand the many nuanced exceptions and conditions for classification. This chapter outlines our methodology for CoT Prompting, exploring how we create our CoT explanations and utilise examples from two distinct data sources to provide new insights into how varying example types influence CoT Prompting performance.

Our research into Prompt Optimisation stems from its proven effectiveness in previous research, where using the LLM as the optimiser to create prompts has significantly outperformed human-generated prompts (Zhou et al. 2023, Yang et al. 2023, Pryzant et al. 2023). We look to leverage Prompt Optimisation to refine our meta-instruction that instructs GPT-4 to classify a conversation, with the aim of optimised prompts enhancing GPT-4's performance as a vulnerability classifier. In this chapter, we propose EdiPrompt, a novel Prompt Optimisation framework that incorporates Genetic Algorithm paradigms to encourage the generation of more effective and diverse prompts.

This chapter also discusses our final experiment, which combines these two areas of research, alongside detailing our data, experimental metrics and baselines.

3.1 Definitions

We begin by defining some terminology used throughout to aid understanding.

- **Meta-prompt**: A meta-prompt is a high-level prompt. It can be though of as a prompt template for an overall task, such as vulnerability classification with CoT

Prompting. We can visualise our meta-prompt for our zero-shot learning baseline in Figure 3.3.

- **Meta-instruction**: A meta-instruction is the high-level instruction within a meta-prompt that provides the guidance to carry out the meta-prompt's task. In our context this would be how to approach and perform vulnerability classification or how to optimise prompts. We use the terms 'meta-instruction' and 'instruction prompt' interchangeably.

3.2 Data

We now define the datasets that we use in this thesis. This clarifies our task of vulnerability detection, a task which is regulated by the FCA in the UK to ensure firms provide the correct measures to identify and treat vulnerable customers.

3.2.1 FCA Guidelines

The FCA has a vulnerability guidelines document that contains a framework for identifying vulnerable customers. This framework is based on a taxonomy of four categories: Health, Life Events, Resilience and Capability. Each of these subcategories have their own list of events or actions that deem a customer vulnerable. This framework makes vulnerability classification an inherently complex task as it involves numerous exceptions and conditions for vulnerability, requiring nuanced understanding and assessment to identify each case accurately. The full framework can be seen in Appendix C. This document also outlines a definition of vulnerability based on this taxonomy, which can be seen in Figure 3.1. This definition is used throughout this thesis within the meta-prompts provided to GPT-4 to define the vulnerability framework it should use to classify vulnerable customers.

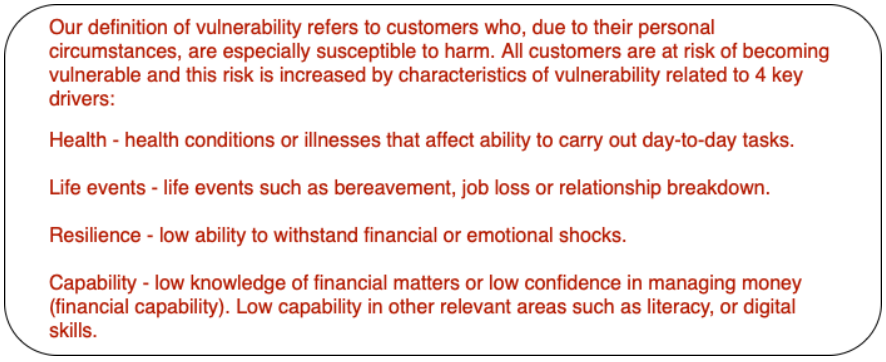

Figure 3.1: Definition of Vulnerability.

Aveni has amended the guidelines to include examples of fake customer utterances, indicating what constitutes vulnerability and what does not for each specific category. These hand-crafted, single-sentence examples, designed to represent customer speech rather than entire conversations, are clean and concise. They are utilised in the CoT Prompting experiments. Figure 3.7 shows an example of these utterances.

3.2.2 Aveni Vulnerability Dataset

The primary dataset used for this thesis is Aveni's vulnerability dataset, which is comprised of transcriptions of phone calls between customers and financial advisors. Due to the nature of transcription, these conversations are inherently noisy due to potential inaccuracies and errors when converting spoken words into text.

The conversations in the dataset are structured as arrays, where each element in the conversation array corresponds to an utterance from either the customer or advisor. Each conversation has a true label associated with it, indicating whether the conversation demonstrates an instance of vulnerability or not. The dataset includes metadata like specific vulnerability subcategories (Health, Life Events, Resilience and Capability). However, we do not utilise this metadata as we have made a simplifying assumption for this thesis, reducing our task to explicitly binary vulnerability classification. This approach allowed us to progress more effectively across all research areas. This simplification limits our thesis, indicating the potential for future research into multi-class vulnerability classification with GPT-4. The Aveni dataset has been split into three separate datasets: train, validation and test.

Dataset	Total Conversations	Non-Vulnerable	Vulnerable
Train	4663	3555	1108
Validation	1012	798	214
Test	1012	786	226

Table 3.1: Dataset Split

It can be seen that in Table 3.1 that there is a clear class imbalance present across all of the datasets. We discuss the effect that this imbalance has on training our RoBERTa baseline model in Section 3.9.2. This imbalance does not impact the provision of examples in CoT Prompting as we investigate various sampling strategies which are carried out with even sampling splits to mitigate any potential bias, this is discussed in Section 3.6.4.

3.2.2.1 Preprocessing and Cleaning

All metadata in the Aveni datasets that were not relevant to the specific binary vulnerability classification task were dropped, leaving just the conversations and their true labels. To train our RoBERTa baseline model, discussed in Section 3.9.2, we converted the original conversation arrays, where each element represented speech from either the customer or advisor, into a single continuous string.

When training RoBERTa, we found conversations exceeding it's 512 token context window. To address this, we adapted the datasets to fit RoBERTa's context window, using Python code and the RoBERTa tokenizer to trim the data. It was vital that the context of the vulnerability present in a conversation was kept, the original dataset provided a list matching each conversation where each utterance was annotated as either vulnerable or not vulnerable. This allowed for the vulnerable conversations to be trimmed around the vulnerability, where the vulnerable part of the conversation

was found, and a 512 token window was placed around it. For the non-vulnerable conversations, a random 512 token window was taken from it. Figure 3.2 displays an example of a vulnerable and non-vulnerable conversation after preprocessing. These are clear examples of the types of conversations we provide GPT-4 to classify.

Conversation with Vulnerability: "Id you say that again? Sorry. The line weren't quite funny. mhm. Oh, no worries. Um, so we've had a mortgage with you for about a year now. Mhm. Yeah. Er we have a joint account with you guys, and I think I've got a single one with you as well. Mhm. Yeah. Um, basically, uh, domestic violence has happened recently. Um, my mortgages is supposed to come out today, Mhm. Mhm. Mhm. Mhm. Mhm, Yeah. and he's informed me that he won't be putting money into the joint account to make that payment. So I just wanted to"

Conversation without Vulnerability: "So if you're thinking about going over for three months or four months, all we need to do is just make sure that in that extended period you were covered from a building's point of view from a rebuild point of view. So if anything, you know, if you had a gas late, god forbid or a you know, a burst pipe or something away, and it has blew up? Yep. You'll be covered in terms of the mortgage and stuff whilst you're whilst you're away. Yep."

Figure 3.2: Example Conversations with Vulnerability and without Vulnerability from Aveni Dataset. This also depicts the noise present in the transcribed conversations.

To ensure robustness and consistency, it was crucial to use the preprocessed datasets for both GPT-4 and RoBERTa to avoid giving GPT-4 an unfair advantage from a larger conversation window which could improve its classification. Figure 3.2 shows that the transcribed conversations between customers and financial advisors lack annotations to distinguish speakers. Since self-annotation for RoBERTa training was infeasible, we also avoided annotating data for GPT-4 experiments, ensuring both models were tested on identical datasets for consistency.

3.3 Meta-prompt Design

With our data understood, the next step is to understand how we use GPT-4 for vulnerability classification, further clarifying our goal with CoT Prompting and Prompt Optimisation. This process is done through the design of meta-prompts, which are identified as the most critical elements, as it is the meta-prompts alone that GPT-4 encounters when instructed to classify a conversation. All experiments are evaluated using the Aveni test dataset, meaning we insert test conversations one at a time into each experiments specific meta-prompt to receive a classification from GPT-4. All experiment meta-prompts can be found in Appendix A. We can visualise our meta-prompt for our zero-shot learning (no examples provided) baseline in Figure 3.3, which is discussed in Section 3.9.3.2.

We maintain the core structure of the meta-prompts across all experiments for uniformity, introducing only slight modifications to accommodate the addition of examples. The ordering of the meta-prompts are also kept consistent to ensure robustness. We can see this order in the meta-prompt for the zero-shot learning baseline in Figure 3.3. When examples are added, such as the reasoned CoT examples or the answer-only examples in few-shot learning, we place them between the definition of vulnerability and the conversation being classified. The meta-instruction in orange in this figure, is the human-generated meta-instruction we aim to optimise with Prompt Optimisation.

Our definition of vulnerability refers to customers who, due to their personal circumstances, are especially susceptible to harm. All customers are at risk of becoming vulnerable and this risk is increased by characteristics of vulnerability related to 4 key drivers.

Capability - low knowledge of financial matters or low confidence in managing money (financial capability). Low capability in other relevant areas such as literacy, or digital skills

Resilience - low ability to withstand financial or emotional shocks.

Life events - life events such as bereavement, job loss or relationship breakdown.

Health - health conditions or illnesses that affect ability to carry out day-to-day tasks.

Conversation to Classify: [...*insert conversation*...]

From the above conversation, is the customer vulnerable based on our definiton of vulnerability, only answer Yes or No, and you must say only Yes or No.

Figure 3.3: Zero-Shot Learning Meta-Prompt: Red indicates the definition of vulnerability, green indicates the conversation being classified and orange is the meta-instruction.

There are other meta-prompts designed to optimise prompts, these are discussed in their respective sections.

3.4 Experimental Metrics

3.4.1 F-1 Score

Throughout all the experiments the main metric used to analyse performance is the F-1 score. The F-1 score helps understand how well a test works by blending precision and recall into one score using the harmonic mean. The F-1 score formula can be seen in Figure 3.5. Precision measures the accuracy of the positive (vulnerable) predictions made by the model. It is calculated by the number of true positives divided by the number of true positives and false negatives. Recall measures the model's ability to correctly identify all relevant instances; in our case, this is vulnerable customers. It is calculated by the number of true positives divided by the sum of true and false positives. Both of these formulas can be seen in Figure 3.4.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

Figure 3.4: Precision and Recall Formulas

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

Figure 3.5: F-1 Score Formula

There are multiple reasons for this metric being the main focus. Firstly, as the dataset is imbalanced, with roughly three times more non-vulnerable examples than vulnerable

examples, using accuracy would be misleading, as a model could simply classify all customers as non-vulnerable and achieve a respectable accuracy.

We calculated the F-1 score for the vulnerable class, though it is important to note that we could have also calculated it for the non-vulnerable class. However, it is essential for our problem domain that importance is placed on the model's performance in classifying vulnerable customers correctly, as the primary goal of classifying customers as vulnerable is to accurately identify those needing additional support or services. Misclassifying vulnerable customers as not vulnerable (false negatives) can have serious implications, potentially leaving them without the help they need. The F-1 score, by combining precision and recall for the vulnerable class, directly measures how well the model identifies this critical group. Every F-1 score shown in the results section is the F-1 score on the vulnerable (positive) class.

3.4.2 Self-BLEU

One of the core research questions to answer regarding Prompt Optimisation is not only if the newly proposed EdiPrompt (Section 3.7.1) can produce stronger performing prompts compared to the OPRO baseline (Section 3.9.5) but also if it produces more linguistically diverse prompts. In order to measure this, the Self-BLEU metric is used. The Self-BLEU metric (Zhu et al. 2018) is a variation of the BLEU metric (Papineni et al. 2002). BLEU evaluates the quality of text generated by machine translation systems. It does so by comparing the machine-produced text to other reference texts. In contrast, Self-BLEU provides a metric to evaluate the diversity of text in a single model's output.

Self-BLEU is calculated by treating each text in a generated set as a "candidate" and the rest as "references," computing the BLEU score for each as the candidate, and then averaging these scores to assess the diversity of the set. Self-BLEU scores range from 0 to 1, with 0 signifying high diversity (no similarity) and 1 indicating low diversity (complete similarity) among generated texts. Self-BLEU is used in this thesis to compare the linguistic diversity of the set of prompts produced by EdiPrompt and OPRO.

3.5 Processing GPT-4 Responses

RoBERTa, our upper baseline, produces quantifiable vulnerability labels, in contrast to GPT-4's natural language outputs, requiring further steps to achieve binary classification from GPT-4. This is done through the meta-instruction, where from Figure 3.3 we saw that we instruct GPT-4 to only answer with "Yes" or "No", and then again ask it to solely output a "Yes" or "No" response. During testing, it was found that GPT-4 would occasionally output the response: "insufficient information". To address this, we decided to default any response that was not a clear "Yes" or "No", strictly to "No" to manage ambiguity and prevent inflating true positives by defaulting them to "Yes". This approach ensures that the F-1 score accurately reflects the models ability in identifying true vulnerability. All processed responses from GPT-4 are stripped of white spaces, punctuation and are reduced to lowercase to process all responses clearly.

3.6 Chain of Thought Prompting

We now explore our methodology for CoT Prompting. Previous research has solely focused on mathematical and logical tasks (Jason Wei et al. 2022, Suzgun et al. 2022, Nye et al. 2021), which posses a clear, rule-based reasoning path leading to a specific answer. However, when extending CoT Prompting to vulnerability classification, we are presented with new challenges. Unlike mathematical problems, vulnerability classification requires complex common-sense reasoning to assess the vast array of potential vulnerabilities present, each with unique contexts and exceptions. For example, a customer could say "I underwent a battle with cancer five years ago, which drained my savings. While physically recovered, my financial strain and anxiety have made it hard for me". This example underlines the complexity of classifying vulnerability, as despite recovering from past struggles, the customer remains vulnerable due to ongoing financial and emotional repercussions. This intricate reasoning suggests adapting CoT Prompting to vulnerability classification could be more difficult as it involves teaching GPT-4 how to break down our classification task with common-sense reasoning.

This thesis seeks to investigate how CoT Prompting performs in new domains that require this complex reasoning, specifically vulnerability classification. This investigation will also offer insights for enhancing LLMs text classification performance across various real-world scenarios, particularly in domains marked by numerous exceptions and specific classification conditions. We now outline how we create our CoT explanations and our novel approach to leveraging two distinct data sources to explore how different example types impact CoT Prompting. Our method addresses key research questions by evaluating CoT Prompting's role in improving GPT-4's vulnerability classification performance and finding the best example strategy for enhanced performance.

3.6.1 Chain of Thought Prompting Strategies

We define 'strategies' as the sampling of different types of in-context examples that we provide to GPT-4 when using CoT Prompting. Previous CoT Prompting methodologies have not investigated what strategies invoke better performance, as they were done on mathematical and logical problems, so did not have access to examples that exhibit polarity (positive/negative) or varying data quality (noisy/clean). Our methodology allows for novel insights into how we can improve CoT Prompting performance through the provision of different example types, focusing on three key characteristics: **Example Quality**, **Similarity to Target Data**, and **Example Polarity**. We carry out this investigation using two distinct example sources:

- **Misclassified**: A subset of the Aveni training dataset, which uses specifically previously misclassified examples. This has been inspired by research that found that using examples the model had previously failed on improved classification performance in a few-shot learning context with LLMs (Gao et al. 2023). This thesis aims to see if these findings are transferable to CoT Prompting. Using previously misclassified examples aims to allow GPT-4 to learn the more difficult edge cases of vulnerability detection. Access to misclassified examples was provided through testing experiment code on the training dataset, where the

conversation to classify, GPT-4's classification and the true classification were saved to a file. The misclassified source is noisy as it is a subset of our core Aveni dataset of transcribed phone calls. It is also similar to the target data GPT-4 will be asked to classify as it originates from the same source as the test set. The misclassified source contains both vulnerable and non-vulnerable instances.

- **Guidelines**: The second example source is from the FCA guidelines, which as discussed prior contains single sentence utterances of customers. It is clean as they were designed to emulate real speech rather than a transcription. These utterances are not similar to the target data GPT-4 will be asked to classify but still provide examples of what constitutes vulnerability and what does not.

Examples of both sources can be seen in Figure 3.7. The difference in quality between the two example sources allows for the hypothesis that the 'cleaner' examples from the guidelines may facilitate more effective learning for GPT-4, potentially boosting its classification performance. However, the structural differences between these guideline examples and the complex, real conversations GPT-4 will encounter raise questions about the model's ability to generalise from guidelines to actual conversations. Conversely, while misclassified examples are noisy, their origin from the same source as the target data might better prepare GPT-4 by exposing it to similar types of content it will classify.

