LLM self-critique for task-oriented text generation

Vilde Schulerud Bøe

Supervisor: Dr. Miri Zilka

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

King's College August 2023

Declaration

I, Vilde Schulerud Bøe of King's College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

The software used for this thesis was written in Python and its standard libraries like Numpy and Pytorch, but I have also used libraries like bitsandbytes, transformer and sentencepiece. All software is available for inspection via GitHub: https://github.com/vildesboe/MLMI_thesis/tree/main. I have two scripts for fine-tuning which were only slightly modified from fine-tuning scripts in the Vicuna and Alpaca-LoRA GitHub repositories (Wang., 2023; Zheng et al., 2023).

The project has used about 220 GPU hours.

Wordcount (including the appendix): 14959

Vilde Schulerud Bøe
August 2023

Acknowledgements

First and foremost, I would like to thank my supervisor Dr. Miri Zilka for devising such an interesting project, and for her advice and support throughout working with this project. I am grateful for the guidance I received from Dr. Miri Zilka and Jiri Hron, and their eagerness to help me solve challenges and form new ideas for this work.

I would also like to thank Aker Scholarship for giving me the opportunity to study at the University of Cambridge, working alongside so many brilliant people.

Lastly, I would like to thank my family and partner for their endless support.

Abstract

Large Language Models (LLMs) are capable of generating well-written text and have the potential to be very useful in real-world applications. For example, if we can trust them to reliably produce output of high quality, they might be useful in legal matters. In this work, I look at one such use case, and evaluate whether LLMs are able to reliably produce high-quality output for a specific task that is more complex than what has typically been considered prior to this work. I introduce a number of methodologies, and consider several LLMs for solving the task. In particular, I explore whether a smaller, fine-tuned LLM can compete with a larger pre-trained LLM. I also introduce a new dataset for the task, as well as my own method for evaluation. The main conclusion of the dissertation is that the larger LLM I considered, produced outputs of better quality than any smaller model I tested. However, none of the LLMs I considered were able to produce reliable outputs of high quality.

Table of contents

List of figures

List of tables

Nomenclature

Roman Symbols

`CR`	Critique Request
`DR`	Direct Refinement
`ER`	Explanation Request
`RR`	Revision Request
COT	Chain-Of-Thought
GPT	Generative Pre-trained Transformer
LLM	Large Language Model
LM	Language Model
LoRA	Low-Rank Adaption
NLP	Natural Language Processing
QLoRA	Quantized Low-Rank Adaption
TOD	Task-Oriented Dialogue

Chapter 1

Introduction

1.1 Motivation

Language is an important part of human life, as it is our main way to communicate with each other. We use language to express our thoughts, ideas, and even emotions. Language is also a crucial component in important life moments, including job interviews, conflict resolution, and possibly courtroom presentations. A person who struggles with expressing themselves well, in writing or orally, might experience bias against them in a courtroom. One example of a situation where this is a challenge is if a person pleads guilty to an offense, and is asked to make a plea in mitigation. Such a plea should contain information about the crime and the defendant, and if performed correctly, it can result in a milder sentence. Many people choose to perform such a plea by themselves instead of seeking counsel. This could be because of financial reasons, or because they wish to speak directly to the judge (GOV.UK, b). Because of this, people who don't have the means to pay a defense attorney to help them write a plea in mitigation, or who are not as skilled in expressing themselves in the correct manner, can end up receiving a harder sentence than someone with a more fortunate background in the same situation. This work aims to address this issue by looking at the possibility of making good representation in court accessible to everyone. As a tool to achieve this, we look to Large Language Models (LLMs).

LLMs are models that use AI to perform different natural language tasks such as interpreting and generating text. Using deep learning, LLMs can perform different types of text analysis, summarisation, translation, and text generation. They are able to generate well-written text which is difficult to distinguish from human-written text, and they might therefore be suitable for the task of helping people write courtroom statements (Brown et al., 2020).

1.2 Primary goals

The overall purpose of this project is not to build a usable product, but rather to evaluate whether or not LLMs can be helpful in generating good pleas in mitigation, in a reliable manner. There are two main goals for this work:

1. Can we reliably produce useful statements that can be read in court as pleas in mitigation? To explore this, I will use the best LLM available.

2. I want to explore and compare different setups for this task, including different configurations and both larger, pre-trained models and smaller, fine-tuned models.

1.3 Contributions

This work has made several contributions.

First, I have looked at whether a state-of-the-art LLM can perform reliably on the complex task of creating a plea in mitigation. My conclusion on this is that it cannot yet perform reliably on this task by using the methods I have tested.

Second, I have made a new dataset containing in full, 281 statements for pleas in mitigation, with different types of annotations that state whether each statement satisfies one out of the multiple criteria I want all statements to satisfy.

I have evaluated a total of 16 different setups on the task of creating a plea in mitigation. The setups use different methods as well as different LLMs.

I have observed that the fine-tuned smaller models could not compete in performance compared to the larger pre-trained state-of-the-art model.

1.4 Dissertation overview

The remainder of the dissertation will be structured in the following way:

Chapter 2 will present an overview of related literature and required background information. It will begin by providing further information on how to write a good plea in mitigation and discuss trends in relevant Task-Oriented Dialogue literature. I will then present the main methods that inspired the framework I used, as well as the LLMs I have used in these methods. Lastly, I will briefly discuss methods for fine-tuning LLMs.

Chapter 3 presents the methodology we have created for our new task. It will further discuss how I have used information and methods from the background to create a system suitable for my particular task.

Chapter 4 contains a description of the preparation of my experiments. I will introduce my new dataset, discuss how prompt engineering was done, and talk about how I have evaluated my results.

Chapter 5 presents the results of this dissertation. It contains information about how I have fine-tuned the smaller LLM, and results of the performance of all the configurations I have looked at.

Chapter 6 further discusses the results from Chapter 5 from a taller viewpoint, and suggests ideas for further work. This chapter also contains a conclusion of the full work.

Chapter 2

Background

This chapter will introduce necessary background material for the project. It will not be a complete guide on all related previous work, but rather a summary of trends, presenting knowledge necessary to understand this thesis. I begin by discussing the task at hand: writing a plea in mitigation, before looking at some technical details and challenges in the Task-Oriented Dialogue (TOD) literature. I will continue by presenting the main inspiration for my project: the methods of self-critique and Constitutional AI (Bai et al., 2022; Saunders et al., 2022). I will then discuss the models used to perform self-critique (Section 2.4), and how I will fine-tune some of them: using Quantized Low-Rank Adaption (QLoRA) in Section 2.5.

2.1 A plea in mitigation

A plea in mitigation is held in a UK court to explain personal circumstances and plead for leniency in sentencing.

When a person has committed a crime in the UK, and decided to plead guilty, they are usually called into a courtroom to be sentenced. To limit the scope of this project, I have only focused on smaller, less serious offenses, which are typically judged in a UK Magistrates' court. What typically happens next, is that a defense attorney will read out a statement to plead for leniency in the sentencing of the defendant. But often, people choose to self-represent instead of using a lawyer. In this case, the defendant does not need to argue about technical criminal details, but there is still some specific information that should be included in their plea.

Mitigating factors are aspects of the offense, or personal circumstances, that indicate a less serious crime, and can therefore result in a lesser sentence if properly explained to the court (Judiciary of England and Wales, 2020). Great provocation, mental illness or disability, age, and playing a minor role in the offense, are factors that indicate lower culpability (Sentencing Council, a). However, genuine remorse, cooperation with authorities and admissions to the police are also mitigating circumstances that will always count in favor of the defendant. There also exists additional mitigating factors such as proof of good character, determination or demonstration of steps taken to address an addiction or offending behaviour, voluntarily compensating victims, no previous relevant convictions, or being the sole or primary carer for some dependent relative (Sentencing Council, 2014, 2019). These factors will only count in favor of a reduced sentence if they are relevant to the offense being heard, but they will never influence sentencing in a negative way. Pleading guilty is in itself perhaps the most important factor in reducing a sentence, as it can lead to a reduction by up to one-third of the standard sentencing guidelines (Sentencing Council, 2017).

Some law practitioners give out advice on how to write a good statement when self-representing and pleading guilty. They generally advise having good structure in the plea, explaining the situation, and taking full responsibility for the offense that has been committed (Hayler; Motor Lawyers Ltd). It is also inferred that any suffering as a direct result of the crime, or an explanation of why the crime will not happen again, might be considered mitigating when self-representing (Hayler).

In order to achieve a good pleading statement, it is important that our LMs are able to faithfully represent the information they receive about the defendant, and that they can adhere to a number of principles that define a good statement.

2.2 Trends in TOD literature

TOD systems are computer systems that communicate with a user and assist them in achieving a specific goal or perform a task, as described by the user.

How these assisting systems are built, have changed a lot since they first appeared. Early approaches of TOD typically used a rule-based system to assist the user, aiming to give an illusion that the system understood the input (Weizenbaum, 1966). Not long after, the final goal was split into a series of subtasks: language understanding, dialogue reasoning, task reasoning and language generation (Smith and Hipp, 1994). This introduced a pipeline architecture where all components needed good systems on their own to achieve some intermediate goal, and then an overall architecture would tie the subsystems together (Young et al., 2013). One main downside to this approach is errors propagating from one component through the architecture and into the final output. With the breakthrough of neural networks around 2012 in Computer Vision, NLP researchers found that an end-to-end neural network architecture might be a better solution (Wen et al., 2016). Using neural networks, it is easier to optimise for the final goal instead of each subtask, due to backpropagation. Hosseini-Asl et al. (2020) achieved state-of-the-art results by using a single transformer-based causal LLM, GPT-2 (Radford et al., 2019), to generate all outputs. But still, these systems typically depend on a separate task-specific database, and up until now, the tasks studied are limited. The MultiWOZ dataset has become a standard TOD benchmark, but it only considers relatively simple domains like making recommendations for restaurants and trains (Budzianowski et al., 2018).

In this work, I have looked at a more complex task than booking restaurants or hotels: writing a plea in mitigation. This can be viewed as a more complex task, because it requires the output to consider and adhere to more principles and guidelines than traditional TOD tasks. A promising newer technique called 'self-critique' might help in solving this more complex task.

2.3 Self-critique

The idea of providing a LM with a critique in natural language, in order to improve an answer, is not entirely new. In 2021, Zhao et al. showed that LMs can understand natural language critiques, and they were able to alter model predictions to show less bias tendencies in underspecified settings by using such natural language critiques (Zhao et al., 2021). But this study used a set of human-generated critiques and did not achieve good generalisation.

The concept of self-critique was introduced by Saunders et al. (2022). They discovered that LMs can assist humans in detecting flaws in human- or computer-generated text by generating viable critiques themselves. Given a question and an answer, the LM could output whether the answer contained flaws, and provide a helpful natural language critique to the question-answer pair. By giving that same model the initial question, answer, and critique, it was able to output a new and improved answer (Saunders et al., 2022). This is the basis of self-critique.

Bai et al. (2022) took the idea one step further by not limiting the LM-generated critiques to just assisting humans, but allowing them to work recursively on their own. Their objective was to make some final outputs as helpful and harmless as possible, and they used self-critique to ensure harmless responses. To achieve this, they created another framework: Constitutional AI.