The polarity in the examples used in CoT Prompting on GPT-4's performance also raises the question of whether it is more effective to use strictly positive, negative, or mixed examples of vulnerability. Negative examples may boost the F-1 score for the positive class by teaching the model what non-vulnerability looks like, allowing for true vulnerability to be identified easier. However, positive examples may enhance the innate understanding of vulnerability.

3.6.2 Chain of Thought Explanations

Many different structures for the CoT reasoned explanations were considered. Each structure offered a unique thought channel, helping GPT-4 'learn' to accurately classify vulnerability in customer conversations, as shown in Figure 3.6.

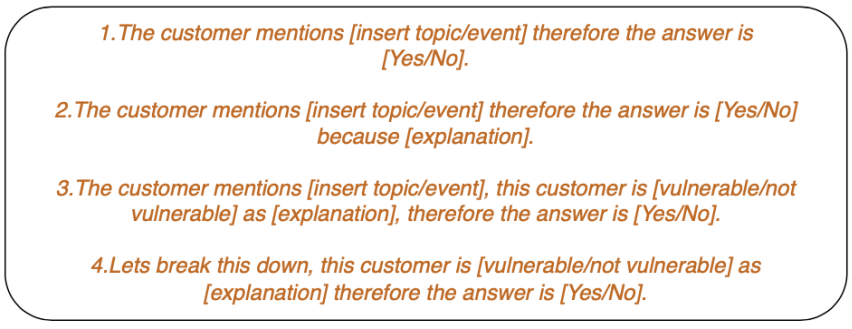

Figure 3.6: Potential Chain of Thought Explanation Structures

Due to budget constraints and our focus on evaluating CoT Prompting and its related strategies, not all structures were tested. We have assumed our chosen structure will

be effective for vulnerability classification as no prior research has provided direction upon crafting an optimal structure. This assumption marks a limitation and elicits future research to investigate the most effective CoT structure for vulnerability classification. Structure 3 was selected for experimentation because of its multi-stage reasoning process, which mirrors human-like thought patterns by identifying key elements, classifying vulnerability with an explanation, and concluding with a final answer.

3.6.3 Chain of Thought Dataset

A hand-constructed dataset combined the two data sources (misclassified and guidelines). The CoT dataset contains 100 examples and their corresponding CoT explanation; it can be split into four subsections: misclassified (vulnerable/non-vulnerable) and guidelines (vulnerable/non-vulnerable), with 25 of each making up the CoT example dataset. An example of each subsection can be visualised clearly in Figure 3.7. The main limitation of the CoT Prompting experiments is their small dataset due to the lengthy processing of hand-crafting CoT explanations. This contrasts our few-shot learning baseline which draws examples from the large training dataset, discussed in Section 3.9.3.2. This smaller example bank might put CoT Prompting at a disadvantage as it has a smaller variety of instances to sample from. However, if its performance still exceeds expectations, it significantly highlights its effectiveness.

Misclassified (Vulnerable example): "Is that excellent? Mhm. So I've actually got the qualification to do that, but we We're the company secrets. We don't do them Yeah. Okay. Yeah. Okay. That's interesting. and they're good. They were good. Uh, they were good. Modern. Hang on. Yeah, yeah. No, Most definitely. Definitely. Um, you know, if this financial hardship does, um, if his health condition does hinder his ability to work in in the near future, and he is able to access any pensions, as we say, then it's something to bear in"
Chain of Thought: The customer mentions financial hardship, alongside someone they know having a health condition, this customer is vulnerable as they are suffering financially. Therefore, the answer is Yes.

Misclassified (Non-vulnerable example): "come up on my screen? What's now? What do you want me to do now? Um, so I'll just carry on with the rest of this form for you. Not quite complete. Give me to the form. So? So I could say so. This list of fees for the standards list to process has £899. Oh, I'm sorry. I've got what I've got on my screen is about our equity release services. That's what it says. And it starts off whose products that we offer, which, uh, this what we provide you with No separate. Oh, that just popped up on the screen. Yes. So"
Chain of Thought: The customer mentions paying fees, this customer is not vulnerable as they are not describing any difficulty paying. Therefore, the answer is No.

Guidelines (Vulnerable example): "I have, you know, I have drinking problems."
Chain of Thought: The customer mentions having a drinking problem, this customer is vulnerable as they have an unhealthy addiction to alcohol. Therefore, the answer is Yes.

Guidelines (Non-vulnerable example): "My daughter had the flu last week and she felt really ill"
Chain of Thought: The customer mentions that their daughter was ill with the flu, this customer is Not Vulnerable as the this does not affect the customer directly and the illness is not severe. Therefore, the answer is No.

Figure 3.7: Different example source instances and their corresponding Chain of Thought explanation.

3.6.4 Chain of Thought Prompting Experimental Approach

To assess CoT Prompting's impact on GPT-4's vulnerability classification performance and identify the optimal example strategy, nine experiments were conducted. These experiments can be seen in Table 3.2. Each experiment is evaluated on classifying conversations from the Aveni test dataset, where each test conversation receives randomly selected in-context CoT examples from its respective experiments example source (misclassified or guidelines) and example polarity (vulnerable or non-vulnerable) from that source.

CoT Experiment	Example Source	Example Polarity
CoT 1	50/50 M&G	50/50 V&NV
CoT 2	50/50 M&G	V
CoT 3	50/50 M&G	NV
CoT 4	G	50/50 V&NV
CoT 5	G	V
CoT 6	G	NV
CoT 7	M	50/50 V&NV
CoT 8	M	V
CoT 9	M	NV

Table 3.2: Chain of Thought Prompting Experiments. M = Misclassified source, G = Guideline source, V = Vulnerable Instances, NV = Non-Vulnerable Instances. 50/50 implies an even sampling split.

We use 12 examples for all CoT Prompting experiments. This number is based on the performance of 12 examples in the few-shot learning baseline, where results plateaued, as detailed in the results and discussion chapter in Section 4.1. Using 12 examples allows for even sampling, such as three examples from the four subcategories in CoT 1. By adopting a 50/50 sampling approach, we ensure result comparability and reliability by preventing biases from oversampling or undersampling different example types.

3.7 Prompt Optimisation

We now begin to explore our methodology for our second area of research, Prompt Optimisation. Previous research has developed Prompt Optimisation frameworks that use the LLM itself to generate new and refined prompts, aiming for more effective communication and improved responses from the model. Frameworks such as OPRO (Yang et al. 2023), APE (Zhou et al. 2023) and APO (Pryzant et al. 2023) that were discussed in the background chapter use methods such as providing examples of past successful prompts to the LLM for help, instructing the LLM to generate semantically similar variations of prompts and asking the LLM for text feedback on prompts. Most notably there was EvoPrompt (Guo et al. 2023), which was the first to use Genetic Algorithms in Prompt Optimisation. Their use is motivated by the ability of Genetic Algorithms to provide extra variability, exploring a broad solution space through combining and mutating solutions (Mitchell 1998).

The key issue identified in previous frameworks, especially EvoPrompt, was their inability to fully leverage the In-Context Learning potential of LLMs by using examples of successful prompts, providing clear instructions, and clarifying the optimisation task through problem examples. Additionally, EvoPrompt did not reliably select and combine high-performing prompts and heavily relied on good-quality and task-specific initial prompts to optimise from, as it began with a large set of candidate prompts and evolved from this initial set. Motivated by these limitations and the demonstrated potential of Genetic Algorithms in Prompt Optimisation as initially explored by EvoPrompt, we propose EdiPrompt (Edinburgh Prompt optimisation), a new Prompt Optimisation framework that is the first to use Genetic Algorithms in Prompt Optimisation with full utilisation of LLMs In-Context Learning abilities.

EdiPrompts novel advancements over EvoPrompt include integrating examples of previous high-performing prompts in its meta-prompts, aiding the LLM in recognising effective language. It also includes problem domain examples to ensure the LLM understands its optimisation task. Unlike EvoPrompt, EdiPrompt does not begin with a large number of candidate prompts but generates them at each iteration and hopes to refine them through iterative updates, reducing reliance on the initial prompt quality. Additionally, EdiPrompt employs a new method for generating candidate prompts aiming to uncover more effective candidate prompts by generating semantically similar variants and an enhanced selection method to ensure that only the strongest candidate prompt is selected. It also features a refined crossover stage, with a meta-prompt that provides scores of the prompts being combined to further guide GPT-4 and an approach that consistently merges the best overall prompt with the newly selected prompt to maintain a high-quality combination baseline.

The construction of EdiPrompt proved very challenging, involving rigorous testing and design alongside the exploration of existing frameworks to assess the proficiency of their concepts, where concepts from other frameworks have been incorporated, we acknowledge them explicitly. EdiPrompt aims to deliver high-performing and linguistically diverse prompts for a wide range of problems. For our task of classifying vulnerability, we employ GPT-4 as the optimiser through EdiPrompt to optimise our original human-generated meta-instruction, which was seen in Figure 3.3, that asks GPT-4 to classify a conversation, with the aim of optimised prompts enhancing vulnerability classification performance from GPT-4.

3.7.1 EdiPrompt

3.7.1.1 Overview

The EdiPrompt framework operates through three phases in each iteration, as shown in Figure 3.8.

- **Selection Phase**: GPT-4 generates an initial prompt along with two semantically similar variations, forming three candidate prompts, from which the best-performing prompt is selected.
- **Genetic Phase**: GPT-4 combines the selected prompt with the best overall prompt (from all previous iterations), it then mutates it to create a new, potentially

improved prompt.

- **Update Phase**: This stage updates the prompt examples in our meta-prompts and the overall best prompt based on the Genetic Phase outcomes.

Following each iteration, we restart the process, aiming for continuous improvement in prompt performance, facilitated by the iterative updates during the update phase. These updates allow GPT-4 to learn from the successes of each iteration. While EdiPrompt can be run for any number of iterations, budget constraints in this research limit us to conducting only eight iterations. We aim for the mutated prompt to show gradual improvement as it is the result of a complete cycle of the framework. We now discuss EdiPrompt in-depth, including its meta-prompts and each distinct phase.

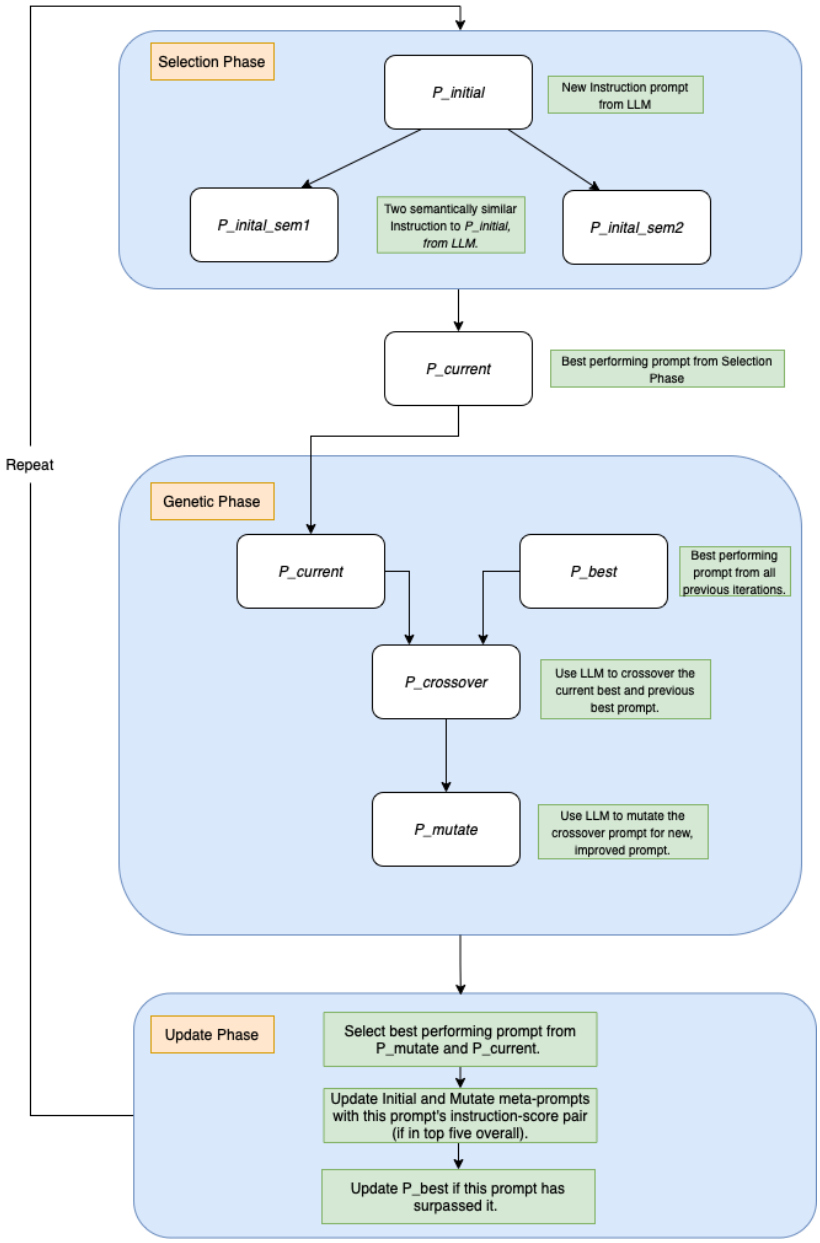

Figure 3.8: EdiPrompt Framework Overview

3.7.1.2 Evaluating Generated Prompts

We evaluate model generated instruction prompts in a zero-shot learning environment where they substitute the human-generated instruction prompt in Figure 3.3. This approach was chosen for budgetary reasons, as this allows for costs to be saved on smaller context windows when querying GPT-4. This approach also reflects the prompts true quality by eliminating potential biases from example inclusion. We test how the instruction prompt performs in classifying conversations in the Aveni test dataset, recording the F-1 score it achieves. Therefore, when we refer to a prompt as high-performing or better than another, it signifies that the prompt has achieved a notable F-1 score. This method enables clear observation of performance improvements and motivates the final experiment, which combines the best generated instruction prompt with the optimal CoT strategy.

3.7.1.3 EdiPrompt Meta-Prompts

In EdiPrompt, we developed four distinct meta-prompts, detailed in Appendix A. A complete EdiPrompt iteration involves GPT-4 generating an initial prompt, two semantically similar prompts, a crossover prompt, and a mutated prompt.

Two key stages are generating the initial prompt and mutating the crossover prompt, as this is where GPT-4 introduces new language into prompts. In these two stages, we provide examples of previously successful prompts to guide GPT-4 in learning language patterns in prompts that have previously led to high F-1 scores. We provide these examples in the form of instruction-score pairs (model-generated instruction prompts and their scores), an approach inspired by OPRO (Yang et al. 2023). We retain the top five performing prompt instruction-score pairs found across all iterations, they are updated in the update phase at the end of each iteration if a new top five prompt has been found. Moreover, these meta-prompts also incorporate our definition of vulnerability and twelve problem examples randomly taken from the training dataset to ensure GPT-4 understands our problem domain and how the instruction will be used when it creates new prompts. The primary distinction between these meta-prompts lies in the mutate step, where we explicitly direct GPT-4 to mutate the crossover prompt. Figure 3.10 shows the meta-prompt for the mutation stage.

To generate semantically similar prompts, we simply provide GPT-4 with the initial prompt and instruct it to generate two semantically similar variants. Finally, to ensure guidance in the crossover stage, our meta-prompt incorporates the scores of the two prompts being combined. This strategy enables GPT-4 to understand which prompt uses more effective language for our task, guiding its decision-making in the prompt combination process. This meta-prompt also instructs GPT-4 to combine the elements it deems more effective. We assume GPT-4 can identify more effective language, a limitation but reasonable given its advanced understanding of language. We also instruct GPT-4 for text feedback on its crossover process, this was used to test that it was working correctly. The guidance given to GPT-4 through instruction-score pairs, problem domain examples and the scores of prompts being combined mark EdiPrompt's advancements over EvoPrompt. Figure 3.9 shows the meta-prompt for the crossover stage.

Below are two instructions with their scores, the score is the f-1 score. We want to maximize this score.

Instruction 1: [...*insert first instruction (from selection phase)*...]
Score: [...*insert first instruction score*...]

Instruction 2: [...*insert second instruction (overall best)*...]
Score: [...*insert second instruction score*...]

Your task is to cross-over the above two instructions, combining the most effective elements from each to create a new, improved instruction. Also provide an explanation into the crossover and what elements you have taken from each instruction.

Figure 3.9: EdiPrompt Crossover Stage Meta-Prompt. Orange indicates the meta-instruction and blue indicates the prompts being combined and their scores.

Your task is to mutate the current crossover instruction to create a new instruction <INS>. Below are some previous instructions with their scores, the score is the f-1 score. We want to maximize this score.

Instruction: [...*insert instruction*...]
Score: [...insert instruction score...]

(..more instruction-score pairs...)

Below are examples of the problems, we are classifying customers conversations as vulnerable or not vulnerable. Our definition of vulnerability refers to customers who, due to their personal circumstances, are especially susceptible to harm. All customers are at risk of becoming vulnerable and this risk is increased by characteristics of vulnerability related to 4 key drivers:
Health - health conditions or illnesses that affect ability to carry out day-to-day tasks.
Life events - life events such as bereavement, job loss or relationship breakdown.
Resilience - low ability to withstand financial or emotional shocks.
Capability - low knowledge of financial matters or low confidence in managing money (financial capability). Low capability in other relevant areas such as literacy, or digital skills

Conversation: [...insert example conversation...]
<INS>
Answer: Yes/No

(...more problem examples...)

Crossover Instruction: [...*insert crossover instruction*...]

Your task is to mutate the above crossover instruction, to generate a new improved instruction that is different from all the instructions above, and has a higher score than all the instructions above. The instruction should be concise, effective, and generally applicable to the classification task at hand. The instruction must contain that we need a Yes or No response.

Figure 3.10: EdiPrompt Mutate Stage Meta-Prompt. Orange is meta-instruction, blue is instruction-score pairs, red is our definition of vulnerability, purple is problem examples, green is crossover instruction.

3.7.1.4 Initialisation of EdiPrompt

To begin the first iteration of EdiPrompt, it is clear from our framework overview in Figure 3.8 that we need an initial prompt, $P_{initial}$, and a starting best prompt, P_{best}. To acquire these a simple approach was taken, we used our meta-prompt for generating an

initial prompt, where the only instruction-score pair provided was our original human-generated meta-instruction. We got GPT-4 to generate two new instruction prompts from this, we then evaluated them both and the highest performing one was made P_{best}, and the other $P_{initial}$. We now proceed to discuss each phase in-depth referencing our overview in Figure 3.8 as we go.

3.7.1.5 Selection Phase

This is the start of an iteration of EdiPrompt, as seen in the first blue box in Figure 3.8. We obtain our initial prompt, P_{inital}, and two more semantically similar prompts, P_{inital_sem1} and P_{inital_sem2}, from GPT-4. Our approach of generating semantically similar prompts to create our candidate prompts is inspired from APE (Zhou et al. 2023) and previous research that showed how semantically similar prompts can produce varying performance (Kojima et al. 2022). This is a completely new approach to generating candidate prompts and was not done by EvoPrompt. The goal is to explore the area of language around the original prompt and uncover more effective prompts. The choice to only generate three candidate prompts was due to budget restrictions as evaluating each prompt incurs a cost. Generating more semantically similar prompts would enhance EdiPrompt's chances of uncovering more effective candidate prompts, marking a limitation of its design. However, generating two semantically similar prompts was deemed enough to explore the language around the initial prompt. We evaluate all three candidate prompts and advance the best-performing one, now called $P_{current}$, to the genetic phase. Unlike EvoPrompt's roulette wheel selection (Lipowski and Lipowska 2012), which allows even lower-scoring candidate prompts a chance to progress, our method ensures only the highest-scoring candidate prompt moves forward.

3.7.1.6 Genetic Phase

In this phase we apply the concepts of crossover and mutation from Genetic Algorithms. This can be seen in the second blue box of Figure 3.8. For the crossover step we use GPT-4 to combine two prompts, the first is the best prompt found in the selection phase of this iteration, $P_{current}$. The second is the top-performing prompt from all past iterations, P_{best}. The goal is to merge the high-quality linguistic elements of P_{best}, with potentially new, effective language from $P_{current}$, aiming to maintain a strong prompt baseline while introducing fresh elements that could further enhance performance. This approach, distinct from EvoPrompt, which did not ensure prompts being combined were high quality whilst also aiming to uncover new prompts, marks another advancement in EdiPrompt's methodology. This process generates $P_{crossover}$ and we finish with asking GPT-4 to mutate it, creating P_{mutate}. We evaluate P_{mutate} and proceed to the update phase.

3.7.1.7 Update Phase

The update phase is the final phase of an EdiPrompt iteration, as seen in the final blue box of Figure 3.8. Firstly, we need to find the best prompt that was found in this iteration. We aim for this prompt to be P_{mutate}, however this is not guaranteed. The best prompt will be from either P_{mutate} or $P_{current}$ as we do not test $P_{crossover}$ to save costs. Once

we have selected the best prompt from this iteration we update our instruction-score pairs in our initial and mutate meta-prompts if this prompt is in the top five overall. Additionally, P_{best} is updated if it has been outperformed. Over time, the updates in this phase are expected to enhance the optimisation process, steadily improving both the instruction-score pairs and the best prompt through each iteration. This enables continuous learning and refinement of GPT-4's generated prompts. After completing these updates, we restart from the selection phase in a new iteration.

3.8 Combining Prompt Optimisation & Chain of Thought Prompting

We now discuss our methodology for our final experiment, which naively combines both CoT Prompting and Prompt Optimisation. We aim to investigate if we can combine CoT Prompting and Prompt Optimisation to enhance GPT-4's performance as a vulnerability classifier, a research question of the thesis. This final experiment combines the best strategy found in CoT Prompting with the highest-performing instruction prompt generated from Prompt Optimisation.

The motivation for this experiment comes from two facts; firstly, all CoT Prompting experiments were completed with the human-generated meta-instruction. Secondly, all prompts tested in Prompt Optimisation were tested in a zero-shot learning context to minimise costs. This leads to the hypothesis that combining the two could increase GPT-4's performance as a classifier compared to how the two techniques performed individually and relative to the RoBERTa baseline. This enhancement is anticipated as the superior prompt benefits from the reinforcement of an effective CoT Prompting example strategy, and vice versa.

We describe our method as 'naive' because it simplistically combines the two areas without tailoring the optimised prompt for CoT examples. Budget and time constraints prevented us from re-optimising the prompt to support this. Essentially, the optimised prompt only instructs GPT-4 to classify a conversation without guiding it to use CoT examples. This presents a limitation and suggests future research to integrate CoT Prompting and Prompt Optimisation more effectively, considering their complexities.

3.9 Baselines

Having established our methodology for CoT Prompting and Prompt Optimisation, we now describe all the baselines against which these methods will be evaluated.

3.9.1 Keyword Baseline

The implementation of a keyword classifier as a lower baseline helps assess GPT-4's performance in vulnerability classification, clarifying the impact of CoT Prompting and Prompt Optimisation in this thesis. The construction of the keyword classifier followed a very simple strategy. Utilising the existing taxonomy for vulnerable customers (Health,

Life Events, Resilience, Capability), Four separate lists were constructed for each category, where each list contained ten 'keywords' associated to that category. The keyword classifier will then positively classify a customer conversation if any of the words in these lists are present in the conversation. The keyword baseline serves as a great lower performance baseline as it only looks for certain keywords, unlike LLM's that can understand complex sentences. The keyword baseline keywords can be seen in Table 3.3 and its results on the Aveni test dataset in Table 3.4.

Health Words	**Life Events Words**	**Resilience Words**	**Capability Words**
Death	Death	Mortgage	Dyslexia
Illness	Divorce	Benefits	Disability
Hospital	Loss	Universal Credit	Older
Cancer	Grieving	Bankruptcy	Young
Chemotherapy	Break Up	Debt	Child
Surgery	Funeral	Bills	Dyspraxia
Medication	Sacked	Overdraft	Lost
Treatment	Fired	PIP	Dementia
Disease	Unemployed	Redundancy	Learning Difficulty
Pain	Separation	Furlough	Assistance

Table 3.3: Keyword Baseline Words

Metric	Value
Accuracy	66.3%
Precision	33.0%
Recall	49.1%
F-1 Score (Positive Class)	**39.4%**

Table 3.4: Keyword Baseline Test Metrics

3.9.2 RoBERTa Baseline

A fine-tuned RoBERTa (Y. Liu et al. 2019) model was used as the main upper baseline for this thesis due to its state-of-the-art performance in text classification. This approach enables clear comparisons between the performances of RoBERTa and GPT-4, particularly when employing CoT Prompting and Prompt Optimisation. By setting this benchmark, we aim to explore the potential of these methods in enhancing GPT-4 to reach or surpass the performance of a fine-tuned but smaller RoBERTa model in the complex domain of vulnerability classification, which represents the core research question of this thesis. Creating the RoBERTa baseline proved challenging, requiring significant time to understand Python's PyTorch library, which I had not used before, and extensive hyperparameter tuning to develop a high-performing model.

3.9.2.1 Model Architecture

The construction of the model utilised the transformers Python library from HuggingFace, where the base RoBERTa model and its tokenizer was imported into the codebase.

PyTorch was used to add a linear classifier layer on top of the base RoBERTa model, this custom layer adapts the model to the specific binary classification task by mapping the model's output to a single value, representing the probability of a positive class (vulnerability).

PyTorch's neural network module is utilised to define a Binary Cross Entropy with Logits loss function. In order to enable effective and precise binary classification, the Binary Cross Entropy with Logits loss function first uses a sigmoid function to convert logits into probabilities (which range from 0 to 1) and then uses binary cross entropy to measure the error between these probabilities and the actual binary labels. An Adam optimiser is employed to update the model's weights based on this loss, with a specified learning rate. The Adam optimiser is an algorithm for gradient-based optimisation of stochastic objective functions, it is ideal for handling sparse gradients in transformer models like RoBERTa. Figure 3.11 allows us to visualise this.

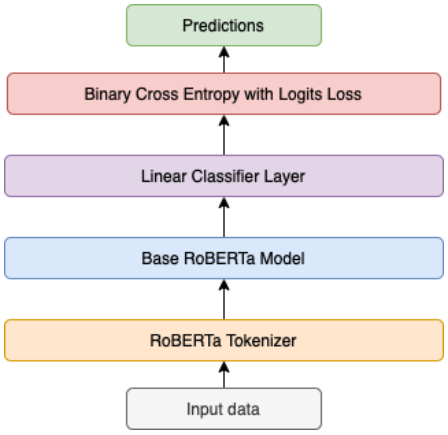

Figure 3.11: Fine-tuned RoBERTa Model Architecture

During training, PyTorch tensors facilitate batch processing, where the model resets gradients, processes input data as tensors for predictions, calculates loss with target labels, and updates weights via backpropagation. The base RoBERTa model was fine-tuned on the Aveni training dataset, which as discussed prior was transformed in order to fit the token context window of RoBERTa.

3.9.2.2 Hyperparameters

Many iterations of hyperparameter tuning took place before the baseline was deemed suitable. Aveni kindly provided metrics for a fine-tuned RoBERTa model that they had built on the same data. I aimed to achieve similar metrics. All hyperparameter tuning was tested on the validation dataset, allowing performance comparison and insights into whether parameters were beneficial to the model or detrimental to it by overfitting on the training set. During the training phase, many factors were considered to improve the model's performance on the validation dataset.

Class Imbalance: A class imbalance was present in the training dataset toward the non-vulnerable class. I experimented with adding weight to the positive class to combat this imbalance; however, this did not have much effect on the model's performance.

Oversampling was also done on the positive class, and this still did not provide the benefit needed. Ultimately, addressing this imbalance became unnecessary once other hyperparameters were effectively tuned.

Learning Rate: The learning rate is a crucial hyperparamter as it governs the magnitude of updates to the model's parameters during the optimisation process. Values between 2×10^{-4} and 5×10^{-4} were experimented with. 2×10^{-4} produced the best results after careful experimentation.

Batch Size: A batch size of 32 was used in the final model to make better usage of the GPU I was allowed to train my model on. Utilising the GPU's parallel processing hardware allowed for a more efficient training phase.

Epochs: During the training stage, the number of epochs was originally set to three, but this led to overfitting and poor performance on the validation set. Reducing the epochs to one resulted in significant improvement in performance on the validation set. This performance aligned closely with the similar model from Aveni, enabling progression to testing the model on the unseen test dataset.

Parameter	Value
Learning Rate	2×10^{-4}
Batch Size	32
Epochs	1
Threshold	0.5

Table 3.5: RoBERTa Model Parameters

3.9.2.3 Testing & Metrics:

The RoBERTa model was evaluated on the Aveni test dataset with parameters from Table 3.5, showing results in Table 3.6. Our RoBERTa model test metrics fell slightly short of Aveni Labs' RoBERTa model, which achieved an F-1 score of 80%. This creates a limitation as the model is weaker than Aveni's. However, the decision to proceed was made due to the considerable time invested in its design and the thesis scope. Our RoBERTa model is still high-performing and serves as a robust upper baseline for comparisons.

Metric	Value
Accuracy	89.7%
Precision	79.0%
Recall	73.5%
F-1 Score (Positive Class)	**76.1%**

Table 3.6: RoBERTa Test Metrics

3.9.3 Zero-Shot and Few-Shot Learning

The initial experiments conducted in this thesis also serve as a baseline to help evaluate GPT-4's performance as a vulnerability classifier. This foundation is crucial for bench-

marking improvements in subsequent experiments with CoT Prompting and Prompt Optimisation to assess how these methods enhance GPT-4's performance. Both of these baselines are evaluated on the Aveni test dataset.

3.9.3.1 Zero-Shot Learning

The first experiment employed GPT-4 as a binary vulnerability classifier, using a zero-shot learning approach, meaning that no examples were provided in the meta-prompt to help steer the model's output. By not providing examples within the prompt, the experiment focuses solely on assessing the model's innate classification ability without the influence of In-Context Learning. This baseline enables a clear comparison of performance improvements when leveraging the GPT-4's In-Context Learning capabilities through the addition of examples.

3.9.3.2 Few-Shot Learning

This baseline experiment marks the initiation of utilising GPT-4's In-Context Learning capabilities and permits comparisons between future experiments with CoT Prompting and Prompt Optimisation. By employing a few-shot learning approach, exemplars are incorporated into the meta-prompt for the first time. At this stage, examples, which are randomly drawn from the training dataset, are only annotated with their binary answers (Yes/No) without reasoned explanations, setting the stage for comparison with CoT Prompting. This baseline experiment used an increasing number of examples at each stage. We experimented with 5,8,10 and 12 examples here to see what produced the best performance. The outcomes informed the selection of example quantities for the CoT Prompting experiments, optimising costs. Few-shot experiments proved more cost-effective, benefiting from smaller context windows due to the exclusive use of Yes/No annotated examples.

3.9.4 Chain of Thought Baseline

As our methodology behind CoT Prompting seeks to explore various example strategies that could improve its performance, it was essential to establish a CoT Prompting baseline representing a standard CoT Prompting experiment. Like other research in CoT Prompting where examples provided are drawn from an example 'bank' with no specific strategy in their selection (Suzgun et al. 2022, Jason Wei et al. 2022), our CoT Prompting baseline uses examples from the training dataset that were selected completely at random.

The baseline dataset comprises of 25 examples with their corresponding hand-crafted CoT explanations. This baseline facilitates comparisons between the few-shot learning baseline discussed above, allowing for insights into general CoT performance. Additionally, the baseline acts as a benchmark to explore which CoT Prompting strategies lead to performance enhancements beyond the standard approach, directly addressing the question of identifying effective CoT Prompting strategies within this problem domain. There is another limitation on the size of this dataset due to the time-consuming

nature of hand-constructing CoT examples. However, it still serves as a strong baseline representative of a standard CoT Prompting experiment.

3.9.5 Prompt Optimisation Baseline

In order to evaluate EdiPrompt effectively, OPRO (Yang et al. 2023) was selected as the baseline for comparison. As EdiPrompt aims to build upon EvoPrompt (Guo et al. 2023) it would seem natural to use this as a baseline. However, EvoPrompt requires a large set of initial prompts and generates multiple prompts, making it infeasible to adapt due to budget constraints. Conversely, OPRO generates a single prompt per iteration and outperformed EvoPrompt in two math problem datasets (Yang et al. 2023), demonstrating its efficiency and cost-effectiveness, making it an ideal baseline for evaluating EdiPrompt. As discussed, OPRO utilises the LLM as a prompt optimiser through iterative updates to the meta-prompt with instruction-score pairs (model-generated prompts and their corresponding 'scores' based on a fitness function). An OPRO overview can be seen in Figure 3.12.

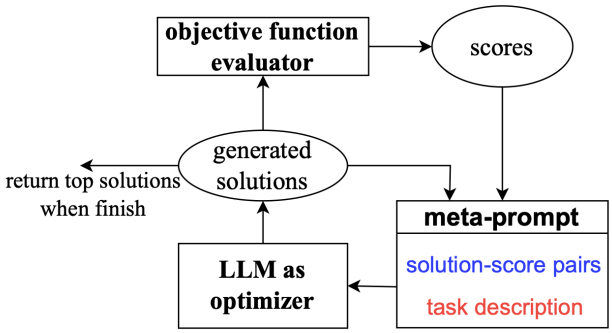

Figure 3.12: OPRO Framework Overview (Yang et al. 2023)

To enable comparisons, OPRO was adapted to the vulnerability classification domain by modifying its meta-prompt content while maintaining its original structure and core principles. The original OPRO meta-prompt and the adapted one for this thesis can be seen in Appendix A. The fitness function for OPRO is now the F-1 score of its generated prompts when tested on the Aveni test dataset.