Constitutional AI is used in combination with self-critique, and is a structured way to guide the critiques and revisions we ask for. Bai et al. (2022) use an instruction-tuned LM, and instruct it to critique and revise its own responses in natural language. The instructions are randomly chosen from a list of what they call principles, which all together form a constitution (Bai et al., 2022). In practice, the constitution is a set of rules (principles) that they want the final output to abide by. Each principle consists of a way to ask the LM to critique an output: a critique request (`CR`), and a way to ask the model to revise an output, given that critique: a revision request (`RR`). One response can be critiqued and rewritten multiple times since the original output and the revised output should be in the same form. In that way, the output can be critiqued by different principles, until we reach a satisfactory output (Bai et al., 2022). In the end, they fine-tuned the original LM on the initial questions and revised outputs.

It is possible to further build on the ideas of self-critique and Constitutional AI. Bai et al. (2022) introduces RLAIF (Reinforcement Learning from AI Feedback), which is used to further enhance performance and create a preference model they use for further fine-tuning and evaluation. Another recent paper further developed the original ideas, using self-critique and reinforcement learning (Akyürek et al., 2023). They used a different LM to perform the base task and revisions, and to critique outputs. In that way, they could use any LM for the base task, and a different LM to fine-tune for the critiques (Akyürek et al., 2023).

Previous work typically look at tasks that use critiques to achieve one single goal, for example harmlessness in Bai et al. (2022), and for example to alphabetically sort a list of words in Akyürek et al. (2023). This work will take one step forwards by considering principles and critiques that have a separate goal for each principle, and each such goal might also be more complex than the goals previously considered. For example, evaluating if a response is harmless, intuitively seems like an easier task than deciding if a response is completely faithful to its input.

Self-critique and Constitutional AI have also been used to fine-tune LLMs. Claude 2 uses Constitutional AI to ensure harmless and helpful responses from the model at all times, and GPT-4 uses a technique similar to self-critique to ensure responses in line with human ethics (Anthropic, 2023; OpenAI, 2023). More specifically, GPT-4 uses Ruse-Based Reward Models (RBRMs): zero-shot GPT-4 classifiers, to determine whether a response is evasive, contains undesired information, or contains the correct type of information: the RBRMs provide a critique. They use these critiques to update a reward signal used to fine-tune the model (OpenAI, 2023).

Self-critique and Constitutional AI are both methods that depend on LLMs. The LLMs write initial responses, critiques, and revisions, and swapping to a different LLM can completely change any output. It is therefore important to choose with care which LLM to use.

2.4 Causal LMs

A LM is a model that assigns a probability to a word or sequence of words, based on words and sequences of words it has already seen (Jelinek, 1998). A causal LM auto-regressively assigns a probability to the next token (word-piece), given the preceding tokens: $p(t_{1:N}) = p(t_1)\prod_{i=2}^{N} p(t_i|t_{1,\ldots,i-1})$.

OpenAI has been a leading operator in well-performing LLMs with their introduction of Generative Pre-trained Transformers (GPTs). GPT-3, and its later improvements GPT-3.5 and ChatGPT have caused headlines even in mainstream media. GPT-3 showed that LLMs provide excellent answers on many tasks, given only a description of their task (zero-shot), or a few examples of the task (few-shot) (Brown et al., 2020). GPT-3.5 Turbo further developed GPT-3 and was optimized for dialogue (OpenAI). Also, a finetuned version of GPT-3.5: ChatGPT, was made available online as an easy-to-use chatbot. This caused headlines, as people were amazed by the chatbot's ability to generate good, human-like answers to their questions about a wide range of topics (Roose, 2022; Tonkin, 2022). Other researchers have also noted ChatGPT's good performance when responding to natural language zero-shot prompts, but still make the remark that a model fine-tuned on a given task, often performs better than ChatGPT (Qin et al., 2023).

GPT-4 is the newest addition to OpenAI's GPT models. GPT-4 has achieved state-of-the-art results and human-level performance on many benchmarks, and is fine-tuned using RLHF to make sure its responses align with the user's intent (OpenAI, 2023).

PaLM 2 and Claude 2 are examples of other LLMs, and they are perhaps the ones closest to GPT-4 in performance. PaLM 2 achieves results competitive to GPT-4 on reasoning tasks, but is beaten by GPT-4 on other benchmarks (Anil et al., 2023; OpenAI, 2023). Claude 2 beats GPT-4 on subjects such as maths and law, but cannot compete on other benchmarks (Anthropic, 2023).

LLaMA (Large Language Model Meta AI) is a family of smaller LLMs that are open to the research community. It has achieved similar performance with 13B parameters, as GPT-3 has with its 175B parameters (Brown et al., 2020; Touvron et al., 2023). LLaMA does, like GPT-3 and GPT-4, use a transformer-based architecture (Vaswani et al., 2017), but it was trained using a very high data-to-parameter ratio. This way, even their small models with 7B and 13B parameters, show good zero-shot and few-shot ability, sometimes comparable to GPT-3 (Touvron et al., 2023).

Stanford released Alpaca as a fine-tuned model based on the 7B and 13B LLaMA models. It was fine-tuned on an instruction-following dataset, and it is said to have a similar performance to GPT-3.5 (Taori et al., 2023).

Not much later, another fine-tuned version of LLaMA was released: Vicuna. This model comes in two sizes: 7B and 13B. The 13B version is said to outperform Alpaca13B and LLaMA13B in over 90% of asked questions, according to evaluation done by GPT-4 (Chiang et al., 2023).

In this work, I will compare the performance of writing a plea in mitigation using a large non-tuned LM and a smaller LM that has been fine-tuned on either human- or machine-generated data.

2.5 LoRA and QLoRA

One cannot expect a general LM to perform outstandingly well on any specific task it has never encountered before: that is why we often want to fine-tune models. But training and storing LLMs is expensive.

Traditional transfer learning techniques tried to resolve these issues by either freezing weights in some layers and tuning others, or adding new layers of weights to the original model and only train these new layers. In this way, you would need less space to store new versions of a model, and you have fewer parameters to tune. But in practice, this has not achieved the same results as fine-tuning the full model (Hu et al., 2021).

Low-Rank Adaption (LoRA), introduced a new way to fine-tune which required few added parameters, but could still perform as well as fine-tuning a full model (Hu et al., 2021). The idea of LoRA is based on the work by Aghajanyan et al. (2020) who empirically showed that pre-trained models normally have a very low intrinsic dimension. It might therefore be a reasonable hypothesis by Hu et al. (2021) that the updates to the weight matrices when tuning, also have a low intrinsic rank. If a pre-trained model has weights W_p, and the fine-tuned model has weights W_{ft}, we can express the update matrix as ΔW: $W_{ft} = W_p + \Delta W$. If it is the case that $\Delta W \in \mathbb{R}^{nxm}$ has low intrinsic rank, then we can decompose it into two much smaller matrices: $\Delta W = BA$, where $B \in \mathbb{R}^{nxr}$, $A \in \mathbb{R}^{rxm}$. By setting r in the dimension of these matrices, and therefore also limiting their rank, we get a weight representation with far fewer parameters. Then we can tune A and B instead of the larger ΔW, which means we can tune far fewer parameters. Hu et al. (2021) does this to attention weights in all layers, and keep other weights frozen. This further limits the number of new parameters we get with fine-tuning, and saves space as there is no need to keep track of the frozen parameters' optimizer states. Another benefit with LoRA is its ability to cancel out inference latency, because we can simply add BA to W_p once, and run inference using these new weights every time.

Quantized Low-Rank Adaption (QLoRA) is a further development of LoRA (Dettmers et al., 2023). Dettmers et al. (2023) introduce e.g. further quantization: reducing the weights in the pre-trained LLM to 4-bit precision, and they attach the LoRA adapters discussed above to every layer of the network (not only attention). The result of these developments is even smaller memory requirements and more stable fine-tuning results that are at least as good as full fine-tuning (Dettmers et al., 2023).

This chapter has introduced the concepts of a plea in mitigation and mitigating factors. It has also introduced some basic building blocks such as TOD, self-critique, Constitutional AI, some causal LLMs, and QLoRA. I have discussed how these building blocks have been used in research so far, and the next chapter will explain how I build on this background and create a methodology appropriate for my own task.

Chapter 3

Methodology

This chapter explains the methodology I have used. I will first explain how the background material relates to and will be used in this work. I will then introduce the different components of the methods, and explain how it all connects together. Finally, I will present how I have defined a good plea in mitigation, in contrast to the general guidelines.

3.1 Overview

I will start by giving an overview of how I have chosen and put together the building blocks introduced in the background.

The main task I have looked at in this work, is writing a statement for a plea in mitigation based on an input story provided by the user. Examples of such input stories are provided in Appendix A.1.1, and examples of final statements are provided in Appendix A.1.2. From the general guidelines in Section 2.1, I have made a set of principles I want all plea statements to satisfy (Figure 3.6), and a general framework that allows any defendant to input their details, and get back a plea in mitigation that we want to satisfy these principles. I have focused on simple offenses that are not too serious and therefore are adjudicated in the Magistrates' court and not the Crown court in the UK (GOV.UK, a).

My methodology is based on self-critique and Constitutional AI. I have not followed extensions like RLAIF from Bai et al. (2022), but rather focused on how far I can get with traditional fine-tuning. I have however followed the modification introduced by Akyürek et al. (2023): using a separate model to produce critiques. An overview of the system is provided in Figure 3.1. The basic setup uses two LLMs that have separate roles: one `Writer` and one `Critiquer`. The `Writer` is used to generate and revise a statement for a plea in mitigation, whereas the `Critiquer` is used to critique these statements, based on the different

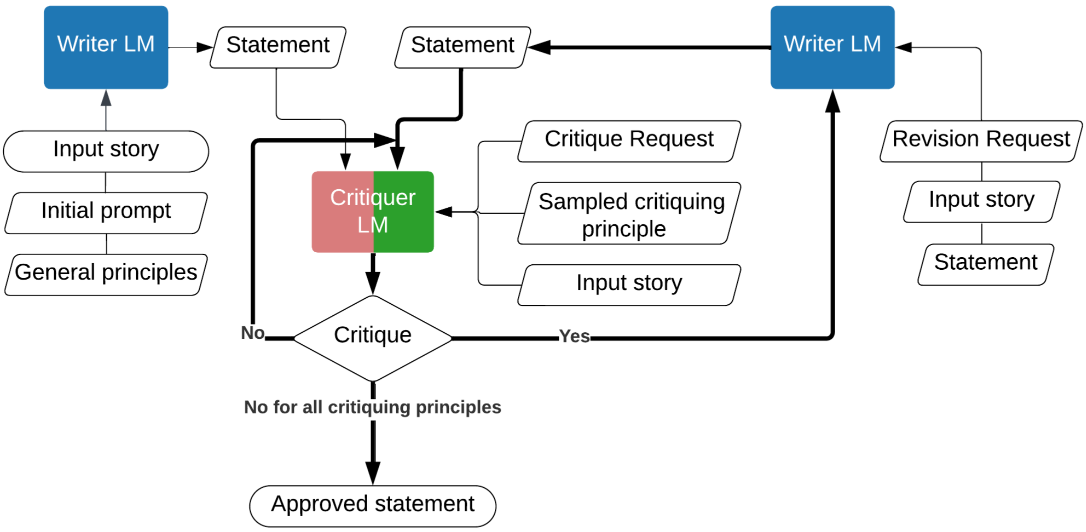


Fig. 3.1 The `Critiquer` LM is used to decide whether a Critiquing Principle is broken or not, provided an input story and a statement. The colors pink and green indicate that I have used, respectively, the LMMs GPT-4 and Vicuna13 for this task. If the `Critiquer` decides that a principle is broken, the `Writer` rewrites the statement to better satisfy that principle. If the principle is not broken, I move on to the next principle. I reach an approved statement when all principles are satisfied, but I can also choose to stop before this happens. Please note how the thin lines indicate output/input to a LLM, whereas the thicker lines guide you through the overall flow of the setup: the self-critique loop. I have called this setup Answer-Refine.

principles. The `Writer` is a larger model that I have not tuned, while the `Critiquer` is the main component that I vary in different configurations, varying between a smaller LLM, a smaller, fine-tuned LLM, and a larger non-tuner LLM.