The adapted OPRO meta-prompt contains the instruction-score pairs in which the maximum number of these pairs in the meta-prompt at any given time is five. This design choice was made as a result of OPRO being implemented on a smaller scale, as the original paper provided a maximum of eight. However, the original paper was done with hundreds of iterations, so reducing this to five was justified as it also kept our context window small, reducing costs when querying GPT-4. By utilising an equal number of instruction-score pairs for both the OPRO baseline and EdiPrompt, more robust conclusions can be drawn, allowing for the evaluation of the respective frameworks based on their underlying principles. OPRO also includes problem examples in its meta-prompt to enable the model to familiarise itself with the problem domain.

For consistency and fairness in comparison, both OPRO and EdiPrompt are run for

eight iterations and their generated prompts are evaluated under the same zero-shot learning scenario, as discussed in Section 3.7.1.2. Both frameworks also use the same starting prompt to compare both frameworks equally after all iterations. The instruction prompt used as $P_{initial}$ in the first iteration of EdiPrompt was also used as the instruction prompt in the first instruction-score pair for OPRO.

3.10 Budget

Aveni Labs provided a generous £1400 budget for this thesis, which determined what experiments were repeated for robustness. In general, a zero-shot learning experiment is roughly £6, and an experiment with 12 in-context examples is roughly £35. The latter is due to an increase in the context window when querying GPT-4 through its API, which incurs a greater cost. To focus on the core research areas in this thesis and manage the budget correctly, only the CoT Prompting experiments and the final experiment of combining CoT Prompting and Prompt Optimisation were repeated. This allowed for robustness in the CoT Prompting experiments and for the EdiPrompt and OPRO frameworks to have the resources to be run for eight iterations each. Generated prompts in Prompt Optimisation were tested only once to avoid the inefficiency of retesting average or low-performing prompts, reserving resources for the final experiment with the strongest prompt identified.

The selected experiments discussed above are repeated three times, a decision driven by two primary factors. Firstly, budgetary constraints discussed and secondly, the robustness of GPT-4 itself. The advanced design and extensive training of GPT-4 ensure consistent performance (OpenAI 2023, Hackl et al. 2023), reducing the need for numerous repetitions to establish reliable results. This robustness also reduces the limitations of results for experiments that were only repeated once. Budget constraints do limit this thesis. However, the reliability of GPT-4's responses and repeated core experiments significantly mitigate this limitation. Now, with a complete understanding of our methodology for this thesis, we proceed to our results and discussion.

Chapter 4

Results & Discussion

In this chapter we aim to answer the question of whether GPT-4 is capable of outperforming our fine-tuned RoBERTa model on vulnerability classification, and we investigate how effective CoT Prompting and Prompt Optimisation are at enhancing GPT-4's vulnerability classification performance. Our analysis starts with GPT-4's performance in zero-shot and few-shot scenarios, followed by examinations of CoT Prompting and Prompt Optimisation, finishing with the evaluation of the naive combination of these techniques.

All experiments use the Aveni test dataset, test conversations are inserted one at a time into the corresponding meta-prompt tailored for each specific experiment. We compare true labels against GPT-4's classification to calculate F-1 scores (on the vulnerable class). As discussed previously, any examples used in approaches like CoT Prompting or few-shot learning are not from this test set to prevent bias. In their respective sections, we present the average F-1 scores from repeated experiments. We omit reporting the standard deviation as it is minimal due to GPT-4's previously discussed consistency.

4.1 Zero-Shot and Few-Shot Learning Results

We begin by evaluating GPT-4's performance in zero-shot and few-shot learning scenarios against RoBERTa in order to evaluate its initial performance before CoT Prompting or Prompt Optimisation. As a reminder, when we mention few-shot learning, we are talking about providing answer-only annotated examples to the model; these examples were randomly sampled from the training dataset. From Table 4.1, we can see the strength of In-Context Learning, with 10 examples boosting GPT-4's F-1 score by 19.6% compared to zero-shot. As the number of examples increases in few-shot, the plateau of performance can also be observed. Despite the improvements observed when leveraging In-Context Learning, GPT-4 has underperformed against RoBERTa in all of these cases. The best GPT-4 few-shot experiment with 10 examples fell short of RoBERTa's F-1 score by 9%.

Considering the publicity and praise that GPT-4 has received, it may be surprising that RoBERTa, being a smaller model, has outperformed it by a significant margin. However,

Model	Examples Used	F-1 Score (%)
Keyword Classifier	N/A	39.4
GPT-4 (Zero-Shot)	0	47.5
GPT-4 (Few-Shot)	5	61.9
GPT-4 (Few-Shot)	8	62.9
GPT-4 (Few-Shot)	10	67.1
GPT-4 (Few-Shot)	12	66.7
RoBERTa	**N/A**	**76.1**

Table 4.1: Comparison of GPT-4 with zero-shot and few-shot learning against RoBERTa.

vulnerability classification is a difficult task that presents many edge cases and involves considerable ambiguity. Tasks of this nature usually tend to be better addressed by fine-tuning a model on task-specific data, as it allows for a model to be tailored to the specific task, in our case, we tailor RoBERTa to vulnerability classification. As GPT-4 remains task-agnostic, we attempt to fine-tune it through In-Context Learning, but this has not been enough to match RoBERTa's performance. Not all tasks have appropriate datasets or labelled examples to fine-tune a model like RoBERTa, so finding ways to enhance GPT-4's performance remains a crucial task, and we aim to do this through CoT Prompting and Prompt Optimisation. We proceed by first analysing our CoT Prompting results.

4.2 Chain of Thought Prompting Results

In this section, we explore the effectiveness of CoT Prompting in enhancing GPT-4's vulnerability classification performance. We aim to build upon the results achieved by few-shot learning with answer-only examples. We also aim to answer the research question surrounding what strategies, in terms of the examples used, yield the best performance from CoT Prompting.

As a reminder, our CoT Prompting methodology involved investigating different example strategies, looking at the effect of example quality, likeness to target data and example polarity through two example sources (misclassified and guidelines). We conducted 9 CoT Prompting experiments, evaluating each on the Aveni test dataset. Each test conversation received 12 randomly sampled in-context CoT examples from the example source and example polarity (vulnerable or non-vulnerable) from that source for its respective experiment.

Table 4.2 displays the results of all 9 CoT Prompting experiments for each respective strategy, where the F-1 score displayed is the average after three repetitions. This table also displays the few-shot learning baseline we aim to improve upon alongside the CoT baseline. As previously discussed, the CoT baseline adopts random sampling of CoT examples taken from the training set to represent a standard CoT Prompting experiment and enable comparisons between the different strategies. We can initially observe in Table 4.2 that the CoT baseline has underperformed against the few-shot learning baseline, with a decrease of 3.3% in its F-1 score. This finding initially suggests that for

vulnerability classification, providing examples to GPT-4 is more important than the explanations attached to them, and that providing CoT explanations does not enhance GPT-4's performance.

We also notice that CoT Prompting was outperformed by the few-shot learning baseline with the same number of examples in 7 out of 9 experiments, indicating that CoT Prompting does not bring substantial gains to vulnerability classification even with tailored example strategies. This contrasts previous research on logical reasoning tasks where CoT Prompting showed large improvements over answer-only examples (Suzgun et al. 2022). Unlike logical reasoning tasks where CoT Prompting can guide the model through logical steps to a solution, vulnerability classification demands complex common-sense reasoning to understand the possible exceptions and conditions for classification. This complexity makes it challenging to effectively teach GPT-4 with CoT explanations, resulting in a situation where these explanations do not improve its performance beyond that achieved with answer-only examples. Our findings show that generalising CoT Prompting to complex domains like vulnerability classification is challenging, and it does not achieve the same improvements seen in logical reasoning tasks (Jason Wei et al. 2022, Suzgun et al. 2022). These findings provide direction for a wide range of tasks that seek to improve LLM's text classification performance in contexts requiring complex common-sense reasoning to understand specifications for classification, as we have shown that examples are more important than explanations in these contexts.

While the general findings indicate CoT Promptings ineffectiveness, it did show some promise. Table 4.2 shows that 2 of the CoT Prompting experiments (CoT 3 & CoT 9) did slightly outperform the few-shot learning baseline with the same number of examples, with CoT 9 leading by a 1% improvement. Despite marginal improvements, it is important to note that the size of the CoT dataset was only 100, compared to the training dataset size of 4663, from which the few-shot baseline could pull examples. This stark contrast in the size of examples that CoT Prompting could use creates the need for further research as these gains could potentially be heightened if CoT Prompting had a larger, more diverse number of instances to use in its examples.

We now look to answer our research question of what CoT Prompting example strategies yield the best performance. Table 4.3 and 4.4 show the average F-1 scores achieved when each specific strategy was utilised. For evaluating example quality, Table 4.3 shows that the clean guideline examples slightly outperform the noisy, misclassified ones by 0.6% on average. The misclassified examples did not enhance performance, maintaining the CoT baseline F-1 score at 63.4% on average, contrasting previous research that found this approach boosted performance in few-shot contexts (Gao et al. 2023). This contrast is surprising as providing previously misclassified instances as examples aims to allow GPT-4 to learn the difficult edge cases of vulnerability detection. However, we applied this approach in a CoT Prompting context, which suggests that the ineffectiveness of the misclassified examples could stem from the CoT Prompting method itself, which we have found to be less effective than answer-only examples. Overall, we see that the differing example sources in CoT Prompting do not significantly affect GPT-4's performance in vulnerability classification, showcasing its capability to generalise effectively from synthetic guideline examples, which differ significantly

GPT-4 Model	Example Source	Example Polarity	F-1 Score (%)
Few-Shot	Training Dataset	Random Mix	66.7
CoT Baseline	Training Dataset	Random Mix	63.4
CoT 1	50/50 M&G	50/50 V&NV	64.4
CoT 2	50/50 M&G	V	59.0
CoT 3	50/50 M&G	NV	67.1
CoT 4	G	50/50 V&NV	65.7
CoT 5	G	V	62.1
CoT 6	G	NV	64.2
CoT 7	M	50/50 V&NV	63.8
CoT 8	M	V	58.7
CoT 9	**M**	**NV**	**67.7**

Table 4.2: Chain of Thought Experiment Results. M = Misclassified source, G = Guideline source, V = Vulnerable Instances, NV = Non-Vulnerable Instances. 50/50 implies an even sampling split. All experiments here use 12 in-context examples. The best performing experiment is highlighted in **bold**.

Example Source	Average F-1 Score (%)
50/50 M&G	63.5
M	63.4
G	**64.0**

Table 4.3: Comparison of Strategies by Example Source. M = Misclassified source, G = Guideline source. The best performing strategy on average is in **bold**.

Example Polarity	Average F-1 Score (%)
50/50 V&NV	64.6
V	59.9
NV	**66.3**

Table 4.4: Comparison of Strategies by Example Polarity. V = Vulnerable Instances, NV = Non-Vulnerable Instances. The best performing strategy on average is in **bold**.

from the actual conversations targeted for classification.

In terms of example polarity, Table 4.4 illustrates an unexpected shift in performance when exclusively using vulnerable examples, leading to a significant 3.5% decrease from the CoT baseline. This contradicts the initial belief that focusing solely on vulnerable instances might enhance GPT-4's understanding of vulnerability. In contrast, using strictly non-vulnerable instances as CoT examples achieves the most notable performance improvement, with an average F-1 score increase of 6.4% over vulnerable instances and 2.9% over the CoT baseline. These findings illustrate that using only non-vulnerable instances improves GPT-4's ability to classify true vulnerabilities by

clearly defining what non-vulnerability looks like, making it easier to detect actual vulnerabilities. Our novel research into example strategies allows us to conclude that the use of strictly non-vulnerable instances is the optimal CoT Prompting example strategy for enhancing GPT-4's vulnerability classification performance. Although these example strategies were carried out in a CoT Prompting context, our findings motivate future research that would utilise the optimal example strategy found in this research to further enhance GPT-4 as a vulnerability classifier in other contexts, such as answer-only few-shot learning, which has proved to be superior to CoT Prompting in our domain.

Our investigation highlights the influence of example source and polarity in CoT Prompting but faces a limitation, our random sampling may not always yield examples relevant to each test case, given the broad spectrum of vulnerabilities. Future research could improve CoT Prompting by selecting examples more relevant to the test conversation, perhaps by adopting a retriever model to identify semantically similar examples, building on approaches from previous studies (J. Liu et al. 2021).

We looked to see if CoT Prompting could enhance GPT-4's vulnerability classification performance near our fine-tuned RoBERTa. However, it was found that answer-only examples are superior to explanations in our context, with the best CoT Prompting experiment attaining an F-1 score 8.4% shy of RoBERTa. We now proceed to see if Prompt Optimisation can be more effective than CoT Prompting in enhancing GPT-4's vulnerability classification performance.

4.3 Prompt Optimisation Results

We now look to Prompt Optimisation to answer our research question of how effectively can Prompt Optimisation enhance GPT-4 as a vulnerability classifier. We also aim to answer whether the Genetic Algorithms within EdiPrompt can improve prompt linguistic diversity and performance compared to its OPRO baseline. Our methodology for Prompt Optimisation looked to optimise our human-generated meta-instruction, which all previous experiments used to instruct GPT-4 to classify a conversation. We ran both EdiPrompt and OPRO for eight iterations, evaluating each generated instruction prompt in a zero-shot learning environment on classifying conversations from the Aveni test dataset.

In analysing EdiPrompt, we focus on evaluating the mutated instruction prompts generated after a complete iteration, as this displays EdiPrompt's overall effectiveness. All instruction prompts generated by OPRO and EdiPrompt at each iteration can be found in Appendix B. Figure 4.1 displays the best instruction prompt that EdiPrompt and OPRO generated. It also provides the original human-generated instruction prompt we aimed to optimise for comparison. We see that both of these prompts contain more complex language and reiterate our definition of vulnerability. This results from including this definition in the meta-prompts for both frameworks where we describe our optimisation task. This indicates that GPT-4 performs better with prompts using sophisticated language and clear task definitions, likely due to a richer vocabulary that enables more precise communication, therefore improving vulnerability classification.

The best-performing prompt, found by EdiPrompt, displays an F-1 score of 71.4%. This instruction prompt outperforms all few-shot learning baselines and CoT Prompting experiments, with the prompt providing a 3.7% increase in the F-1 score compared to the previous best experiment (CoT 9). Remarkably, this score was achieved without in-context examples, as all prompts were tested in a zero-shot environment, demonstrating the prompt's effective design to harness GPT-4's pre-training knowledge. This is a significant advancement in using GPT-4 as a vulnerability classifier, highlighting Prompt Optimisation's effectiveness over example-driven methods for our problem domain and demonstrating that for complex tasks requiring understanding of specifications for classification, clear task-descriptive prompts yield better results than providing examples.

Framework	Prompt	F-1 Score (%)
Human-Generated Prompt	*From the above conversation, is the customer vulnerable based on our definition of vulnerability, only answer Yes or No, and you must say only Yes or No.*	*47.5*
EdiPrompt (Mutated) - Best	**Analyze the dialogue meticulously for undeniable signs that the customer may be vulnerable, in line with our criteria: significant health issues that impede daily routines, profound life events, resilience in the face of hardships, or capability in financial management and other pivotal areas. Provide a succinct 'Yes' if such signs are evident, or a 'No' otherwise, drawing strictly from the concrete evidence in the conversation. Your answer must solely consist of 'Yes' or 'No'.**	**71.4**
OPRO - Best	Review the customer's dialogue for definitive references to health complications impacting routine activities, marked life events like loss of employment or personal upheaval, tangible signs of severe emotional or financial distress, or explicit difficulties with fundamental financial, literacy, or digital skills. Confirm vulnerability with a 'Yes' if manifest; if absent, provide a 'No'. Emit an unambiguous 'Yes' or 'No' verdict accordingly.	70.3

Figure 4.1: EdiPrompt & OPRO best instruction prompts. The initial human-generated instruction prompt is supplied in *italics*. The best instruction prompt overall is in **bold**.

Figure 4.2 shows that EdiPrompt consistently outperforms OPRO across the eight iterations, except for slight underperformance in iterations 1 and 8. The graph highlights EdiPrompt's stable improvement and OPRO's fluctuating performance, demonstrating EdiPrompt's effective optimisation of prompts due to its iterative updates and design. Table 4.5 provides an overview of EdiPrompt against OPRO, and we can see that EdiPrompt has surpassed OPRO in every category. The average F-1 score from all of the generated mutated prompts from EdiPrompt is 67.9%, which is a 3.8% increase from the 64.1% average of OPRO.

We also provide the standard deviation across the F-1 scores of each frameworks generated prompts, where EdiPrompt displays a smaller standard deviation than OPRO. OPRO's variability may stem from its meta-prompts lacking direction and its method of

generating only one prompt per iteration, aligning with observations from the original OPRO paper about its performance oscillation (Yang et al. 2023). EdiPrompts consistent performance is likely due to the novel design choice of keeping the best prompt found as one of the two prompts being combined at every iteration, ensuring a high-quality combination baseline. The average F-1 score and standard deviation of EdiPrompt show that it has not only produced better-performing prompts than OPRO but also does so with more consistency.

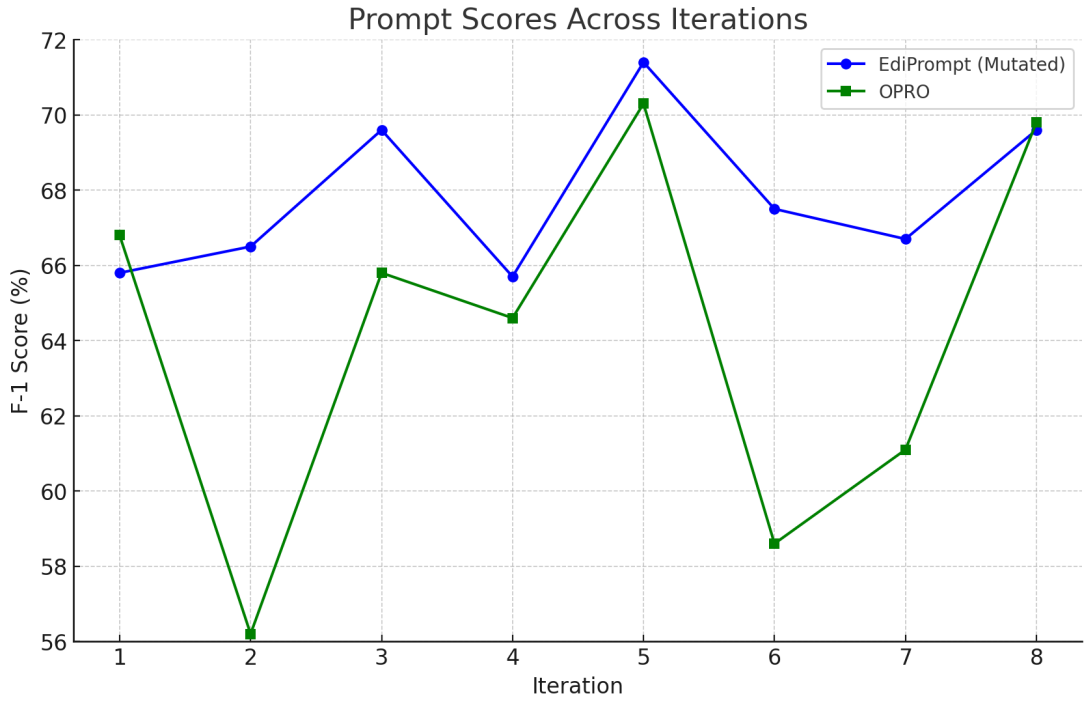

Figure 4.2: EdiPrompt vs OPRO - Prompt Scores Across Iterations

Table 4.5 shows that the best prompt from EdiPrompt produced a relative percentage increase of 50.3% in performance compared to our original human-generated prompt with an F-1 score of 47.5%. Finally, we look at the Self-Bleu metric, which ranges between 0 (high diversity) and 1 (low diversity). We see that the inherent variability of Genetic Algorithms in EdiPrompt has successfully uncovered more linguistically diverse prompts than OPRO over the eight iterations. This variability, stemming from varied vocabulary, phrases, and writing styles through the combination and mutation of prompts, uncovers more effective task communication to GPT-4, boosting EdiPrompt's prompt performance. Despite EdiPrompt's improvement over OPRO, it is worth noting that OPRO remains competitive. Its top prompt trails EdiPrompt's by just 0.9%, and its single-prompt generation approach offers cost efficiency, making it a viable choice.

EdiPrompt's strong performance confirms that Genetic Algorithms can be powerful within Prompt Optimisation, as initially explored in EvoPrompt (Guo et al. 2023). It displays that EdiPrompt has successfully harnessed the power and extra variability of Genetic Algorithms, establishing a robust new Prompt Optimisation framework. EdiPrompt's novel design choices and refined meta-prompts have likely contributed to its success, as EvoPrompt failed to surpass OPRO (Yang et al. 2023), whereas

EdiPrompt has initially shown its superiority to OPRO. However, as this research focused on investigating if EdiPrompt could successfully harness Genetic Algorithms, further research is needed on EdiPrompt's specific concepts to assess the full potential of the frameworks design.

While these results are impressive, it is crucial to understand that running EdiPrompt for only eight iterations is substantially small compared to the scale displayed in previous research, such as OPRO (Yang et al. 2023), which was done over hundreds of iterations. This motivates future research to explore EdiPrompt's performance at a larger scale and across different domains.

Metric	OPRO	EdiPrompt
Average F-1 Score (Across all 8 iterations)	64.1%	**67.9%**
Highest scoring prompt (F-1 Score)	70.3%	**71.4%**
Lowest scoring prompt (F-1 Score)	56.2%	**65.7%**
Standard Deviation	0.048	**0.019**
Self-Bleu	0.148	**0.112**
Best Relative Percentage Increase on the F-1 Score of Human Generated Prompt	48.0%	**50.3%**

Table 4.5: Overview of OPRO vs EdiPrompt. **Bold** indicates where one framework has surpassed the other.

Our findings from Prompt Optimisation have answered both related research questions. Prompt Optimisation was more effective than CoT Prompting in enhancing GPT-4 as a vulnerability classifier, achieving an F-1 score only 4.7% shy of RoBERTa. We have also demonstrated that the Genetic Algorithms within EdiPrompt have produced more diverse prompts and ultimately higher-performing prompts than our OPRO baseline on our vulnerability classification task. We now see if we can naively combine CoT Prompting and Prompt Optimisation to further enhance GPT-4 as a vulnerability classifier compared to our fine-tuned RoBERTa model.

4.4 Chain of Thought Prompting & Prompt Optimisation Results

Our final experiment naively combined the highest-performing instruction prompt from EdiPrompt with the most effective CoT Prompting strategy, CoT 9. This CoT Prompting strategy specifically utilised examples of non-vulnerability from the misclassified data source. This experiment aims to answer our research question of how effective the combination of these areas is at enhancing GPT-4 as a vulnerability classifier. We can see the results of the final experiment in Table 4.6, where the F-1 score for the combination of the two areas is the average after three repetitions.

Table 4.6 displays each of the scores for the CoT strategy and the best prompt for comparison. We see that adding the best instruction prompt into the best CoT strategy,

Experiment/Baseline	**F-1 Score (%)**
RoBERTa Baseline	76.1
CoT 9: Misclassified, Non-Vulnerable	67.7
Best-Instruction Prompt	71.4
CoT 9: Misclassified, Non-Vulnerable + Best Instruction Prompt	70.1

Table 4.6: Prompt Optimisation & Chain of Thought Prompting Results

which was initially carried out with the human-generated instruction prompt, has increased the performance of the best CoT experiment by 2.4%. However, the addition of the CoT examples has decreased the original performance of the prompt, which was previously tested in a zero-shot environment, by 1.3%.

This finding suggests that while the CoT Prompting strategy benefits the optimised prompt, adding examples to an effective prompt can unexpectedly reduce performance. This decline in performance may be due to the inefficacy of CoT explanations that has been observed in this thesis. The explanations may have introduced unnecessary complexity, shifting the model's focus from the well-crafted prompt. Another reason the combination of both of these areas has not succeeded could be that prompts were optimised without examples in mind. Due to budget and time constraints, we could not re-optimise the prompt to provide explicit instruction to GPT-4 to use the provided CoT examples for help. We hypothesise that the effectiveness of the combination could be further enhanced if this was done, motivating the need for future research.

Table 4.6 allows us to answer our research question as it displays that we could not combine CoT Prompting and Prompt Optimisation to further enhance GPT-4 as a vulnerability classifier in this research. However, Prompt Optimisation alone was most effective, nearly matching our fine-tuned RoBERTa model's performance with an F-1 score just 4.7% lower, even without In-Context Learning. This result is crucial because it shows that GPT-4 does not require labelled data to function as a strong vulnerability classifier when it uses effective Prompt Optimisation. This is particularly vital in real-world applications where labelled data is scarce, as we can quickly adapt GPT-4 to accurately identify individuals requiring support. This adaptability means that environments with limited labelled resources can benefit from the deployment of new services with enhanced support, reducing risks for those in need. Importantly, for tasks such as vulnerability classification that present complexity with intricate exceptions and conditions for classification, our results display that for such tasks where labelled data is scarce, GPT-4 combined with Prompt Optimisation offers a viable solution. Conversely, in such tasks where appropriate labelled data is available, our research into CoT Prompting and Prompt Optimisation demonstrates that fine-tuning a model like RoBERTa remains the optimal approach. This insight provides a clear direction for future efforts, suggesting a tailored approach depending on the data availability and specific complexity of the task at hand.

Chapter 5

Conclusions & Further Research

This thesis has successfully researched how we can leverage GPT-4 as a vulnerability classifier in two complementary areas of research, Chain of Thought Prompting and Prompt Optimisation, and answered all related research questions.

For Chain of Thought Prompting, we explored its effectiveness in new domains, specifically vulnerability classification, and carried out novel research into how different example types impact its performance. We found that the optimal example strategy was supplying strictly non-vulnerable instances, which improved upon few-shot learning with answer-only examples by 1% in a specific experiment. However, Chain of Thought Prompting ultimately did not enhance GPT-4's performance as a vulnerability classifier, with few-shot learning using answer-only examples proving superior. We illustrate that for tasks such as vulnerability classification, which demand complex common-sense reasoning to understand the conditions and exceptions for classification, providing GPT-4 with examples is more beneficial than providing explanations. This result provides strong insights for future research looking to leverage LLMs in similar domains.

In Prompt Optimisation, we explored its potential in enhancing GPT-4 as a vulnerability classifier and proposed the novel EdiPrompt framework. We found that EdiPrompt effectively harnessed the variability of Genetic Algorithms and created a new robust Prompt Optimisation framework that outperformed Google DeepMind's OPRO framework in both prompt performance and prompt linguistic diversity. EdiPrompt generated a prompt that boasts a 50.3% relative increase in performance compared to the original human-generated prompt. Prompt Optimisation brought GPT-4 the closest to RoBERTa's performance, without In-Context Learning, missing the baseline by only 4.7%, thus proving Prompt Optimisation's effectiveness over Chain of Thought Prompting in our domain.

We also looked to see if the naive combination of Chain of Thought Prompting and Prompt Optimisation could further enhance GPT-4 as a vulnerability classifier past that of RoBERTa. It was found that the inclusion of the most effective Chain of Thought strategy into the highest-performing instruction prompt unexpectedly decreased the prompt's original performance. We hypothesised that the observed ineffectiveness is due to our naive combination that did not optimise prompts to account for the inclusion

of examples. We could not re-optimise prompts to do this due to budget and time constraints, suggesting areas for future research.

In conclusion, our research showed that Prompt Optimisation closes the performance gap between GPT-4 and fine-tuned models, such as RoBERTa, in complex tasks like vulnerability classification, even without labelled data. This positions it as a feasible alternative. This is crucial for tasks in real-world scenarios with complex specifications for classification and limited labelled data, allowing quick adaptation of GPT-4 for support services. However, when labelled data is available, our research has shown that fine-tuning a model like RoBERTa remains the optimal approach. These findings pave the way for tailored strategies in future research and applications, depending on data availability and task complexity.

5.1 Further Research

We finish this thesis by outlining potential avenues for further research to deepen the exploration of Chain of Thought Prompting and Prompt Optimisation. This further research would extend the findings and address the limitations of this thesis.

A key limitation of this thesis is its exploration of Chain of Thought Prompting and Prompt Optimisation on only one model, GPT-4, a large-scale LLM. Further research would examine how these techniques perform on smaller models, such as 1B or 7B LLMs, to determine if the observed results hold consistent across varying model sizes. This would provide a more comprehensive understanding of the effectiveness and scalability of these techniques.

Further research into Chain of Thought Prompting is needed, comparing its performance with answer-only few-shot learning using datasets of equal size, addressing our small Chain of Thought dataset limitation. This could be achieved by eliminating the time-consuming nature of constructing Chain of Thought examples, possibly by using a LLM to create them. Furthermore, our results that found examples were superior to Chain of Thought explanations for complex tasks like vulnerability classification elicits future research to explore a broader spectrum of tasks to investigate what types of tasks benefit from explanations over answer-only examples.

This research highlights EdiPrompt's potential in vulnerability classification but acknowledges the limitation that it underwent fewer iterations than prior studies on new Prompt Optimisation frameworks. Future work should increase EdiPrompt's iterations and apply it across various domains to thoroughly evaluate its potential and versatility in adapting to an array of problems. Additionally, future research in Prompt Optimisation should also look at how to deliver strong prompts in a more cost-effective manner and assess whether it is more effective to explore a broad or a deeper search of prompts.

Our naive combination of Chain of Thought Prompting and Prompt Optimisation in this thesis found an unexpected decrease in performance when adding Chain of Thought examples to the optimised prompt. Further research would aim to optimise instruction prompts with examples in mind to investigate whether we can improve results when combining Prompt Optimisation and Chain of Thought Prompting.

Bibliography

Alzubaidi, Laith et al. (2021). "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". In: *Journal of big Data* 8, pp. 1–74.

Brown, Tom B. et al. (2020). "Language Models are Few-Shot Learners". In: *CoRR* abs/2005.14165. arXiv: 2005.14165. URL: https://arxiv.org/abs/2005.14165.

Devlin, Jacob et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL].

Dong, Qingxiu et al. (2023). *A Survey on In-context Learning*. arXiv: 2301.00234 [cs.CL].

Gao, Lingyu et al. (2023). *Ambiguity-Aware In-Context Learning with Large Language Models*. arXiv: 2309.07900 [cs.CL].

Guo, Qingyan et al. (2023). *Connecting Large Language Models with Evolutionary Algorithms Yields Powerful Prompt Optimizers*. arXiv: 2309.08532 [cs.CL].

Hackl, Veronika et al. (Dec. 2023). "Is GPT-4 a reliable rater? Evaluating consistency in GPT-4's text ratings". In: *Frontiers in Education* 8. ISSN: 2504-284X. DOI: 10.3389/feduc.2023.1272229. URL: http://dx.doi.org/10.3389/feduc.2023.1272229.

Kojima, Takeshi et al. (2022). "Large Language Models are Zero-Shot Reasoners". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., pp. 22199–22213. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

Lester, Brian, Rami Al-Rfou, and Noah Constant (2021). *The Power of Scale for Parameter-Efficient Prompt Tuning*. arXiv: 2104.08691 [cs.CL].

Li, Qian et al. (2021). *A Survey on Text Classification: From Shallow to Deep Learning*. arXiv: 2008.00364 [cs.CL].

Lipowski, Adam and Dorota Lipowska (2012). "Roulette-wheel selection via stochastic acceptance". In: *Physica A: Statistical Mechanics and its Applications* 391.6, pp. 2193–2196.

Liu, Jiachang et al. (2021). "What Makes Good In-Context Examples for GPT-3?" In: *CoRR* abs/2101.06804. arXiv: 2101.06804. URL: https://arxiv.org/abs/2101.06804.

Liu, Pengfei et al. (2023). "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing". In: *ACM Computing Surveys* 55.9, pp. 1–35.

Liu, Yinhan et al. (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *CoRR* abs/1907.11692. arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

Markov, Todor et al. (2023). *A Holistic Approach to Undesired Content Detection in the Real World*. arXiv: 2208.03274 [cs.CL].

Minaee, Shervin et al. (2021). *Deep Learning Based Text Classification: A Comprehensive Review*. arXiv: 2004.03705 [cs.CL].

Mitchell, Melanie (1998). *An introduction to genetic algorithms*. MIT press.

Nye, Maxwell I. et al. (2021). "Show Your Work: Scratchpads for Intermediate Computation with Language Models". In: *CoRR* abs/2112.00114. arXiv: 2112.00114. URL: https://arxiv.org/abs/2112.00114.

OpenAI (2023). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL].

Papineni, Kishore et al. (2002). "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318.

Pryzant, Reid et al. (2023). *Automatic Prompt Optimization with "Gradient Descent" and Beam Search*. arXiv: 2305.03495 [cs.CL].

Qiu, XiPeng et al. (Sept. 2020). "Pre-trained models for natural language processing: A survey". In: *Science China Technological Sciences* 63.10, pp. 1872–1897. DOI: 10.1007/s11431-020-1647-3. URL: https://doi.org/10.1007%2Fs11431-020-1647-3.

Radford, Alec and Karthik Narasimhan (2018). "Improving Language Understanding by Generative Pre-Training". In: URL: https://api.semanticscholar.org/CorpusID:49313245.

Radford, Alec, Karthik Narasimhan, et al. (2018). "Improving language understanding by generative pre-training". In.

Röttger, Paul et al. (2021). "HateCheck: Functional Tests for Hate Speech Detection Models". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.acl-long.4. URL: https://doi.org/10.18653%2Fv1%2F2021.acl-long.4.

Shah, Foram P. and Vibha Patel (2016). "A review on feature selection and feature extraction for text classification". In: *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 2264–2268. DOI: 10.1109/WiSPNET.2016.7566545.