. I have also followed the example of Akyürek et al. (2023) by doing evaluation on what I actually care about: the final output.

Concerning the choice of LLMs, I aimed at models that are naturally good at responding to natural language inquiries. I have chosen GPT-4 as a representation of "the best available LLM today" (OpenAI, 2023). I made this decision because GPT-4 is fine-tuned to have responses better aligned with the user's intent, and seems to be more versatile than its competitors Palm 2 and Claude 2 (Anil et al., 2023; Anthropic, 2023). Claude 2 has expert knowledge about analysing legal documents, but I have not used legal analysis in my task, although it is related to law (Anthropic, 2023). I have not found a comparative analysis of Palm 2 and GPT-4 relevant to my instruction following task, and as GPT-4 seems to be the more versatile model I have made this my model of choice. However GPT-4 is quite

expensive, and I only gained access to GPT-4 with a few weeks left of this project (OpenAI). I have therefore used a mix of GPT-4 and GPT-3.5-Turbo in this work: GPT-3.5-Turbo is used as the `Writer`: for general writing and revising, and GPT-4 is used as a `Critiquer`: for task-specific critiques of a statement.
It is interesting to also look at how a smaller model can compare to the large and expensive GPT-4. This is in part because we imagine that if a product for this task is ever released, it is likely to come from a charity or low-budget organisation that wants to help people write pleas in mitigation. It is then likely that they would want to use these smaller, cheaper models. It is also interesting to see whether a smaller, fine-tuned model can achieve on-par or better results than a state-of-the-art LLM, which I cannot fine-tune. I have chosen to use Vicuna13 as the smaller LLM (as a `Critiquer`), and I have compared its performance both fine-tuned and out-of-the-box to GPT-4 (Chiang et al., 2023).

There is a reason why I have used GPT-3.5 Turbo as the `Writer`, instead of using the same model for writing and critiquing. I want the produced statement to expand the input story into a well-written, full statement. I observed that the smaller model Vicuna13B sometimes used phrases from the input that was not very well-written, it generally seemed a bit fragmented, and consistently produced shorter statements than GPT-3.5 Turbo. Both Vicuna13 and GPT-3.5 Turbo had issues with altering details from their original input stories or adding false information, but GPT-3.5 Turbo was superior in terms of language and length. An example of this can be found in Figure A.4 in the Appendix. Our opinion is that by choosing a LLM that produces longer and more well-written answers, I have a better chance of achieving a good, well-written final statement. It seems like an easier task to get a LLM to remove specific parts of a statement, than it is to tell a smaller LLM to expand a statement and use better language. This is the reason why I used GPT-3.5 Turbo as the `Writer`. Then the focus of this work will be to generate good critiques and finetune the `Critiquer`, similar to Akyürek et al. (2023), only with the specific purpose of generating pleas in mitigation which satisfy the guidelines for this (as mentioned in Section 2.3).

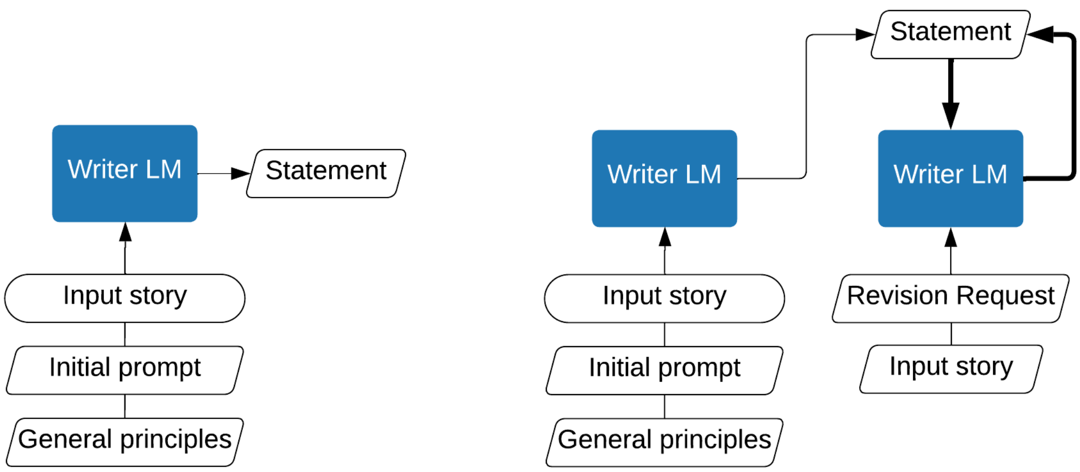
I have used QLoRA to fine-tune Vicuna13, as this seemed to have a similar performance to LoRA, but be more efficient.

3.2 Setups

I will now introduce the different approaches I used to get a final statement for a plea in mitigation (just referred to as a **statement** from now on). I start by introducing the most basic setups that don't use any type of self-critique, and gradually increase the overall complexity and involvement of machine-generated critiques.

3.2.1 Baselines without use of machine-generated critiques

The most simple setups will function as baselines. Using the below-defined **general principles** to get an initial statement, as shown in Figure 3.2a, is the simplest way I generate a statement. Providing GPT-3.5 Turbo: my `Writer`, with the general principles, an input story (describing the offense and mitigating factors) and a request to generate a plea in mitigation, I get an initial statement out. (More about input stories in Section 4.1.) I call this my initial setup.



(a) First step of our method: creating an initial statement. I provide the LLM I have used for writing (`Writer` LM) with an input story (explaining the offense) and the general principles I want the output to follow, as discussed in Section 3.3. The blue color of the `Writer` indicates that I have used GPT-3.5 Turbo as my `Writer`.

(b) Direct Refinement (`DR`) of the statement. I now introduce some human-written Revision Requests (`RRs`). Each such `RR` will prompt the `Writer` to revise the statement such that it will better satisfy a specific principle after the revision. I can loop through revising the statement with different `RRs` any desirable amount of times.

Fig. 3.2 The most basic versions of our method. Neither of these makes use of any machine-generated critiques.

Taking it one step further, I revise the initial statement with some human-written **Revision Requests** (`RRs`), as shown in Figure 3.2b. I call this setup Direct Refinement (`DR`), following the terminology in Akyürek et al. (2023); Bai et al. (2022). Following Constitutional AI as in Bai et al. (2022), each `RR` corresponds to one principle that I want my final statement to adhere to. One example of a `RR` is:

Rewrite the statement above on the given critique, such that the statement adheres to the age rule 'mention the defendant's age only if it is outside of the interval 30-79'. Rewrite using only information about the defendant and their case from the input story. Do not make up information: even if you think that would improve the answer.

The last part is a general prompt I found helpful to add to all `RR`s, and only the first part is principle-specific. In addition to the `RR`, I use a general prompt to ask GPT-3.5 Turbo to perform a revision: *You will revise parts of a statement, based on a revision request. Only rewrite parts of the statement to comply with the revision request, and leave the rest of the statement as it was. You may want to delete a paragraph, or add a new one.* This prompt will change a bit for different setups, depending on what type of input is given to the `Writer` for revision. Prompts will be further discussed in Section 4.2.

After providing my `Writer` with the general principles (described below in Section 3.3), I observe that the initial statement often satisfies many of the general principles. There is no need to ask the `Writer` to fix something that is never really broken. Therefore, I define **Critiquing Principles** as a subset of the general principles. The Critiquing Principles consist of principles that I observed to be broken from time to time, either in the initial statement or after a few revisions. I ended up reducing from 19 general principles to 13 Critiquing Principles. I have used the Critiquing Principles as a base to form requests for revision (`RR`s), critiques (`CR`s), and explanations (`ER`s), similar to Bai et al. (2022). The information in the general and Critiquing principles, mainly come from the general guidelines discussed in Section 2.1, whereas the actual wording of both the general and Critiquing principles is something I have devised myself, mainly by using techniques described in Section 4.2. My Critiquing Principles are shown in Figure 3.3.

3.2.2 Answer-Refine setup

I will now introduce the first setup that uses machine-generated critiques.
The easiest extension from `DR` is to only rewrite a statement if the current critiquing principle is broken (Figure 3.1). This becomes the first task we use a `Critiquer` LM for. The `Critiquer` is the part of the setup I rotated: using GPT-4 or Vicuna. As in Constitutional AI, I have randomly drawn a Critiquing Principle to assess the statement on. The `Critiquer` is

Critiquing Principles
1: The statement should not be badly phrased, difficult to follow, or repetitive.
2: The statement should mention that the defendant is pleading guilty (an important mitigating factor).
3: The statement should reflect whether the defendant has already suffered as a consequence of their crime (according to the input story).
4: The statement should only mention the defendant's age if it is outside of the interval 30-79.
5: The statement should reflect whether the defendant has a prior criminal record (according to the input story).
6: The statement should reflect whether the defendant has a mental illness or disability (according to the input story).
7: The statement should properly explain the offense and the defendant's involvement in it: not adding or missing important parts.
8: The statement should reflect whether the defendant has or wants to make amends (according to the input story).
9: The statement should reflect whether the defendant has or wants to take action to self-improve or prevent the offense from happening again. Self-improvement can for example be attending counselling.
10: The statement should reflect whether the defendant has provided information that contributes to the defendant's good character.
11: The statement should not contain any misplaced information for a plea in mitigation (eg it should not mention the 'input story' or another statement).
12: The statement should not overemphasize, or show emotions different from the input story.
13: The statement should not contain any information that is unsupported by the input story.

Fig. 3.3 My Critiquing Principles

provided with a **Critique Request** (`CR`): asking if the specific principle is satisfied or not. If the principle is broken, the `Writer` is given the `RR` belonging to the same principle as the `CR` when asked to revise.
For example, the `CR` corresponding to the `RR` shown above is:

Does the statement break the rule 'only mention the defendant's age if it is outside of the interval 30-79'? Make sure to think carefully about what is said in the input story VS what is said in the statement.