Shin, Taylor et al. (2020). *AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts*. arXiv: 2010.15980 [cs.CL].

Suzgun, Mirac et al. (2022). *Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them*. arXiv: 2210.09261 [cs.CL].

Vaswani, Ashish et al. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang, Yaqing et al. (2020). "Generalizing from a few examples: A survey on few-shot learning". In: *ACM computing surveys (csur)* 53.3, pp. 1–34.

Wei, Jason et al. (2022). "Chain of Thought Prompting Elicits Reasoning in Large Language Models". In: *CoRR* abs/2201.11903. arXiv: 2201.11903. URL: https://arxiv.org/abs/2201.11903.

Wei, Jerry et al. (2023). *Larger language models do in-context learning differently*. arXiv: 2303.03846 [cs.CL].

White, Jules et al. (2023). *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*. arXiv: 2302.11382 [cs.SE].

Yang, Chengrun et al. (2023). *Large Language Models as Optimizers*. arXiv: 2309.03409 [cs.LG].

Zhao, Tony Z. et al. (2021). *Calibrate Before Use: Improving Few-Shot Performance of Language Models*. arXiv: 2102.09690 [cs.CL].

Zhou, Yongchao et al. (2023). *Large Language Models Are Human-Level Prompt Engineers*. arXiv: 2211.01910 [cs.LG].

Zhu, Yaoming et al. (2018). "Texygen: A benchmarking platform for text generation models". In: *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 1097–1100.

Appendix A

Meta Prompts

A.1 Zero-Shot and Few-Shot Learning Meta-Prompts

Our definition of vulnerability refers to customers who, due to their personal circumstances, are especially susceptible to harm. All customers are at risk of becoming vulnerable and this risk is increased by characteristics of vulnerability related to 4 key drivers.

Capability - low knowledge of financial matters or low confidence in managing money (financial capability). Low capability in other relevant areas such as literacy, or digital skills

Resilience - low ability to withstand financial or emotional shocks.

Life events - life events such as bereavement, job loss or relationship breakdown.

Health - health conditions or illnesses that affect ability to carry out day-to-day tasks.

Conversation to Classify: [...*insert conversation*...]

From the above conversation, is the customer vulnerable based on our definiton of vulnerability, only answer Yes or No, and you must say only Yes or No.

Figure A.1: Zero-Shot learning baseline meta-prompt. Red is definition of vulnerability, green is the conversation to classify, and orange is meta-instruction. This is the same meta-prompt we use to evaluate model-generated prompts in Prompt Optimisation, replacing the model-generated instruction prompts with our human generated one.

Our definition of vulnerability refers to customers who, due to their personal circumstances, are especially susceptible to harm. All customers are at risk of becoming vulnerable and this risk is increased by characteristics of vulnerability related to 4 key drivers.

Capability - low knowledge of financial matters or low confidence in managing money (financial capability). Low capability in other relevant areas such as literacy, or digital skills

Resilience - low ability to withstand financial or emotional shocks.

Life events - life events such as bereavement, job loss or relationship breakdown.

Health - health conditions or illnesses that affect ability to carry out day-to-day tasks.

Below is [*number of examples*] examples of conversations with customers, where the customer is either vulnerable (yes) or not vulnerable (no), we have provided the answer to each example for help.

Example Conversation: [...*insert example conversation*...]
Answer: Yes/No

(...*More examples and answers*...)

Conversation to Classify: [...*insert conversation*...]

Using the provided examples above for help, is the customer from the above conversation vulnerable based on our definiton of vulnerability, only answer Yes or No, and you must say only Yes or No.

Figure A.2: Few-Shot learning baseline meta-prompt. Red is definition of vulnerability, purple are answer-only examples, green is the conversation to classify, and orange is meta-instruction.

A.2 Chain of Thought Prompting Meta-Prompts

Our definition of vulnerability refers to customers who, due to their personal circumstances, are especially susceptible to harm. All customers are at risk of becoming vulnerable and this risk is increased by characteristics of vulnerability related to 4 key drivers.

Capability - low knowledge of financial matters or low confidence in managing money (financial capability). Low capability in other relevant areas such as literacy, or digital skills

Resilience - low ability to withstand financial or emotional shocks.

Life events - life events such as bereavement, job loss or relationship breakdown.

Health - health conditions or illnesses that affect ability to carry out day-to-day tasks.

Below is [*number of examples*] examples of conversations with customers, where the customer is either vulnerable or not vulnerable, we have provided chain of thought reasoning to answer each example for help.

Example Conversation: [...*insert example conversation*...]
Answer: [...*insert Chain of Thought reasoning*...]

(...*More examples and Chain of Thought reasoning*...)

Conversation to Classify: [...*insert conversation*...]

Using the provided examples and their correct reasoned answers above for help, is the customer from the above conversation vulnerable based on our definiton of vulnerability, only answer Yes or No, and you must say only Yes or No.

Figure A.3: Chain of Thought Prompting meta-prompt. When carrying out the final experiment with the best instruction prompt from Prompt Optimisation and best Chain of Thought Prompting strategy, we use this meta-prompt and substitute the best instruction prompt for the meta-instruction seen here.

A.3 Prompt Optimisation - OPRO Meta-Prompts

Your task is to generate the instruction <INS>. Below are some previous instructions with their scores. The score ranges from 0 to 100.

text:
Let's figure it out!
score:
61

text:
Let's solve the problem.
score:
63

(. . . more instructions and scores . . .)

Below are some problems.

Problem:
Q: Alannah, Beatrix, and Queen are preparing for the new school year and have been given books by their parents. Alannah has 20 more books than Beatrix. Queen has 1/5 times more books than Alannah. If Beatrix has 30 books, how many books do the three have together?
A: <INS>

Ground truth answer:
140

(. . . more exemplars . . .)

Generate an instruction that is different from all the instructions <INS> above, and has a higher score than all the instructions <INS> above. The instruction should begin with <INS> and end with </INS>. The instruction should be concise, effective, and generally applicable to all problems above.

Figure A.4: Original OPRO Meta-Prompt (Yang et al. 2023)

A.4 Prompt Optimisation - EdiPrompt Meta-Prompts

Your task is to generate the instruction <INS>. Below are some previous instructions with their scores. The score is the f-1 score and ranges from 0 to 1.

Instruction: [...*insert instruction*...]
Score: [...insert instruction score...]

(..more instruction-score pairs...)

Below are examples of the problems, we are classifying customers conversations as vulnerable or not vulnerable. Our definition of vulnerability refers to customers who, due to their personal circumstances, are especially susceptible to harm. All customers are at risk of becoming vulnerable and this risk is increased by characteristics of vulnerability related to 4 key drivers:
Health - health conditions or illnesses that affect ability to carry out day-to-day tasks.
Life events - life events such as bereavement, job loss or relationship breakdown.
Resilience - low ability to withstand financial or emotional shocks.
Capability - low knowledge of financial matters or low confidence in managing money (financial capability). Low capability in other relevant areas such as literacy, or digital skills

Conversation: [...insert example conversation...]
<INS>
Answer: Yes/No

(...more problem examples...)

Generate an instruction that is different from all the instructions <INS> above, and has a higher score than all the instructions <INS> above. The instruction should be concise, effective, and generally applicable to the classifcation task at hand. The instruction must contain that we need a Yes or No response.

Figure A.5: Adapted OPRO Meta-Prompt.

Your task is to generate the instruction <INS>. Below are some previous instructions with their scores. The score is the f-1 score and ranges from 0 to 1. We want to maximise this score.

Instruction: [...*insert instruction*...]
Score: [...insert instruction score...]

(..more instruction-score pairs...)

Below are examples of the problems, we are classifying customers conversations as vulnerable or not vulnerable. Our definition of vulnerability refers to customers who, due to their personal circumstances, are especially susceptible to harm. All customers are at risk of becoming vulnerable and this risk is increased by characteristics of vulnerability related to 4 key drivers:
Health - health conditions or illnesses that affect ability to carry out day-to-day tasks.
Life events - life events such as bereavement, job loss or relationship breakdown.
Resilience - low ability to withstand financial or emotional shocks.
Capability - low knowledge of financial matters or low confidence in managing money (financial capability). Low capability in other relevant areas such as literacy, or digital skills

Conversation: [...insert example conversation...]
<INS>
Answer: Yes/No

(...more problem examples...)

Generate an instruction that is different from all the instructions above, and has a higher score than all the instructions above. The instruction should be concise, effective, and generally applicable to the classifcation task at hand. The instruction must contain that we need a Yes or No response.

Figure A.6: EdiPrompt Initial Instruction Meta-Prompt. This is used to initialise EdiPrompt as well, where the only instruction-score pair is the original human-generated instruction.

Here is the current instruction: [...insert initial instruction...]
Generate two new instructions that are semantically similar to the instruction above. The instruction should be concise, effective, and generally applicable to the classifcation task at hand. The instruction must contain that we need a Yes or No response.

Figure A.7: EdiPrompt Semantically Similar Instructions Meta-Prompt. Orange indicates meta-instruction, blue is where we insert the initial instruction.

Below are two instructions with their scores, the score is the f-1 score. We want to maximize this score.

Instruction 1: [...*insert first instruction (from selection phase)*...]
Score: [...*insert first instruction score*...]

Instruction 2: [...*insert second instruction (overall best)*...]
Score: [...*insert second instruction score*...]

Your task is to cross-over the above two instructions, combining the most effective elements from each to create a new, improved instruction. Also provide an explanation into the crossover and what elements you have taken from each instruction.

Figure A.8: EdiPrompt Crossover Step Meta-Prompt. Orange is meta-instruction, blue is the two prompts being combined and their respective scores.

Your task is to mutate the current crossover instruction to create a new instruction <INS>. Below are some previous instructions with their scores, the score is the f-1 score. We want to maximize this score.

Instruction: [...*insert instruction*...]
Score: [...insert instruction score...]

(..more instruction-score pairs...)

Below are examples of the problems, we are classifying customers conversations as vulnerable or not vulnerable. Our definition of vulnerability refers to customers who, due to their personal circumstances, are especially susceptible to harm. All customers are at risk of becoming vulnerable and this risk is increased by characteristics of vulnerability related to 4 key drivers:
Health - health conditions or illnesses that affect ability to carry out day-to-day tasks.
Life events - life events such as bereavement, job loss or relationship breakdown.
Resilience - low ability to withstand financial or emotional shocks.
Capability - low knowledge of financial matters or low confidence in managing money (financial capability). Low capability in other relevant areas such as literacy, or digital skills

Conversation: [...insert example conversation...]
<INS>
Answer: Yes/No

(...more problem examples...)

Crossover Instruction: [...*insert crossover instruction*...]

Your task is to mutate the above crossover instruction, to generate a new improved instruction that is different from all the instructions above, and has a higher score than all the instructions above. The instruction should be concise, effective, and generally applicable to the classification task at hand. The instruction must contain that we need a Yes or No response.

Figure A.9: EdiPrompt Mutate Step Meta-Prompt.

Appendix B

Full Prompt Optimisation Prompts

This contains all of the EdiPrompt prompts that were generated across the 8 iterations. We only analysed the mutated prompt as this is the prompt generated after a full run of the framework. However, we detail all prompts here such as initial, semantically similar, crossover and mutate. We have columns for the current prompt (which is the best from selection phase) and the best prompt (which is best from all previous iterations). The crossover prompt also supplies the explanation that we got from GPT-4 on its crossover process.

We also provide all of the prompts generated by OPRO across its eight iterations in Figure B.9.

P_inital / Score	Sem1/ Score	Sem2/ Score	P_current/ Score	P_best / Score	P_crossover/ Explanation	P_mutate/ Score
Classify the customer conversation as 'Yes' if it indicates vulnerability due to health conditions, life events, low resilience, or low capability affecting their situation. Otherwise, classify it as 'No'. We need a clear 'Yes' or 'No' response based on these criteria. **F1 Score: 0.490**	Assign a 'Yes' classification to any customer interaction that demonstrates the customer is experiencing vulnerability stemming from health issues, significant life changes, lacking resilience, or limited ability impacting their circumstances. For all other interactions, classify as 'No'. A definitive 'Yes' or 'No' answer is required based on these conditions. **F1 Score: 0.578**	Review each customer conversation and designate it as 'Yes' if there's evidence of vulnerability due to medical conditions, critical life events, insufficient resilience, or reduced capability influencing their condition. If these elements are not present, categorise the conversation as 'No'. Ensure to provide a clear 'Yes' or 'No' classification based on these specified criteria." **F1 Score: 0.643**	Review each customer conversation and designate it as 'Yes' if there's evidence of vulnerability due to medical conditions, critical life events, insufficient resilience, or reduced capability influencing their condition. If these elements are not present, categorise the conversation as 'No'. Ensure to provide a clear 'Yes' or 'No' classification based on these specified criteria." **F1 Score: 0.643**	Based solely on the evidence of vulnerability present in the conversation excerpt, indicate whether the customer is vulnerable or not by answering with a simple "Yes" or "No". Your determination should reflect the presence of any of the four key drivers of vulnerability (Health, Life events, Resilience, Capability) according to our vulnerability framework. Remember, your response must be either "Yes" or "No". **F1 Score: 0.610**	"Evaluate each interaction with the customer carefully, and determine if the customer exhibits signs of vulnerability. Indicate your assessment with a definitive 'Yes' if there is clear evidence of any of the following specific factors: serious medical conditions; significant life events; low resilience; or decreased capability that influences their current situation. Otherwise, respond 'No'. Your binary classification (Yes/No) should be grounded in the presence of these four critical elements of vulnerability, as outlined in our established vulnerability framework. Make sure your categorization is consistent and is solely based on the tangible evidence of vulnerability documented in the customer conversation." **Explanation of Cross-over Elements:** Both sets of instructions are synthesized to emphasize a careful review of each customer conversation. The need to provide a binary 'Yes' or 'No' answer is reiterated from both sets of instructions. The four key drivers of vulnerability (Health, Life events, Resilience, Capability) which form the basis of the decision are highlighted and specifically named to ensure clarity in what reviewers are looking for.. The importance of evidence and clear justification for the decision, pulled from the emphasis in Instruction 1, is reinforced, while the structure and phrasing are adjusted for better readability. The framework mentioned in Instruction 2 is specifically called the "vulnerability framework" for clarity and connection to organizational standards. Abstract terms from Instruction 1 such as "evidence of vulnerability" and "critical life events" are concretized, and the notion of "insufficient resilience" and "reduced capability" from Instruction 1 are maintained for continuity and specificity. By using an imperative tone ("Indicate," "Ensure," "Make sure"), the new instruction makes clear what the reviewers\' tasks and responsibilities are, drawing from the directive language used in both initial sets of instructions.This crossover approach combines the best elements of specificity, clarity, directive language, and focus on evidence, aiming to enhance understanding and potentially improve the f-1 score.	Assess the dialogue for indicators of the customer\'s vulnerability according to our outlined categories: Health issues impacting daily tasks, significant Life events, low Resilience, and limited Capability in managing finances or similar important matters. Upon identifying evidence that aligns with any of these vulnerability drivers, answer with 'Yes'. If such evidence is absent, reply with 'No'. Ensure your response is a clear 'Yes' or 'No'', based explicitly on the conversational evidence of these identified vulnerability factors. **F1 Score: 0.658**

Figure B.1: EdiPrompt First Iteration.