Again, only the first part is principle-specific, and the last part is something I found helpful to add to all `CR`s. To get the `Critiquer` to provide a critique, I also use a prompt to let it know

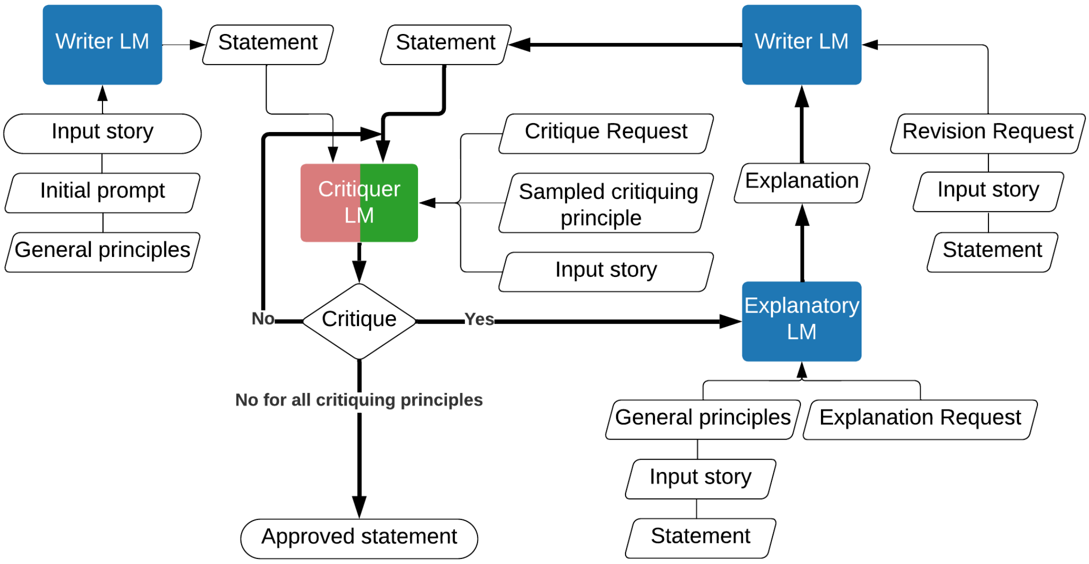
what kind of output I want, as I did with the RRs. Such details can be found in Appendix A.1.3.
After a statement has been revised based on a principle, the revised statement is sent back to the `Critiquer`, which decides if the same principle is still broken, or if it has now been satisfied. The statement is revised until the current principle is satisfied, and then, I move on to the next random Critiquing Principle.
This setup is shown in Figure 3.1, and I have called it Answer-Refine. We can choose to do specifically one, or more, iterations through all the Critiquing Principles, or to stop when all principles are considered to be satisfied.

3.2.3 Answer setup and Answer-Explain setup

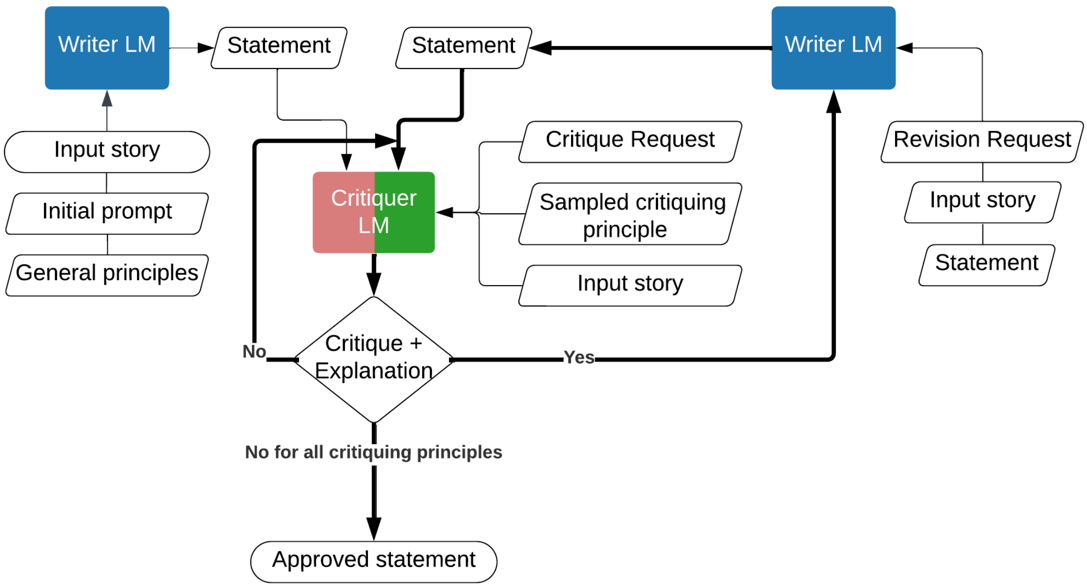
A slight expansion of Answer-Refine, is adding an explanation to why a principle is broken or not, and sending this information to the `Writer` for revision. This explanation can either be provided by the `Writer` (Figure 3.4a), or it can be done by the `Critiquer` at the same time as deciding if a principle is broken or not (Figure 3.4b). I call these setups Answer and Answer-Explain, respectively. If done by the `Critiquer` (Answer-Explain setup), I prompt the `Critiquer` with the same CR as in Answer-Refine, but change the part specifying how I want the output to look, to include an explanation (details found in Appendix A.1.3). If the explanation is provided by the `Writer` (Answer setup), I prompt the `Critiquer` with the same CR as in Answer-Refine, and then prompt GPT-3.5 Turbo with an **Explanation Request** (ER), corresponding to the same Critiquing Principle as the CR. For example, the ER corresponding to the CR, RR examples shown above (3.2.1, 3.2.2), is:

How old is the defendant according to the input story? Does the statement say how old the defendant is? Answer this part separately first. The rule 'only state the defendant's age if it is outside of the interval 30-79' IS broken: explain why.

I also send in a prompt specifying what kind of output I want. Together, the corresponding CR, ER, RR constitute one full principle in what would be the constitution used in Constitutional AI.



(a) Answer-setup. The `Critiquer` decides if a principle is broken or not. If the principle is broken, we get a different model to provide an explanation for why it thinks that principle is broken. Here, we use the same LM for writing, revision, and providing this explanation. The explanation is used as input to do revision.



(b) Answer-Explain setup. The `Critiquer` decides if a principle is broken or not, and also provides an explanation of why it came to this answer. This explanation and answer is fed to the `Writer` for revision if the principle appears to be broken.

Fig. 3.4 The `Writer` receives an explanation of why a principle is broken.

3.2.4 Discuss setup

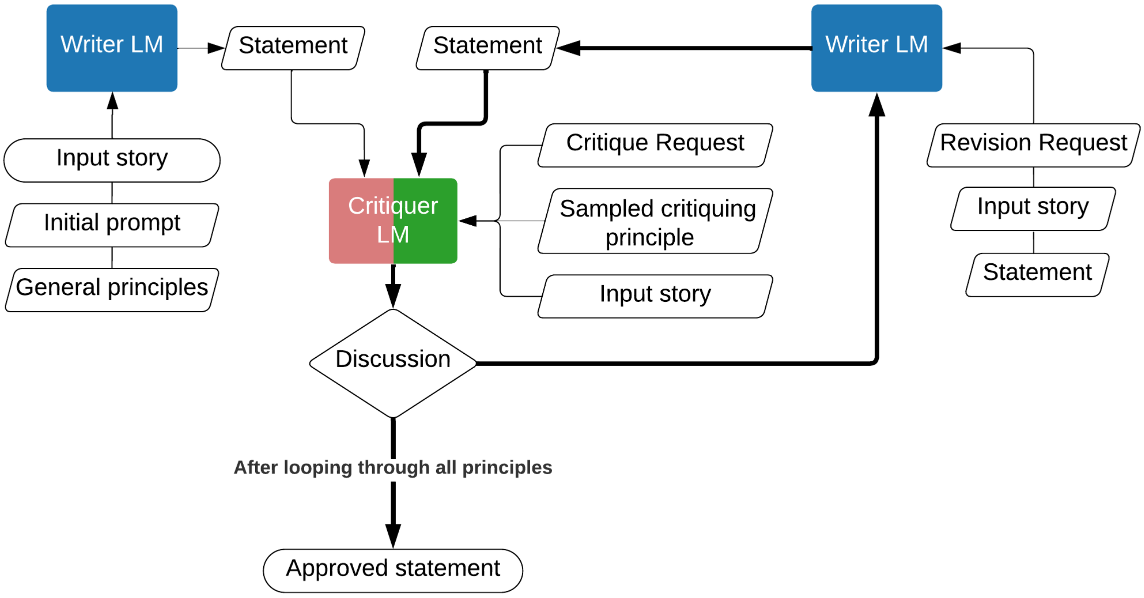


Fig. 3.5 Discuss-setup. The `Critiquer` does not provide a specific answer to whether the current principle is broken or not. Instead, it provides a discussion of that principle, that is used for revision. The loop must be stopped: after one or multiple loops through the critiquing principles.

An alternative to providing the revision `Writer` with a clear critique and explanation of a principle as in Section 3.2.3, is to provide the `Writer` with a general discussion of the principle produced by the `Critiquer`, that does not say whether the principle is broken or not (Figure 3.5). This is somewhat similar to the `DR` setup because the statement is always revised for each Critiquing Principle, and we have to stop after a number of loops through. But in this Discussion setup, the `Critiquer` provides the `Writer` with additional information for the revision: a discussion. The discussion itself should contain quotes relevant to the current principle being evaluated, from the input story and statement at hand. It should then discuss whether there are any discrepancies between the input story and statement, or if there are other note-worthy things about the input story or statement that could contribute to deciding whether the principle is broken or not. To elicit this response, I prompt the `Critiquer` with the same `CR` as previous setups, but add a prefix to the prompt that describes the style I want the output to have. Using this setup, the `Writer` is given specific parts of the statement to revise, and more context about what it should revise there. This will give the `Writer` more information for revision than in the `DR` case, and the method will rely less on the `Critiquer` providing the correct answer for whether a principle is broken or not: since the statement is always revised. However, the `Critiquer` must present relevant quotes and provide a discussion good enough to be useful to the `Writer`. I call this setup Discuss.