P_inital / Score	Sem1/ Score	Sem2/ Score	P_current/ Score	P_best / Score	P_crossover/ Explanation	P_mutate/ Score
After reviewing the dialogue, issue a "Yes" if you pinpoint any signs aligning with our vulnerability indicators: Health, significant Life events, Resilience, or Capability challenges. If these elements are not present, provide a "No". A precise "Yes" or "No" is required as a response.' **F1 Score: 0.537**	Examine the conversation and respond with 'Yes' if you detect any evidence of concerns related to Health, major Life events, Resilience, or Capability difficulties. Otherwise, reply with 'No'. Your answer should be a clear 'Yes' or 'No'. **F1 Score: 0.550**	Analyze the dialogue for any indications of vulnerability in the areas of Health, significant Life milestones, Resilience, or Capability impairments. Confirm with a 'Yes' if any are found, and with a 'No' if absent. Please provide only a 'Yes' or 'No' as your answer.' **F-1 Score: 0.531**	Examine the conversation and respond with 'Yes' if you detect any evidence of concerns related to Health, major Life events, Resilience, or Capability difficulties. Otherwise, reply with 'No'. Your answer should be a clear 'Yes' or 'No'. **F1 Score: 0.550**	Assess the dialogue for indicators of the customer\'s vulnerability according to our outlined categories: Health issues impacting daily tasks, significant Life events, low Resilience, and limited Capability in managing finances or similar important matters. Upon identifying evidence that aligns with any of these vulnerability drivers, answer with 'Yes'. If such evidence is absent, reply with 'No'. Ensure your response is a clear 'Yes' or 'No", based explicitly on the conversational evidence of these identified vulnerability factors. **F1 Score:0.658**	"Evaluate the dialogue for signs indicating the individual's susceptibility as per our specified sectors: Health complications influencing daily activities, critical Life transitions, diminished Resilience, and restricted Capability in dealing with financial or analogous crucial concerns. Should you discern any indications that correspond with these defined vulnerability elements, respond with 'Yes'. In the absence of such indications, your response should be 'No'. It is essential that your answer is an unequivocal 'Yes' or 'No', predicated solely on the tangible conversational evidence of these defined vulnerability dimensions." **Explanation of Crossover**: From Instruction 1 - The directness of the response ("Your answer should be a clear \'Yes\' or \'No\'."). The enumeration of the four specific areas of concern regarding vulnerabilities (Health, Life events, Resilience, Capability difficulties). From Instruction 2:- The explicit mention of what constitutes evidence ("based explicitly on the conversational evidence"). The detailed description of the areas of concerns which were somewhat broader, allowing for a more comprehensive assessment (Health issues impacting daily tasks, significant Life events, low Resilience, and limited Capability). The emphasis on evidence aligning with the vulnerability factors (ensuring the response is tied to concrete evidence from the conversation). The new instruction combines consistency and clarity from both sets of instructions while ensuring that what is to be evaluated is well-defined and evidence-based. The directness of the final response is maintained from Instruction 1 while drawing on the more detailed descriptions and requirement for explicit evidence from Instruction 2. This should ideally provide a balanced approach that maximizes the ability to identify vulnerabilities accurately, thereby aiming to improve upon the f-1 score previously achieved.	Review the conversation carefully for any signs that the individual falls into our vulnerability categories: significant Health issues impacting their everyday tasks, major Life events, low Resilience, or poor Capability in managing finances and other critical activities. If you detect factors that meet these conditions of vulnerability, answer plainly with 'Yes'. If no such factors are present, answer directly with 'No'. Provide only a 'Yes' or 'No' response, grounded strictly in the observable evidence of vulnerability from the dialogue. **F1 Score: 0.6625**

Figure B.2: EdiPrompt Second Iteration.

P_inital / Score	Sem1/ Score	Sem2/ Score	P_current/ Score	P_best / Score	P_crossover/ Explanation	P_mutate/ Score
‘ Examine the conversation and identify any explicit instances associated with the customer's potential vulnerability, focusing particularly on the four key drivers: significant Health challenges, notable Life events, diminished Resilience, or compromised Capability in critical areas such as financial management. Your response should be confined to a definitive 'Yes' for presence of vulnerability indicators, or 'No' for their absence, aligning strictly with the criteria based on the conversation content. Your answer must consist solely of 'Yes' or 'No.' **F1 Score: 0.652**	Review the dialogue to pinpoint and confirm if the customer shows signs of vulnerability due to significant health issues, major life events, reduced resilience, or limited ability to manage essential tasks like finances. Respond only with 'Yes' if such indicators are present, or 'No' if they are not, based solely on the conversation details, requiring a clear 'Yes' or 'No' answer. **F1 Score: 0.684**	Scrutinize the exchange to detect any clear evidence of the customer being at risk due to serious health conditions, impactful life changes, weakened resilience, or impaired ability in crucial domains like handling finances. Respond exclusively with 'Yes' if indicators of vulnerability are identified, or 'No' if they are absent, according strictly to the exchange, and limit your answer strictly to 'Yes' or 'No.' **F1 Score: 0.705**	Scrutinize the exchange to detect any clear evidence of the customer being at risk due to serious health conditions, impactful life changes, weakened resilience, or impaired ability in crucial domains like handling finances. Respond exclusively with 'Yes' if indicators of vulnerability are identified, or 'No' if they are absent, according strictly to the exchange, and limit your answer strictly to 'Yes' or 'No.' **F1 Score: 0.705**	Review the conversation carefully for any signs that the individual falls into our vulnerability categories: significant Health issues impacting their everyday tasks, major Life events, low Resilience, or poor Capability in managing finances and other critical activities. If you detect factors that meet these conditions of vulnerability, answer plainly with 'Yes'. If no such factors are present, answer directly with 'No'. Provide only a 'Yes' or 'No' response, grounded strictly in the observable evidence of vulnerability from the dialogue. **F1 Score: 0.6625**	"Examine the conversation meticulously to identify any definitive signs that the customer may be vulnerable because of severe health issues that could affect their daily life, significant life events, diminished resilience, or limited capacity in essential areas such as managing finances. Your answer should be a concise \'Yes\' only if you discern clear evidence of such vulnerability factors according to the exchange, or 'No' if these elements are not present. Limit your response to solely ‘Yes’ or 'No', based strictly on the concrete information provided within the interaction." **Explanation of Crossover:**From Instruction 1:- The term "scrutinize" was changed to "examine meticulously" for a similar meaning but using a combination of words from both instructions for clarity The concept of "evidence of the customer being at risk" was maintained but rephrased for clarity and brevity- The phrase "impacted life changes" from Instruction 1 was replaced with "significant life events" from Instruction 2 since "significant" is generally more precise than "impactful".The mention of "weakened resilience" was directly taken from Instruction 1, as it is understandable.From Instruction 2: The structure of "detect any signs that the individual falls into our vulnerability categories" was simplified and made direct to improve readability. The term "Health issues" from Instruction 2 was modified to "health issues" for grammatical correctness. "poor Capability" was changed to "limited capacity" which is a combination of the terms used in both instructions for more straightforward language. The phrase "Provide only a \'Yes\' or \'No\' response, grounded strictly in the observable evidence" was taken from Instruction 2 but was fine-tuned to fit the new combined instruction.From both instructions: The clarity and directness of the yes/no response requirement were maintained from both sets, ensuring that the instruction is adhered to without ambiguity. The overall goal of basing responses on concrete signs of vulnerability from the conversation was emphasized, a critical point present in both original instructions.Terms such as "health conditions," "life changes," "resilience," and "handling finances" were taken from both instructions and synthesized for maximum clarity and effectiveness.	Carefully analyze the customer\'s conversation to pinpoint any unequivocal indicators of vulnerability grounded in our established criteria: substantial health complications impeding daily routine, major life-changing events, weakened resilience, or insufficient capability in vital undertakings like finance management. Respond with a clear-cut 'Yes' if evidence of such vulnerability factors is evident within the dialogue, or a 'No' if these conditions are not met. Your reply must be strictly ‘Yes' or 'No', based solely on the tangible evidence within the conversational data presented. **F1 Score: 0.696**

Figure B.3: EdiPrompt Third Iteration.

P_inital / Score	Sem1/ Score	Sem2/ Score	P_current/ Score	P_best / Score	P_crossover/ Explanation	P_mutate/ Score
Evaluate the conversation for any explicit indications that classify the customer as vulnerable, with vulnerability defined by the presence of substantial health impairments, adverse life events, financial or emotional frailty, or a deficiency in handling essential tasks like money management. If such signs exist according to our parameters, answer solely with 'Yes'. If such signs are not evident, provide only the singular reply 'No'. Remember, your response should be confined to either 'Yes' or 'No'. **F1 Score: 0.692**	Review the dialogue to detect if the individual is exhibiting signs of vulnerability, such as significant health issues, recent traumatic experiences, limited financial/emotional capacity, or difficulties in basic personal financial management. Respond with 'Yes' if such signs align with our criteria; otherwise, respond with 'No'. Ensure your answer consists of only 'Yes' or 'No'. **F1 Score: 0.615**	Scan the conversation to ascertain whether the customer shows any signs of being vulnerable, defined by severe medical conditions, recent distressing events, weakness in financial/emotional stability, or trouble with fundamental activities such as managing finances. Only answer 'Yes' if these indicators are present; if they are not, respond with 'No'. Your response must be limited to 'Yes' or 'No'." **F1 Score: 0.634**	Evaluate the conversation for any explicit indications that classify the customer as vulnerable, with vulnerability defined by the presence of substantial health impairments, adverse life events, financial or emotional frailty, or a deficiency in handling essential tasks like money management. If such signs exist according to our parameters, answer solely with 'Yes'. If such signs are not evident, provide only the singular reply 'No'. Remember, your response should be confined to either 'Yes' or 'No'. **F1 Score: 0.692**	Scrutinize the exchange to detect any clear evidence of the customer being at risk due to serious health conditions, impactful life changes, weakened resilience, or impaired ability in crucial domains like handling finances. Respond exclusively with 'Yes' if indicators of vulnerability are identified, or 'No' if they are absent, according strictly to the exchange, and limit your answer strictly to 'Yes' or 'No.' **F1 Score: 0.705**	"Examine the dialogue attentively for unequivocal signs that the customer may be categorized as vulnerable. This includes a detailed look for any severe health impairments, significant life disruptions, diminished financial or emotional strength, or challenges in managing key activities, particularly those related to financial stewardship. Should any indicators of such vulnerability be observed as per our guidelines, please respond exclusively with 'Yes'. Conversely, if these indicators are absent, your response must be a definitive 'No'. Stay strictly within these parameters, offering only one-word answers: 'Yes' or 'No'." **Explanation of Crossover:**What has been taken from Instruction 1: The phrase "unequivocal signs" which suggests clarity in identification of vulnerability.The explicit mention of "emotional frailty," which encompasses a broader range of vulnerabilities.The mention of "deficiency in handling essential tasks like money management" which is a specific example of the challenges faced by vulnerable customers.The clear instruction to respond with \'Yes\' or \'No\' only What has been taken from Instruction 2: The term "scrutinize" to convey the intensity and thoroughness of the evaluation required. "Substantial health impairments" was changed to "severe health impairments" for consistency with Instruction 2\'s phrasing. "Adverse life events" was changed to "significant life disruptions" to maintain Instruction 2\'s phrasing, which may imply a broader set of circumstances. The term "weakened resilience" to recognize the broader scope of personal adversity. "Crucial domains like handling finances" has been incorporated to underscore the functional aspect of vulnerability. Combined elements: Both instructions\' emphasis on a binary \'Yes\' or \'No\' response has been maintained. Clarification that the scope is strictly limited to the exchange under scrutiny without any external assumptions or inferences. A streamlined combination of both sets of vocabularies, to ensure comprehensive evaluation while maintaining clarity and simplicity in the instruction for optimal understanding and adherence.'	"Carefully evaluate the conversation for solid indicators that may classify the customer as vulnerable, following our guidelines. Look for substantial health impairments, notable life upheavals, reduced resilience or financial/emotional strength, or difficulties in handling vital responsibilities, especially concerning financial management. If you observe any such signs of vulnerability, reply with a 'Yes'. If these signs are not present, your response should be a 'No'. Your answer must be concise: only 'Yes' or 'No'.'" **F1 Score: 0.657**

Figure B.4: EdiPrompt Fourth Iteration.

P_inital / Score	Sem1/ Score	Sem2/ Score	P_current/ Score	P_best / Score	P_crossover/ Explanation	P_mutate/ Score
Examine the interaction meticulously for any manifestations of the client's susceptibilities as outlined in our framework: critical health issues, significant life events, resistance to adversity, or capability in financial and similar essential matters. Respond concisely with 'Yes' if any signs of these vulnerabilities are revealed within the interaction, or with 'No' if none are observed. Your answer should be strictly either 'Yes' or 'No', derived from the explicit evidence of vulnerability indicators within the dialogue" **F1 Score: 0.646**	Carefully scrutinize the exchange to identify any indications of the customer''s weaknesses as per our established criteria: serious health conditions, pivotal life changes, resilience to challenges, or adequacy in managing finances and other vital domains. Provide a succinct answer of 'Yes' if the conversation shows any such vulnerabilities, or 'No' if none are present, based solely on the clear evidence of vulnerability indicators in the interaction. **F1 Score: 0.623**	"Thoroughly review the dialogue for signs of vulnerability in the client as per our guidelines: severe health problems, major personal events, fortitude in the face of hardship, or proficiency in economic and critical issues. Reply tersely with 'Yes' if signs of these vulnerabilities are evident in the conversation, or with 'No' if they are absent, based strictly on observable evidence of these vulnerabilities within the dialogue." **F1 Score: 0.697**	"Thoroughly review the dialogue for signs of vulnerability in the client as per our guidelines: severe health problems, major personal events, fortitude in the face of hardship, or proficiency in economic and critical issues. Reply tersely with 'Yes' if signs of these vulnerabilities are evident in the conversation, or with 'No' if they are absent, based strictly on observable evidence of these vulnerabilities within the dialogue." **F1 Score: 0.697**	Scrutinize the exchange to detect any clear evidence of the customer being at risk due to serious health conditions, impactful life changes, weakened resilience, or impaired ability in crucial domains like handling finances. Respond exclusively with 'Yes' if indicators of vulnerability are identified, or 'No' if they are absent, according strictly to the exchange, and limit your answer strictly to 'Yes' or 'No.' **F1 Score: 0.705**	Examine the conversation carefully for undeniable indications of the client''s potential vulnerabilities according to our standards: serious health concerns, significant life events, resilience during adversity, or capability in handling financial and critical matters. Respond concisely with 'Yes' only if such signs of vulnerability are present, or with 'No' if they are not, based solely on the concrete evidence of these vulnerabilities in the dialogue. **Explanation of Crossover**: 1. From Instruction 1, I kept the phrase "Thoroughly review the dialogue for signs of vulnerability in the client as per our guidelines," but modified it to "Examine the conversation carefully for undeniable indications of the client\'s potential vulnerabilities according to our standards." This maintains the directive to thoroughly review but emphasizes the need for undeniable indicators, aligning with the precise nature of Instruction 2. 2. The specific vulnerabilities to look for are kept consistent with both instructions: "serious health problems, major personal events, fortitude in the face of hardship, or proficiency in economic and critical issues" from Instruction 1, and "serious health conditions, impactful life changes, weakened resilience, or impaired ability in crucial domains like handling finances" from Instruction 2. I created a synthesis by using "serious health concerns, significant life events, resilience during adversity, or capability in handling financial and critical matters," which captures the essence of both sets of terms without repeating them verbatim. 3. The term "Reply tersely" from Instruction 1 is retained, updated to "Respond concisely" to align with the succinctness required from Instruction 2 without using the word "exclusively," which could be seen as slightly more restrictive than necessary. 4. Instruction 2's focus on clear evidence and limiting the response to strictly \'Yes\' or \'No\' is maintained by instructing to respond only if such signs are present or not, based solely on the concrete evidence in the dialogue.By blending the thoroughness and structure of Instruction 1 with the emphasis on clarity and conciseness from Instruction 2, the new instruction aims to maximize the effectiveness of the evaluation process with a clear and directive approach. The crossover attempts to combine the strengths of both initial instructions to improve the f-1 score by being both thorough and precise.	Analyze the dialogue meticulously for undeniable signs that the customer may be vulnerable, in line with our criteria: significant health issues that impede daily routines, profound life events, resilience in the face of hardships, or capability in financial management and other pivotal areas. Provide a succinct 'Yes' if such signs are evident, or a 'No' otherwise, drawing strictly from the concrete evidence in the conversation. Your answer must solely consist of 'Yes' or 'No'. **F1 Score: 0.714**

Figure B.5: EdiPrompt Fifth Iteration.