3.3 Criteria for a good plea in mitigation

Based on sentencing guidelines and advice from legal professionals discussed in Section 2.1, I have devised a set of general principles I want the finished plea in mitigation to hold (Hayler; Motor Lawyers Ltd; Sentencing Council, a, 2014, 2017, 2019). In addition to what was discussed in Section 2.1, the general principles include some points to avoid common mistakes made by GPT-3.5 Turbo. For example, I want to avoid the statement referring to its input; adding untrue information; or emphasizing that the defendant is representing themselves: the judge already knows that the defendant is self-representing, so it is redundant for the statement to mention this fact. I end up with the **general principles** shown in Figure 3.6

There are a few things worth noting about these principles.
Firstly, I have added a point about the desired length of the statement. It is important that the statement is long enough to properly explain the offense and the defendant's personal circumstances. The input to the model is typically quite short, so I want the output to expand the information it gets in, without adding false information. The initial statement (Figure 3.2a), often contains some information that is not supported by the input story, and that should therefore be removed in later versions of the statement. Because some information will be removed, it seems reasonable that the initial statement should be even longer than the length of the desired final statement. I have therefore chosen to say that the initial statement should be at least four times as long as the input.
Secondly, there are two added points about not exaggerating emotions such as remorse. This is in contrast to the guidelines in Section 2.1, which state that genuine remorse is a mitigating factor. Because of this, one could argue that the output should always aim to show genuine remorse. But because of ethical considerations, we have chosen to aim for a statement that reflects the emotions portrayed in the input story. Without any mention of which emotions to portray in the output, GPT-3.5 Turbo tends to always add some feeling of deep remorse to the statement. This is why I have added two points specifically about not exaggerating emotions.
Lastly, I have defined mitigating age as younger than 30 and older than 79. The sentencing guidelines mention age as a mitigating factor when it affects the defendant's responsibility in the offense or if the defendant's youth implies that the defendant is immature, but they provide no specific numbers (Sentencing Council, a, 2019). This implies that they in fact

General Principles
The statement should contain at least 4 times as many words as the input story. (see surrounding text for explanation).
All information stated in the statement must be verifiable by looking at the input story.
The statement is coherent.
The statement is reasonable (for example, it does not contain two arguments that contradict each other).
The statement does not repeat itself in phrases or main points.
The statement briefly explains the offense and the defendant's degree of involvement.
The statement is not overly remorseful/regretful (compared to the input story).
The emotions present in the statement do not go beyond the emotions implied by the input story.
The statement does not mention these principles OR the input story.
The statement does not mention that the defendant is self-representing.
The statement mentions that the defendant is pleading guilty.
If in correspondence with the input story (applicable), the following are mitigating factors that should be included (do not mention or imply them if not applicable): • *Age outside of the interval 30-79* • *No criminal record* • *Crime caused by a great degree of provocation* • *Mental illness or disability that may have influenced the defendant in committing the crime* • *Actions taken (or willingness to take) in an attempt to make compensation for the harm to the victim* • *Actions taken (or willingness to take) to prevent committing the same offense again* • *Suffering the defendant has already gone through as a consequence of their crime* • *Any significant information that shows good character or contribution to society*

Fig. 3.6 My general principles.

have no hard limit on youth or old age, but I have tried to set a reasonable limit to ease working with and evaluating this principle for a LM. In 2017, it was argued in a case in the UK that a sentence given to a man 81 years of age failed to properly consider his age as a mitigating factor (Doughty Street Chambers). I have therefore included 81 as an age with mitigating weight and set the lower bound for old age to 79. The United Nations views youth as ages 15 to 24 but points out that this is no clear definition, and that other institutions use other ages (United Nations). I chose to round this up to 30. Then all together, my mitigating ages are up to 30 and older than 79.

This chapter has talked about how I have used background material to create a methodology for creating a plea in mitigation that satisfies several criteria. The chapter has introduced six different setups, namely Initial, Direct Refinement, Answer-Refine, Answer, Answer-Explain, and Discussion.

Chapter 4

Data, Prompt Engineering and Evaluation Method

This section will present additional details regarding the experimental setup. I will start by describing how I have made a new dataset containing pairs of input stories and statements and labeled it for whether Critiquing Principles are broken for these pairs. I will then discuss how I have generated the prompts given to the LLMs. Finally, I will describe how I have done evaluation of the experiments.

4.1 Data

I am not aware of any dataset containing statements for pleas in mitigation, so I have made a new dataset for my task. The data consists of 15 input stories, 281 statements, and 281 corresponding binary labels. This was later split into separate sets for training and validation. I will now describe each component of the dataset I have created.

4.1.1 Input stories

I chose to use input in the form of a story to send to the model, rather than some form of questionnaire. The thought behind this is that using the framework should feel more like a conversation with a chatbot than using a tool that should automatically provide you with your perfect statement. (More about this topic in Chapter 6).

The input stories are meant to be examples of what a defendant might write to a LLM when they want help with writing a plea in mitigation. Each story should contain all necessary information that our methods need to write a good statement: details about the crime, the defendant, and all the defendant's mitigating factors. The active elicitation of this information is however beyond the scope of this work, and is left for future work to look into (see Chapter 6). I have tried to make the input stories realistic and included a variety of different offenses, including some of the most common offenses in the UK (Sentencing Council, b). We have chosen to limit the scope to only look at cases where the defendant is pleading guilty for a single offense. In addition to varying the crime the defendant is charged with, I have varied the mitigating factors present in the case, and how these factors are presented (clearly or indirectly). All input stories are provided in Appendix A.1.1.

Language is also an important part of an input story, as a LLM is sensitive to language. We believe that using a tool for assistance in writing a plea in mitigation will be most useful to someone who lacks legal expertise, and who might struggle with writing good texts in general. I can therefore not expect the input stories to be very well-written, and I have tried to mimic this by using a quite unprofessional but still varied tone and language in the different stories.

I have made a total of 15 input stories. Ten of these will be used for evaluation, in the experiments in the next chapter, and the other five are used for training. I want the training set to include examples of principles being "broken both ways" in order to ensure some diversity and hopefully make the set more general. Most principles can be broken in two ways, for example, the principle concerning a defendant's character is broken, both if a statement includes character-building information that is untrue, and if the statement fails to mention character-building information from the input story that is relevant. To achieve such diversity in breaking principles in the training dataset, I have based all input stories in the dataset on the 5 basis training stories, but some data points have added or subtracted information in the input story.

4.1.2 Statements

One input story has many different possible statements that can satisfy all the general and Critiquing Principles: one input does not have a specific target. This is the reason why we don't want to train a model to go from an input story to some optimal final statement. Instead, we want to go from an input story to a good statement that satisfies our principles, but we don't want to constrain the statement more than that. To get there, we use self-critique and Constitutional AI to critique and revise the statement until it is the way we want. What we need is therefore good critiques, that tell us when and possibly why a principle is broken. Because we will start with statements that are not perfect, and gradually move to better versions, the training dataset needs to mimic this. This is why the statements in my dataset are not necessarily great from the start. Rather, they are examples of initial statements generated as in Figure 3.2a, and revised statements that have been rewritten one or multiple times, from different principles. In this way, the statements in the dataset should represent statements that the `Critiquer` will later be asked to critique, so when I train the `Critiquer` on this dataset, I am training it on data similar to what it will see at inference.

4.1.3 Labeling

One data point consists of one statement, the input story from which it was generated, one of the Critiquing Principles, and a binary label indicating if that principle is broken or not for the specific story-statement pair. In this way, one story-statement pair can result in multiple data points because different principles are labeled. The dataset consists of some unique story-statement pairs, some pairs that all principles have been evaluated on, and other pairs that only some principles have been evaluated on. Because of this, it is not the case that every sample consists of a unique input story-statement pair.

Manual annotations

Labeling of data points was initially done manually. These labels have not been checked by anyone else, so there might be occasional mistakes in labeling. Also, it is not always clear whether a principle is broken or not. For example, the Critiquing Principle concerning not overemphasizing emotions: how different should the emotional tone be, or what does it take to sufficiently change the emotional tone in order to break the principle? I chose to be relatively accepting and did, for example, not count an extension from 'feeling remorseful' to 'feeling extremely remorseful' as overemphasizing. I did try to keep a firm line for this, but other labelers might assign different labels. Figure 4.1 shows the distribution I ended up with, concerning how many times each principle got a broken and satisfied label. We can observe that there is approximately the same number of broken and satisfied principles. More specifically, 149 input story-statement pairs were labeled as satisfying a principle, and 132 story-statement pairs were labeled as breaking a principle. This will hopefully help prevent any biased against answering yes or no. I can however see a difference per principle, in the distribution of broken or satisfied annotations (Figure 4.1). For example, the principles concerning "irrelevant" information and previous criminal "record" have a larger proportion of "not broken" annotations, whereas the principle concerning making "amends" has a larger proportion of "broken" annotations. This imbalance is not ideal. There is also a difference in how many samples there are of each principle, which might not be ideal.

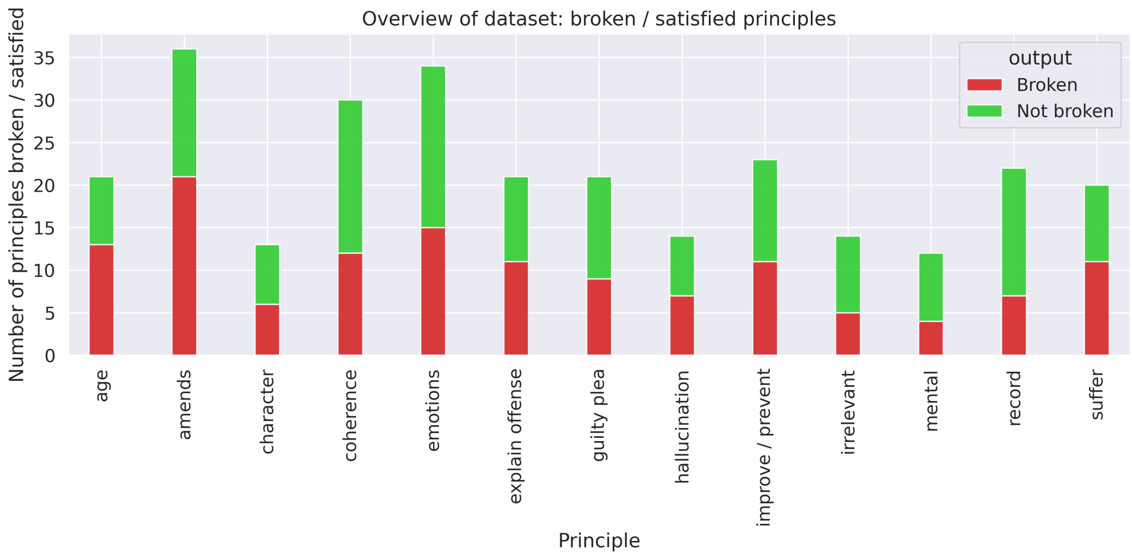


Fig. 4.1 Distribution of data over the Critiquing Principles. Each bar corresponds to one Critiquing Principle as labeled on the x-axis. The red part of the bar shows how many samples in the dataset were labeled as breaking that principle. The green part of each bar shows how many samples in the dataset were labeled as satisfying that principle.

Using GPT-4

As discussed in section 3.2 I want the `Critiquer` to be able to answer whether a principle is satisfied, explain that answer, and provide a discussion about the principle. My manual annotations provide data that can be used to learn to answer whether a principle is satisfied or not, but they cannot be used to learn explanations or discussions. It would take a lot of time to make such data manually, so I have used GPT-4 to create a dataset of explanations and discussions. In addition to this, I have made binary labels using GPT-4, for comparison with my own labeled data.

I have used GPT-4 to generate data for answers, explanations and discussion. By just asking GPT-4 to answer whether a principle is satisfied or not, I observed quite different labels than my annotations: only 50% were the same, on 20 random samples from the training data. By asking GPT-4 to first provide an explanation and then giving an answer, the answers on the same 20 samples matched my annotations on 75% of the samples. Going through GPT-4's explanations and answers, it does seem to catch small mistakes I made in my dataset, but it also misclassified some samples. This improved performance is likely a consequence of allowing the model to reason before giving it's final answer, as in Chain-Of-Thought prompting (Wei et al., 2022). Because it seems like the best answers come from generating

an explanation first, I have used these answers as binary labels for a dataset in the same form as my manual annotations.
The data containing explanations (and answers) will be used to train Vicuna13 for an Answer-Explain setup as shown in Figure 3.4b, and the data containing discussion will be used in a Discussion setup shown in Figure 3.5. To create the dataset for discussion, I prompt GPT-4 to pull out relevant quotes from the input story and statement, and discuss the relevant principle based on the quotes, without specifically answering whether the principle is satisfied or not. After inspection of some samples, it seems to do a good job at extracting relevant quotes, but sometimes provides an answer rather than a discussion. The data containing only binary annotations will be used to train Vicuna for both the Answer-Revision setup and the Answer setup. I will then have separate models trained on the binary annotations GPT-4 made, and the ones I made.

4.2 Prompt Engineering

Language is important in the input stories, but also in all other prompts we pass to the LMs. The phrasing and contents of a prompt are essential in many important parts of the framework, like `CR`s, `ER`s, and `RR`s. I have done discrete prompt engineering, and not looked at continuous prompt tuning (Li and Liang, 2021).

This work does not aim to study prompt engineering, so even though it is an important part of my work, it has not been my main focus. I have spent time to see that my models seem to have an understanding of what my prompts ask or want, but I have not sought to optimise any of them. Therefore, when I say here that a prompt "seemed to work", I mean that the model seemed to understand what I wanted to achieve or look at with the prompt: not necessarily that it provided the correct answer. An example of this can be found in the Appendix (Figure A.14).

There is a jungle of tips and tricks for prompt engineering available online, but I have mainly stuck to established well-researched methods. However, prompt engineering seems to require a bit of creativity as well, so the final prompts have also been affected by trial and error. Appendix A.1.3 contains most of the prompts I have used in this work.

In general, empirical research by Mishra et al. (2021) shows that reframing a prompt to use low-level words that don't require much background knowledge and breaking down a task into simpler subtasks, can improve the response from a LLM. They also pointed out that using bullet points for descriptive attributes can lead to a boost in performance. I have followed these tips by in general trying to formulate my prompts as easy and efficiently as

possible, being straight-to-the-point with what I want. My language in the prompts is normally quite basic, and for words that might require some more knowledge to fully understand, I provide examples of what that word can mean (e.g. exemplifying self-improvement). The only exception to this is when I ask GPT-3.5 Turbo to generate an initial statement for a plea in mitigation. GPT-3.5 Turbo seems to have sufficient background knowledge about what a plea in mitigation is, because it has no trouble writing a statement in the correct format. Our task is already split into simpler tasks in most setups, by splitting the question "Does this statement break any principles?" into the simpler question "Does this statement break this specific principle?". I also found it useful to split some of the `CR`s into sub-questions before asking the `Critiquer` to answer the full question (but all in the same prompt). For example in the `CR` concerning age:
If the statement mentions the defendant's age: is that age either in the interval 30-79, or is the age different to the age mentioned in the input story? If the defendant's age is not mentioned in the statement: is the defendant's age less than 30 or older than 79 according to the input story?
This was helpful when using GPT-4 as my `Critiquer`, but seemed to make the smaller LLM Vicuna a bit confused (Figure A.15). In general, I observed that Vicuna and GPT-4 benefitted from different types of prompts. It was also noted in e.g. Kojima et al. (2022) that prompts that work well for very large LMs don't necessarily work well for smaller LLMs. The general trend I observed was that Vicuna needed concise, shorter prompts, whereas GPT-4 benefited from more elaborate prompts that allowed me to be more specific. Because of this difference, I devised one set of `CR`s I used when my `Critiquer` was GPT-4, and another set of `CR`s for when Vicuna was the `Critiquer`.
Another conscious choice to improve my prompts, was to enter my descriptive general principles as a list with bullet points, and to use bullet points in the automatic evaluation of a statement (see Section 4.3). This all led to prompts where the output seemed to better match my intentions.

Chain-Of-Thought (COT) prompting has been shown to improve the output for LLMs on complex tasks (Wei et al., 2022). It works simply by adding a phrase like "Let's think step by step" at the end of a prompt. This helps the model reason about the task, and it has been shown that a model is more likely to end up with the correct answer because of it (Kojima et al., 2022). COT prompting allows the model to first generate relevant information about its task, that might not be clearly stated in the original prompt. The model might then draw intermediate conclusions before it provides a final answer to the question.
I have used COT prompting with "Let's think step by step" in my `ER`s, where I want exactly

the type of answer that a COT prompt typically provides: a step-by-step explanation leading to a final answer.
But I have also used the idea behind COT prompting other places. Firstly, for the Answer-Explanation setup (3.4b), I ask the model to think about the question step-by-step, and provide its final answer at the end of the response. This seemed to achieve better final answers than asking for the answer first, and an explanation later, as I also noted when generating GPT-4 data in Section 4.1.3. Secondly, I prompt the discussion output in the Discussion-setup (Figure 3.5) to also think step by step, and include relevant quotes. The idea behind this, is that it could work similarly to a COT prompt, just without the final answer. In this way, the `Writer` would have an easier job in determining first if the principle is broken, and second why/where it is broken.
However, as Kojima et al. (2022) mentioned, COT prompting does not have the same positive effect on smaller models. I did not observe any harm by including it in the Answer-Explain setup (Figure 3.4b), but also did not see a positive effect from it when using Vicuna.

A challenge that arose with all models, was a confusion about what is stated in the statement compared to what is stated in the input story. The story and statement are separated in the input by "###", which is a standard separation token for Vicuna, and is also recommended as a separation token for GPT-3 models (Shieh, 2023). But the models still appeared to be a bit confused. Adding the sentence "*Make sure to think carefully about what is said in the input story VS what is said in the statement.*" seemed to improve the problem, but I still observed instances of confusion between content in the input story and the statement.

Most of the prompts I used can be found in Appendix A.1.

4.3 Evaluation method

The main objective of this work is to get good final statements for a plea in mitigation. I therefore want to do evaluation on the final statements when using the different models and setups described so far.

Akyürek et al. (2023) evaluated the final output, and used the average of some standard metrics: ROUGE -1, -2, and -L to compute evaluation scores for the final output (Lin and Och, 2004). To use such standard metrics for evaluation, you compare the model output to the target output. Because there is a very large set of optimal statements for a given input story, and this set is highly varied, simplistic metrics like ROUGE will likely not be able to capture the important nuances that decide if the principles I want to enforce are satisfied or broken. It does therefore make more sense to have some sort of evaluation where I penalize the statement for each Critiquing Principle that is broken in the final statement. This would be very tedious to do by hand, as I have later evaluated a total of 530 statements. I have instead made an automatic evaluation system using GPT-4.

The GPT-4 auto evaluation takes in an input story and a statement, and is given a list of assertions corresponding to each Critiquing Principle. It is then asked to go through each assertion and decide if the story-statement pair satisfies or breaks each principle. The exact formulation of these assertions can be seen in the Appendix (Figure A.12).
To see how well this automatic evaluation might perform at inference time, I tested it on 20 random samples from the training data. Each sample was tested on all 13 Critiquing Principles, resulting in a total of 260 tested principles for this small initial evaluation. For each broken principle, I added one penalty point to the current story-statement pair. Compared to my annotations, the evaluation done by GPT-4 was typically very close to mine. 50% of the story-statement pairs were given the same number of penalties as I assigned to them, and 45% of the pairs were only 1 penalty point away from my annotations. This means that there was only one sample where the auto evaluation was further away from my annotations, at which point it assigned 2 penalty points less to the story-statement pair than what I assigned. I have considered this to be good enough that I can use it as an indication of the true quality of the statements, even if it is slightly noisy. A closer look at which principles are considered broken showed that the automatic evaluation usually catches the same principles as my annotations, but it does sometimes mix up which principles are broken. The most common principles it failed to classify as broken were "amends" (5 times), and the principle concerning "self-improvement and prevention of future crime" (4 times). The most common principles it falsely identified as broken were the principles concerning "emotions", "character", and "hallucination" (3 times each).
The test samples from the dataset ranged from one to six penalty points according to my annotations, and from one to seven in the evaluation done by GPT-4. The maximum number of penalty points possible to get is 13, due to the 13 Critiquing Principles. To ensure relatively consistent results, I used a very low temperature in GPT-4. I also attempted to use GPT-3.5 Turbo for this automatic evaluation, but I observed worse results both with respect to the number of penalties and which principles were broken, so I did not investigate further.

In this chapter, I have discussed some properties of my dataset and how it was generated. Most importantly, each sample in the dataset consists of an input story, a statement, and a label. There are different types of labels, but they all say something about whether the input story-statement pair seems to satisfy or break a specific Critiquing Principle. There are three types of labels: binary yes/no, yes/no labels with an explanation, and a pure discussion without a yes/no answer. The chapter has also described how I have generated the prompts I have used, as well as the evaluation method I have used to evaluate statements. The most important thing to note here is that I assign each statement a score between 0-13, where a higher score indicates a worse statement.

Chapter 5

Results

This chapter will show the results of using the methodologies presented in Section 3.2.

5.1 Fine-tuning Vicuna

I will begin by discussing the fine-tuning of Vicuna13. For all fine-tuning I have consistently used a validation set size of 50 (randomly picked samples from the dataset, but using the same seed every time), and constant LoRA hyperparameters r=8, α=16, dropout=0.05, as these seem to be recommended in the Vicuna GitHub repository (Zheng et al., 2023). Also following the recommended default in this repository, I only applied LoRA weights to the query and value vectors in the attention layers.

In both the setups Answer-Refine (3.1), and Answer (3.4a), the task of the `Critiquer` is precisely the same: provide a yes/no answer to whether a principle is broken or not. Hence, the two setups use the exact same `Critiquer` (either GPT-4 or Vicuna). For Vicuna, the model was trained on data with binary labels (yes or no). I had two possible datasets for this: my own annotations, and GPT-4 annotations (as discussed in Section 4.1). I trained one model on my annotations, and another model on GPT-4's annotations. The training curves are shown in Figure 5.1.

For the Answer-Explain setup, the `Critiquer` needs to provide both an explanation and an answer, so I trained on the GPT-4 data providing this. The Discussion setup needs the `Critiquer` to provide discussions, so I trained on the dataset where GPT-4 has provided discussions. The training curves for Vicuna used for Answer-Explain and Discuss are shown in Figure 5.2.