P_inital / Score	Sem1/ Score	Sem2/ Score	P_current/ Score	P_best / Score	P_crossover/ Explanation	P_mutate/ Score
Examine the dialogue to determine the customer's level of vulnerability in accordance with our benchmarks: any substantial health concerns that may disrupt daily activities, noteworthy life occurrences such as loss of employment or loss of a loved one, diminished resilience to cope with life's adversities, or a reduced capacity to manage financial and other key matters. If your assessment reveals elements corresponding to these areas of vulnerability, state 'Yes'; if not, state 'No'. Your judgment should be solely based on the evidence from the conversation and adhere to a binary 'Yes' or 'No' format. **F1 Score: 0.614**	Analyze the provided conversation to determine if the customer displays any signs of vulnerability, such as significant health issues that affect daily life, recent major life changes like job loss or bereavement, low ability to face hardships, or struggles with handling personal finances or other essential responsibilities. Your evaluation should be based on these criteria; answer 'Yes' for detected vulnerability or 'No' if none are found, sticking to this binary response format. **F1 Score: 0.512**	Review the dialogue carefully to identify if the customer exhibits vulnerability by presenting serious health problems impacting regular activities, undergoing significant life events like unemployment or the death of a loved one, showing weak ability to endure life challenges, or having trouble managing financial or central life tasks. Give a 'Yes' response if these vulnerabilities are evident or 'No' if not, keeping to the binary answer requirement **F1 Score: 0.639**	Review the dialogue carefully to identify if the customer exhibits vulnerability by presenting serious health problems impacting regular activities, undergoing significant life events like unemployment or the death of a loved one, showing weak ability to endure life challenges, or having trouble managing financial or central life tasks. Give a 'Yes' response if these vulnerabilities are evident or 'No' if not, keeping to the binary answer requirement **F1 Score: 0.639**	Analyze the dialogue meticulously for undeniable signs that the customer may be vulnerable, in line with our criteria: significant health issues that impede daily routines, profound life events, resilience in the face of hardships, or capability in financial management and other pivotal areas. Provide a succinct 'Yes' if such signs are evident, or a 'No' otherwise, drawing strictly from the concrete evidence in the conversation. Your answer must solely consist of 'Yes' or 'No'. **F1 Score: 0.714**	Examine the dialogue closely to detect clear indications of customer vulnerability, guided by our specified criteria: substantial health issues disrupting daily functions, experiencing critical life transitions such as unemployment or bereavement, demonstrating a diminished capacity to cope with adversity, or difficulties in handling financial matters or vital life operations. Respond with a definitive 'Yes' if such vulnerabilities are present based on the evidence in the dialogue, or a 'No' if they are absent. Your response must be binary, strictly limited to either 'Yes' or 'No'. Explanation of the Crossover:From Instruction 1:. A detailed criterion for identifying vulnerability is given, which includes serious health problems, significant life events, coping ability, and managing financial or central life tasks. This establishes a comprehensive framework for assessment. The clear binary answer requirement of \'Yes\' or \'No\' is emphasized, which simplifies the decision-making process and avoids ambiguity.From Instruction 2:. The instruction uses terms like "undeniable signs" and "concrete evidence," which imply a rigorous standard for determining vulnerability and underscore the need for a clear, evidence-based approach. The phrase "meticulously" from Instruction 2 is replaced with "closely" from Instruction 1 to maintain the attention to detail while avoiding redundancy. The use of words such as "succinct," "strictly," and "solely" in Instruction 2 underscores the necessity of a focused and unequivocal response, reinforcing the need to adhere strictly to the observed evidence.The new instruction combines these elements to create a directive that is detailed, evidence-based, and emphatic about the necessity for a binary \'Yes\' or \'No\' answer. It leverages the strengths of both previous instructions, hoping to provide clarity in how to identify customer vulnerability whilst ensuring the response is grounded in traceable facts from the dialogue, thus maximizing the likelihood of improving the F1 score.	'Examine the dialogue attentively for unequivocal evidence of customer vulnerability, as per our criteria: severe health conditions hampering everyday living, pivotal life changes (e.g., loss of a loved one or employment), diminished resilience to cope with setbacks, or challenges in managing finances or fundamental duties. Issue a definitive 'Yes' only when these vulnerability characteristics are clearly observed in the conversation, or a 'No' if they are not. Your response should be strictly binary: 'Yes' or 'No'. **F1 Score: 0.675**

Figure B.6: EdiPrompt Sixth Iteration.

P_inital / Score	Sem1/ Score	Sem2/ Score	P_current/ Score	P_best / Score	P_crossover/ Explanation	P_mutate/ Score
Determine the customer's vulnerability from the interaction by identifying any explicit references that match our vulnerability criteria: substantial daily life impairment due to health conditions; critical life events such as bereavement, unemployment, or relationship breakdown; diminished ability to cope with unforeseen financial or emotional challenges; or evident difficulties in managing fiscal responsibilities and essential tasks. If any of these criteria are distinctly mentioned, respond decisively with 'Yes'. Otherwise, provide a clear 'No'. Your answer must be strictly a singular 'Yes' or 'No', based on the observable evidence within the dialogue. Answer: <Yes/No> **F1 Score: 0.687**	Assess if the client exhibits any signs of vulnerability by noting explicit statements that indicate: significant disruption in day-to-day activities due to health issues; recent severe life events like losing a loved one, job loss, or breakups; an apparent lack of resilience facing unexpected financial or emotional hardship; or clear struggle with handling basic financial obligations or life necessities. Record your definitive analysis as 'Yes' if such indicators are present, or 'No' otherwise. The response should be a single 'Yes' or 'No' based solely on the details provided in the conversation. **F1 Score: 0.669**	Evaluate the customer's conversation for any clear signals of vulnerability, specifically mentioning: serious impact on everyday life function due to medical conditions; acute life changes including death in the family, being laid off, or separation; reduced capacity to deal with unanticipated fiscal or emotive stress; or discernible troubles with managing money or essential daily duties. Respond with a categorical 'Yes' if these vulnerabilities are identified, or 'No' if they are not. Ensure that your answer is either a solitary 'Yes' or 'No', founded on the explicit content of the interaction." **F1 Score: 0.653**	Determine the customer's vulnerability from the interaction by identifying any explicit references that match our vulnerability criteria: substantial daily life impairment due to health conditions; critical life events such as bereavement, unemployment, or relationship breakdown; diminished ability to cope with unforeseen financial or emotional challenges; or evident difficulties in managing fiscal responsibilities and essential tasks. If any of these criteria are distinctly mentioned, respond decisively with 'Yes'. Otherwise, provide a clear 'No'. Your answer must be strictly a singular 'Yes' or 'No', based on the observable evidence within the dialogue. Answer: <Yes/No> **F1 Score: 0.687**	Analyze the dialogue meticulously for undeniable signs that the customer may be vulnerable, in line with our criteria: significant health issues that impede daily routines, profound life events, resilience in the face of hardships, or capability in financial management and other pivotal areas. Provide a succinct 'Yes' if such signs are evident, or a 'No' otherwise, drawing strictly from the concrete evidence in the conversation. Your answer must solely consist of 'Yes' or 'No'. **F1 Score: 0.714**	Carefully evaluate the conversation for any definitive indications that the customer may be vulnerable by checking for explicit mentions that align with our criteria for vulnerability: significant daily life impairment due to health issues, severe life disruptions like bereavement, joblessness, or the end of a relationship; a reduced capacity to handle unexpected financial or emotional challenges; or apparent challenges in managing financial obligations and day-to-day tasks. If the customer explicitly meets any of these criteria based on the transparent evidence in the dialogue, respond with a 'Yes'. If these criteria are not clearly met, answer 'No'. Your response should be solely a single 'Yes' or 'No', relying squarely on the information presented in the interaction **Explanation of Crossover**:From Instruction 1, the new instruction maintains the specific examples of vulnerability criteria ("substantial daily life impairment due to health conditions; critical life events such as bereavement, unemployment, or relationship breakdown; diminished ability to cope with unforeseen financial or emotional challenges; or evident difficulties in managing fiscal responsibilities and essential tasks"), which are detailed and allow for a comprehensive analysis of the customer\'s situation. It also retains the decisive instruction to respond with a "Yes" or "No" based solely on observable evidence.From Instruction 2, the new instruction takes the language of meticulous analysis ("Analyze the dialogue meticulously for undeniable signs") which emphasizes the need for careful evaluation, as well as terms like "undeniable signs" and "evident" that suggest a high threshold for determining vulnerability. It continues with succinct response instructions, indicating the response must be "solely consist of \'Yes\' or \'No\'", reinforcing clarity in the final answer.\n\nThe crossover aims to blend the thorough and explicit criteria for vulnerability from Instruction 1 with the careful and definitive analysis emphasized in Instruction 2. The resulting instruction seeks to maximize the F1 score by providing a clear, criteria-based decision-making process for identifying the vulnerability while enforcing a strict and evidence-based response protocol.'	Scrutinize the customer's conversation, seeking explicit signs that meet our vulnerability criteria: substantial health afflictions disrupting daily life, traumatic life occurrences such as bereavement or job loss, decreased resilience to life's financial or emotional adversities, or evident struggles with managing essential finances and tasks. Where the dialogue contains transparent examples of these vulnerability factors, answer with a definitive 'Yes'. If the dialogue lacks such evidence, respond with a 'No'. Your judgment must be based solely on the concrete information provided in the interaction, and your answer should be limited to a simple 'Yes' or 'No'." **F1 Score: 0.668**

Figure B.7: EdiPrompt Seventh Iteration.

P_inital / Score	Sem1/ Score	Sem2/ Score	P_current/ Score	P_best / Score	P_crossover/ Explanation	P_mutate/ Score
Scrutinize the dialogue for indicators of vulnerability according to the following distinct criteria: health issues hindering day-to-day life, significant life disruptions such as loss, unemployment, or family difficulties, frailty in dealing with unexpected personal or financial crises, and limited understanding or self-assurance in money handling or other essential competencies. If the discourse unmistakably reflects these vulnerability factors, respond with a definitive 'Yes'. If these attributes are not present, issue a 'No'. Your response must be a categorical 'Yes' or 'No', based solely on the evidence in the exchange." **F1 Score: 0.694**	Examine the conversation to identify if there are clear signs of vulnerability, specifically health challenges affecting daily activities, major life disruptions like bereavement, job loss, or family issues, an inability to cope with unforeseen personal or financial events, and a poor grasp or confidence in managing finances or essential skills. Provide a simple 'Yes' if these vulnerabilities are evident, otherwise give a 'No'. **F1 Score: 0.667**	Review the dialogue to determine if there are explicit signs of vulnerability including but not limited to health problems impacting regular life, considerable life disturbances such as death, unemployment, or familial strife, weakness in facing sudden personal or economic emergencies, and a lack of knowledge or self-assurance in financial management or crucial life skills. Reply with 'Yes' if these signs are apparent, and 'No' if they are absent. Your answer must be solely 'Yes' or 'No' **F1 Score: 0.635**	Scrutinize the dialogue for indicators of vulnerability according to the following distinct criteria: health issues hindering day-to-day life, significant life disruptions such as loss, unemployment, or family difficulties, frailty in dealing with unexpected personal or financial crises, and limited understanding or self-assurance in money handling or other essential competencies. If the discourse unmistakably reflects these vulnerability factors, respond with a definitive 'Yes'. If these attributes are not present, issue a 'No'. Your response must be a categorical 'Yes' or 'No', based solely on the evidence in the exchange." **F1 Score: 0.6943942133815552**	Analyze the dialogue meticulously for undeniable signs that the customer may be vulnerable, in line with our criteria: significant health issues that impede daily routines, profound life events, resilience in the face of hardships, or capability in financial management and other pivotal areas. Provide a succinct 'Yes' if such signs are evident, or a 'No' otherwise, drawing strictly from the concrete evidence in the conversation. Your answer must solely consist of 'Yes' or 'No'. **F1 Score: 0.714**	Carefully examine the dialogue for unequivocal signs of vulnerability, employing the specified benchmarks: health problems that substantially interfere with daily activities, major life upheavals like bereavement, joblessness, or familial issues, resilience or fragility when faced with unexpected personal or financial troubles, and adequacy in managing finances or other crucial life skills. If you discern these vulnerability indicators clearly evidenced in the conversation, state a resolute 'Yes'. Otherwise, provide a 'No'. Render your verdict as either 'Yes' or 'No', based strictly on tangible proof found in the dialogue. **Explanation of crossover**: Language Precision: The use of "unequivocal signs" from Instruction 2 was chosen for its precision, reinforcing that the indicators must be clear and undeniable. Criteria Specification: The detailed criteria from Instruction 1, such as "health issues hindering day-to-day life" and "frailty in dealing with unexpected personal or financial crises," were maintained for their specificity. These details help in providing clear guidelines for what constitutes vulnerability. Comprehensive Criteria: The merged instruction includes both the mention of "major life upheavals" from Instruction 2 and the extended examples of what constitutes upheavals from Instruction 1, like "loss, unemployment, or family difficulties." This approach combines the succinctness of Instruction 2 with the comprehensive nature of Instruction 1. Clarity in Directions: Both instructions\' insistence on a categorical \"Yes\' or \'No\' answer was preserved, highlighting the need for a definitive and evidence-based conclusion. Structural Flow: The new instruction preserves the logical flow of key points to assess, appearing in a similar order as seen in both original instructions. This flow aids in clarity and the systematic assessment of vulnerability factors. Vocabulary: The term "frailty" in Instruction 1 was replaced with "resilience or fragility" from Instruction 2 to encapsulate the spectrum of responses to crises and add nuance. Evidence-Based Decision: The phrase "based strictly on tangible proof found in the dialogue" from Instruction 2 was used in the final part of the new instruction to emphasize the importance of relying on concrete evidence from the conversation.By blending the clarity and sharp focus of Instruction 2 with the detailed descriptors and comprehensive nature of Instruction 1, the new instruction aims to maximize understanding and effectiveness in identifying vulnerability in dialogues. The f-1 score is anticipated to be higher since the instruction is designed to combine the strengths of both source instructions.'	Rigorously scrutinize the dialogue for definitive indications of the customer's vulnerability using our established criteria: significant health complications affecting day-to-day functionality, severe life events like loss, unemployment, or domestic disturbances, lack of resilience to socioeconomic or emotional adversity, and proficiency in handling monetary affairs or essential life competencies. Confirm a clear 'Yes' if such vulnerability attributes are concretely presented in the discussion, or state 'No' in their absence. Your determination should be a clear-cut 'Yes' or 'No', anchored in the solid evidence of vulnerability disclosed through the interaction. **F1 Score: 0.696**

Figure B.8: EdiPrompt Eighth Iteration.

Iteration	PO Framework	Prompt	F-1 Score
1	OPRO	Evaluate the customer conversation and respond with 'Yes' if there's explicit mention of any health conditions, debilitating life events, evidence of financial or emotional strain indicating low resilience, or signals of limited understanding or confidence in managing their finances or everyday technology. Otherwise, respond with 'No'. A definitive 'Yes' or 'No' is required based on these stated factors.	**0.668**
2	OPRO	Assess the customer conversation and determine if it displays any indicators of vulnerability, such as explicit references to health conditions impacting daily tasks, adverse life events causing significant distress, evidence of low financial or emotional resilience, or signs of difficulty with financial or technological literacy. Issue a 'Yes' if any of these vulnerabilities are present; otherwise, issue a 'No'. A decisive 'Yes' or 'No' answer is required.	**0.562**
3	OPRO	Review the dialogue to identify any explicit mentions of health conditions impacting daily functioning, recent significant life changes such as bereavement or unemployment, observable evidence of substantial emotional or financial hardship, or difficulty in understanding and handling financial or digital tasks. If any of these vulnerability indicators are present, then respond with 'Yes'; if none are observed, then respond with 'No'. A definitive binary 'Yes' or 'No' conclusion is required for classification.	**0.658**
4	OPRO	Analyze the customer conversation, and allocate a 'Yes' if it directly references any underlying medical conditions, traumatic life changes, or explicit signs of financial or emotional struggle affecting daily activities, or if there are clear indications of the customer's limited ability or confidence in managing financial tasks, technology, or essential literacy. If no such indications are present, assign a 'No'. Provide only a concise 'Yes' or 'No' determination based on these guidelines. Answer: <Yes/No>"	**0.646**
5	OPRO	Review the customer's dialogue for definitive references to health complications impacting routine activities, marked life events like loss of employment or personal upheaval, tangible signs of severe emotional or financial distress, or explicit difficulties with fundamental financial, literacy, or digital skills. Confirm vulnerability with a 'Yes' if manifest; if absent, provide a 'No'. Emit an unambiguous 'Yes' or 'No' verdict accordingly.	**0.703**
6	OPRO	Examine the client conversation and determine vulnerability by identifying any explicit statements concerned with health conditions impacting daily life, significant life events such as bereavement or unemployment, signs of emotional or economic fragility, or pronounced difficulties in financial or digital comprehension and management. If any of these elements are present, respond 'Yes'; if absent, respond 'No'. A singular 'Yes' or 'No' answer is required, reflecting these criteria.	**0.586**
7	OPRO	Review conversations to ascertain customer vulnerability by identifying explicit indications of substantial health issues affecting daily functionality, severe life disruptions such as the loss of a loved one or employment, any tangible signs of emotional or economic vulnerability, or distinct shortcomings in handling financial responsibilities or navigating digital platforms. If any such indicators are unmistakably present, reply with 'Yes'; otherwise, issue a 'No'. A clear-cut 'Yes' or 'No' answer is required solely based on these established criteria. Answer: <Yes/No>"	**0.611**
8	OPRO	Examine the customer dialogue and categorize it as 'Yes' if there are definitive indicators of health issues impacting routine tasks, significant life events that could substantially affect the customer's well-being, or clear evidence of low resilience or capability in dealing with financial, emotional, technological, or literacy challenges. If these stated indicators of vulnerability are not present, classify it as 'No'. A categorical 'Yes' or 'No' is essential, reflecting these specific vulnerability parameters.Answer: <Yes/No>	**0.699**

Figure B.9: All OPRO prompts.

Appendix C

Vulnerability Framework

Health	Life events	Resilience	Capability
Physical disability	Retirement	Inadequate (outgoings exceed income) or erratic income	Low knowledge or confidence in managing finances
Severe or long-term illness	Bereavement	Over- indebtedness	Poor literacy or numeracy skills
Hearing or visual impairment	Income Shock	Low savings	Poor English language skills
Mental health condition or disability	Relationship Breakdown	Low emotional resilience	Poor or non-existent digital skills
Addiction	Domestic abuse (including economic control)		Learning difficulties
Low mental capacity or cognitive disability	Caring responsibilities		No or low access to help or support
	Other circumstances that affect people's experience of financial services eg, leaving care, migration or seeking asylum, human trafficking or modern slavery, convictions		

Figure C.1: FCA Vulnerability Framework.