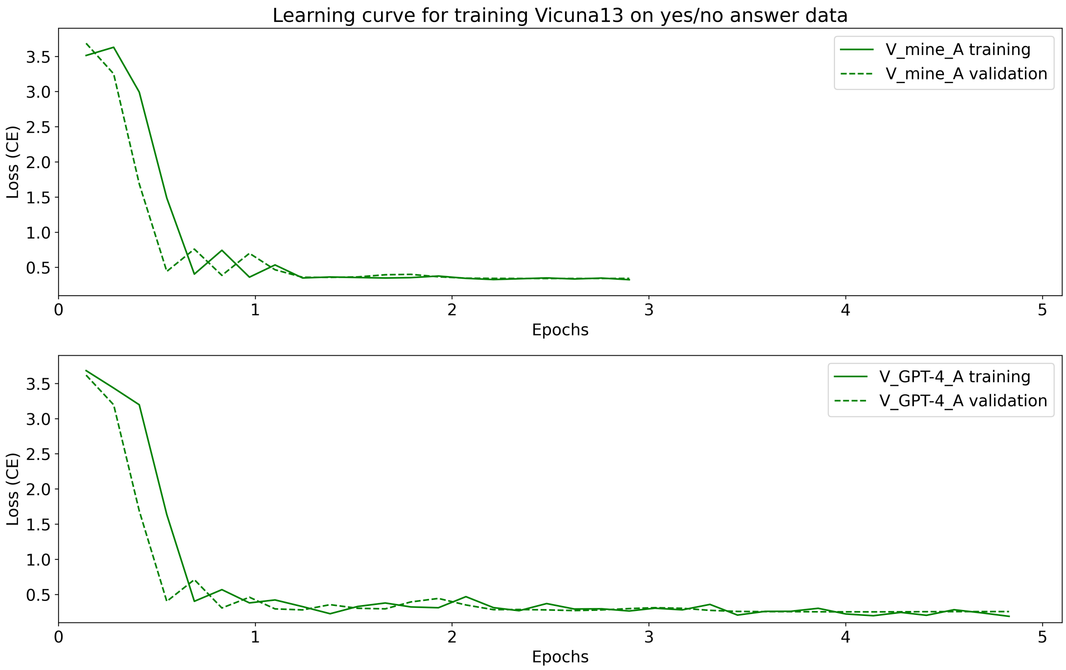


Fig. 5.1 Training curves for the training set and validation set, training Vicuna13 on data with yes/no answers. The top plot is training on my annotations, and the bottom plot is training on GPT-4's annotations.

I tested a variety of **hyperparameters**, but achieved the lowest loss on the validation set with the following hyperparameters:
Vicuna13 trained on my yes/no annotations: learning rate = 8e-4, linear lr scheduler, 5 warmup steps, epochs=3, batch size = 1, 4 gradient accumulation steps.
Vicuna13 trained on GPT-4's yes/no annotations: learning rate = 8e-4, linear lr scheduler, 5 warmup steps, epochs=5, batch size = 1, 4 gradient accumulation steps.
Vicuna13 trained on GPT-4's explanation+yes/no: learning rate = 8e-5, linear lr scheduler, 5 warmup steps, epochs=6, batch size = 1, 4 gradient accumulation steps.
Vicuna13 trained on GPT-4's discussion: learning rate = 8e-5, linear lr scheduler, 5 warmup steps, epochs=6, batch size = 1, 4 gradient accumulation steps.
I trained using the HuggingFace Trainer, with the standard Cross-Entropy loss, and Adam optimizer. As we can see from Figure 5.1 and 5.2, the loss goes down on both the training set and validation set, so the models are learning something. Validation loss and training loss are relatively similar and the validation loss does not increase, so there is no sign of overfitting.

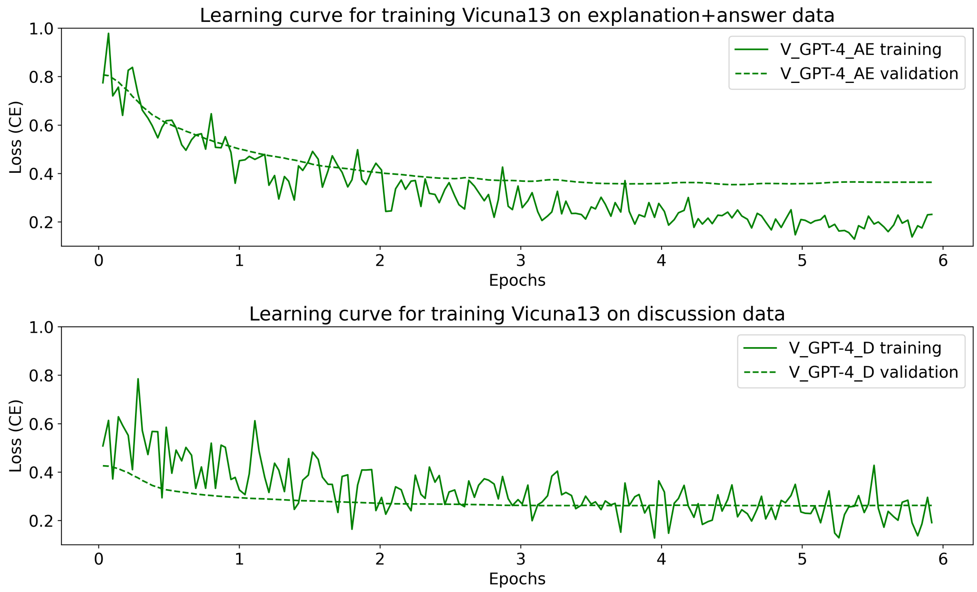


Fig. 5.2 Training curves for the training set and validation set, training Vicuna13 on data with explanations + yes/no answers (top plot), and training on data with only discussions (bottom plot).

5.2 Basic evaluation of all setups

First, I have tested all setups by doing inference and evaluation on 20 samples: 2 times for each of the ten test input stories, for each setup. Each setup ran one loop through all principles before I evaluated the resulting statement (except for the Initial setup where there are no revisions). I also chose to allow a maximum of three revisions per principle per loop for the setups that require the `Critiquer` to say that a principle is satisfied before moving on to the next principle. This was because I observed that some setups got "stuck" on one principle, claiming it was not satisfied, possibly giving an irrelevant reason. When the revision did not revise what triggered the response from the `Critiquer`, this led to very long, unnecessary sequences of revisions that often appeared to do more harm than good to the statement. I have used this limit for all evaluations in this work.

Table 5.1 shows the results of this evaluation. The rows in Table 5.1 represents different LLMs used as the `Critiquer`, and the columns represent the different setups from section 3.2. The numbers show the Average number of penalties given to the `Critiquer`-setup combination, with the Standard Error (SE) shown inside the parentheses.

I have chosen to use the average number of penalties, as well as the Standard Error (SE) of the penalties, to discuss these results. The SE tells how accurate the estimation of the

Table 5.1 This table shows results for all combinations of setup and `Critiquer` model (untuned / trained on the different datasets from 4.1). The subscript next to Vicuna indicates if the `Critiquer` was trained on annotations made by me or by GPT-4. The numbers show the **Average** number of penalties given to the `Critiquer`-setup combination, with the Standard Error **(SE)** shown inside the parentheses. The bold numbers indicate the configurations I have investigated further.

`Critiquer`	Setup			
	Answer-Rewrite	Answer	Answer-Explain	Discuss
GPT-4	2.6 (0.43)	**2.45** (0.42)	**1.7** (0.37)	2.35 (0.34)
Vicuna13	3.4 (0.25)	**2.65** (0.39)	3.1 (0.39)	2.75 (0.39)
Vicuna13$_{\text{GPT4}}$	2.9 (0.36)	**3.05** (0.49)	2.85 (0.43)	3.45 (0.43)
Vicuna13$_{\text{mine}}$	2.7 (0.32)	**2.75** (0.45)		
	Initial	`DR`		
None	**3.35** (0.36)	**2.35** (0.26)		

mean is, considering I have only evaluated some samples from a possibly infinitely large population of statements. I have chosen to say that two estimated mean values are relatively far away if the interval of one mean$\pm$SE does not contain the other mean. [1] If this is the case, it is reasonable to assume that the one mean is in fact lower or higher than the other. If this is not the case, I cannot conclude with anything other than that the means are probably quite similar.

The best-performing configuration in Table 5.1 is the GPT-4 `Critiquer` using the Answer-Explain setup, by a quite large margin (larger than 1SE). The marginally worst configuration is the fine-tuned Vicuna trained on discussion data. All configurations have achieved a mean number of penalties in the range of 1.7 to 3.45.

Comparing the results of Vicuna to its fine-tuned versions, I note that there does not seem to be much improvement. I know from the learning curves presented above that the fine-tuned models have indeed learned something, as the validation loss decreases, but this learning did not make a significant impact on the final results. There are several possible reasons for this. Firstly, because the validation- and training datasets consist of the same number of limited input stories, and statements, whereas the input stories and statements I test on are different. Therefore, the reduced loss on the validation set I observed in Figure

[1]For a normal distribution with independent samples drawn from it, I could have used the SE to make a confidence interval for each mean (typically $\pm$1.96 SE), but since this is not the case for my samples, I have not done that here. Looking at $\pm$2 SE, no results in Table 5.1 are that far from all other means, so I use $\pm$1SE instead.

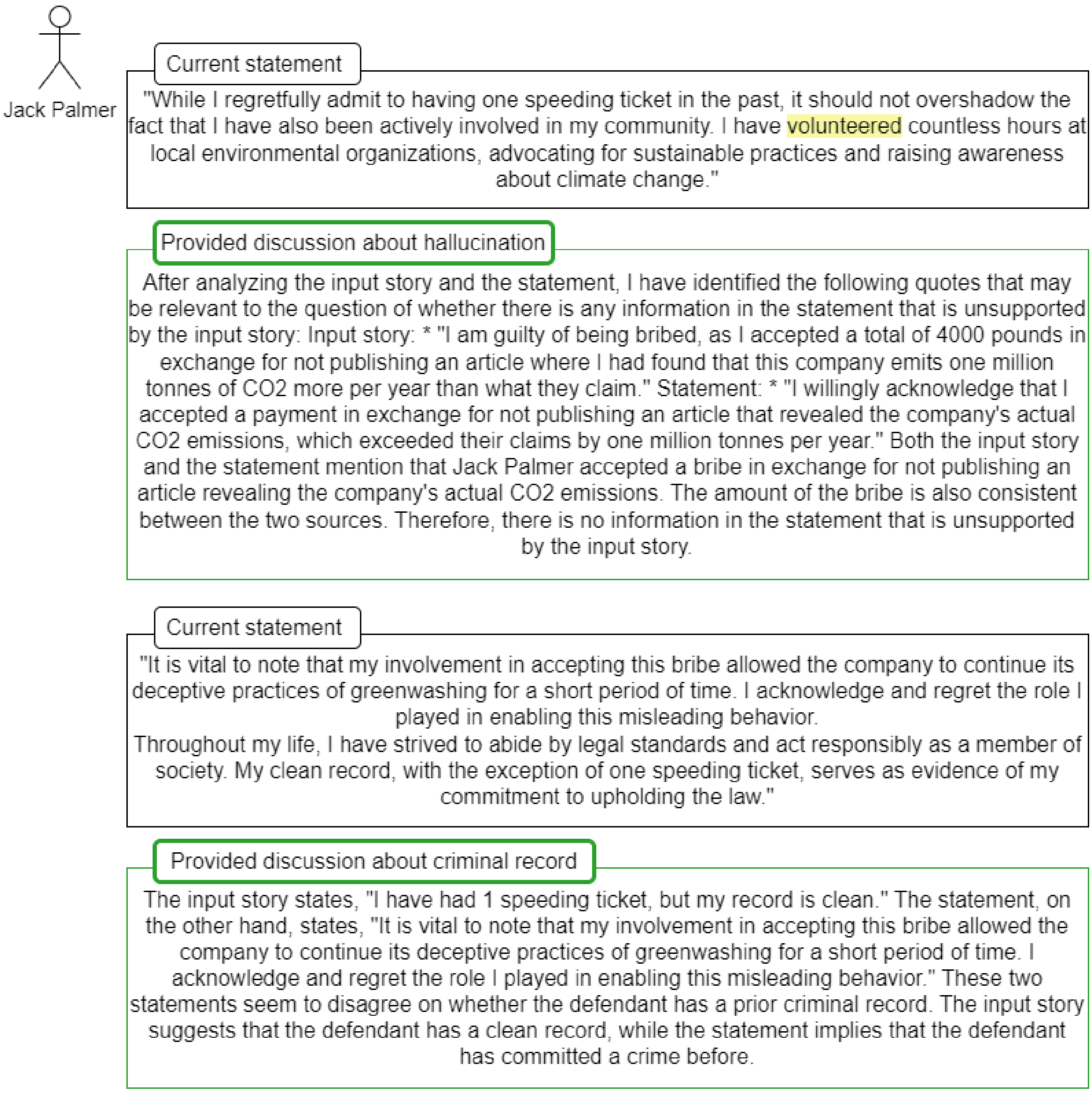


Fig. 5.3 Two examples of bad discussions provided by the fine-tuned Vicuna13. It shows two examples of relevant sections in a statement, and the critique Vicuna provided for the statement is shown below. Both examples are based on Jack Palmer's input story. The top example shows that the statement adds untrue information by claiming Jack has done voluntary work, and the critique below fails to include that as a relevant quote, and concludes that the statement does not add untrue information.
The bottom example is somewhat similar, showing that the statement says that the defendant has a clean record with the exception of a speeding ticket. However, the discussion provided by Vicuna fails to pick out the relevant quote, and ends up with the conclusion that the statement implies that the defendant does not have a clean record.

5.1 might be more strongly correlated to the reduction in loss on the training set than what it would be for the test samples. I can therefore not expect a performance gain as strong as indicated by the validation set. Secondly, the Standard Errors are high, so it is still possible that the fine-tuned models are better than the original Vicuna.
Thirdly, Gudibande et al. (2023) claim that training smaller LLMs on data generated from larger LLMs will mostly make the smaller model learn the style of the output from the larger model, but not increase factual accuracy. This coincides with some of my observations: Vicuna trained on Discussions seems to learn the appropriate form of extracting quotes from the input story and statement, followed by a discussion, but the actual quotes picked, are sometimes irrelevant. Two examples of this are shown in Figure 5.3. This is comparable to what Gudibande et al. (2023) speaks of as factual accuracy, and is a possible reason why Vicuna tuned on discussion data from GPT-4 appears to have a higher value for mean number of penalty points: more than 1SE above the estimated mean value of the original Vicuna. Gudibande et al. (2023) also noted that fine-tuning sometimes could worsen performance.

The Answer-Explain setup has a slightly lower mean than the original Vicuna, but cannot be assumed to have an improved performance. In this setup, I rely on a yes/no answer in the critique to be able to decide if the statement should be revised or not. Vicuna sometimes fails to provide this format, as exemplified in Figure 5.4. When this happens, I have chosen to interpret this as a principle not being broken, to not risk sending an explanation of why a principle is satisfied into revision. The Vicuna tuned on explanation-answer data (including the word yes or no), has fewer instances of producing answers in an unwanted format. But from my observations, the quality of the explanations themselves seems similar to those of the original Vicuna. This also coincides with observations from Gudibande et al. (2023), as tuning Vicuna here has led to an improvement in the style of the answer, but maybe not the factual quality. The improved style might slightly improve the overall performance of the configuration, but without better quality in the explanation, this is a possible explanation for why this configuration did not achieve results more similar to GPT-4 on Answer-Explain.

Overall, I note that **all average numbers of penalties ($\pm$1 SE) are above 1 and below 4** (see Table 5.1). Out of 13 possible penalties, this is not very high, but it is also not very close to 0. To do further analysis, I have picked out a subset of the configurations to perform more inference and evaluation on. These configurations are those marked in bold in Table 5.1. The configurations are chosen because I want to do further comparison of how the methods have or have not improved on the baselines: Initial and `DR`. I also want to further explore the configuration that appears to be the best one overall: GPT-4 on Answer-Explain. In addition, I want to compare the different `Critiquers` on the same setup. For this, I have chosen the setup where using GPT-4 and an untuned Vicuna had the closest number of average penalties,

```
Critique request: Does the statement miss or add important parts of explaining the offense (and the
defendant's involvement in it), compared to the input story?
 Start by thinking about the question step by step, and include relevant quotes from the input story
 and from the statement. Then always end by answering the question concisely with <yes> or <no> being
 the last word.
Critique:  <The statement adds important parts of explaining the offense and the defendant's
involvement in it, compared to the input story. It provides more context about the bribe, the
company's greenwashing, and the impact of the defendant's actions. The statement also highlights the
defendant's remorse, the steps taken to make amends, and the lessons learned from the experience.>
```

```
Critique request: Does the statement contain any misplaced information for a plea
in mitigation (eg mentioning the 'input story' or another statement)?
 Start by thinking about the question step by step, and include relevant quotes
 from the input story and from the statement. Then always end by answering the
 question concisely with <yes> or <no> being the last word.
Critique:  <Based on the input story and the statement, there is no misplaced
information in the statement for a plea in mitigation. The statement addresses the
offense, accepts responsibility, and provides context for the offender's actions.
It also highlights the offender's previous good conduct, community involvement,
and commitment to making amends. The statement effectively conveys the offender's
remorse and willingness to learn from the experience.>
```

Fig. 5.4 Two examples of Vicuna producing answers in an unwanted format in the Answer-Explain setup (not starting or ending with yes or no).

as this is where I had most hope of Vicuna being as good as or possibly better than GPT-4: this is at the Answer setup. To further investigate this, I will consider all four configurations for the Answer-setup.

5.3 Further analysis

Now considering my seven chosen configurations, I have evaluated these on 50 samples in Table 5.2 and 5.3. In Table 5.2, each column represents a configuration (combination of setup and `Critiquer`). In particular, it shows my two baseline setups, and the best-performing configuration: GPT-4 `Critiquer` using Answer-Explain. Each row in the table corresponds to one input story, except for the bottom row which shows statistics over all input stories combined. In addition to looking at the updated estimated mean and SE of the configurations, I look at statistics for each input story. In particular, I look at the mean number of penalties and the difference between the highest and lowest number of penalties scored for each input story in each setup.

From Table 5.2, I note that the best configuration achieves an average number of penalty points of 1.56, and the worst performing configuration has an average of 3.38.

Table 5.2 Shows the two baseline setups: Initial and `Direct Refinement`, compared to what appeared to be the best performing configuration in Table 5.1: GPT-4 and Answer-Explain. Each configuration is evaluated over 50 samples: five times for each input story. The table shows the **mean** number of penalties given to each input story for each setup, and the **diff**erence between the highest and lowest scoring penalties for that input story and configuration. The bottom part shows the **mean** number of penalties and **SE** over all input stories, for that configuration.

Input story	Initial_{50}		$\texttt{DR}_{50}$		$\text{GPT}-4\ \text{AE}_{50}$	
	mean	diff	mean	diff	mean	diff
Jack	5.8	2	2.8	5	1.0	3
Charlotte	4.0	2	2.8	3	1.8	2
Robert	3.8	6	3.0	2	2.0	5
Aleksander	3.8	4	2.8	2	2.0	2
David	3.8	1	3.2	1	0.4	1
Paul	3.4	3	3.0	2	2.0	2
Olivia	3.2	5	2.4	4	1.0	4
Melissa	3.0	6	1.6	2	2.2	4
Lauren	2.2	5	2.0	3	2.2	3
Chloe	0.8	3	2.4	3	1.0	3
	mean	SE	mean	SE	mean	SE
	3.38	0.27	2.6	0.17	1.56	0.20

Looking at the Initial column, I see that some input stories seem to be more difficult to make initial statements for than others. The average number of penalties given to each story's initial statement ranges from 5.8 to 0.8, as an average of five on each story. This difference between difficult/easy stories causes uncertainty around the mean value, especially in the Initial setup. Looking at the difference between what seems to be the most difficult input story: Jack Palmer, and the easiest input story: Chloe Mills, there are some clear differences (all input stories are provided in Appendix A.1.1). Jack's story does not include many mitigating factors, and even though he has a clear record, the factor is complicated by a speeding ticket. Chloe's story, on the other hand, includes several mitigating factors, as well as a longer description of the crime. This indicates that it's easier to produce good statements for input stories that provide more information and more mitigating factors. By qualitative inspection, this appears to be the case because without much information, the `Writer` tends to invent untrue mitigating factors, resulting in multiple broken principles. There is also variance in how many penalties are given to the same story, using the same setup. This also causes a higher SE, and shows that the Initial setup does not reliably produce statements of any

quality.
Secondly I look at the DR setup. Here, the highest average penalty for a story is 3.2: lower than for the initial setup. There seems to be less separation between "difficult" and "easy" input stories, resulting in a lower SE than for the initial setup. However, there is still variation between the number of penalties given the final statement for each run on the same input story, although this variation appears to be slightly lower than in the Initial setup. Combined, this causes the overall SE for DR to be much lower than for the Initial setup. The mean number of penalties per statement is also lower, providing more confidence to the hypothesis that the DR setup provides better statements than the initial setup. Hence, the Revision Requests (RRs) appear to be a helpful part of the setup.
Third, I look at using GPT-4 as a `Critiquer` in the Answer-Explain setup. Notice that the number of penalties is reduced to a maximum of 2.2 for all input stories. This indicates that this configuration does equally well on most input stories, independent of how “difficult” they may be. There is still variance in the number of penalties given to each story using the same setup, varying at most with 5 penalty points for Robert’s input story. So even though this configuration on average provides relatively good statements, it is not 100% reliable. **GPT-4 using Answer-Explain achieves an average of 1.56 penalty points per statement**, and a relatively low SE of 0.2, meaning the true average is quite likely to be below 2. This also means that the average number of penalties for GPT-4 on Answer-explain is further than 5 Standard Errors away from the DR setup, indicating that this is a significant improvement. Overall, it seems like the DR setup improves the initial statements, but the GPT-4 Answer-Explain configuration improves the initial statements the most. After evaluating 50 samples on these three configurations, they are all separated by far more than ± 1SE, which implies that **the methodology used indeed makes a difference on the final statements**.

Table 5.3 shows the evaluation done on the remaining four setups I decided to further investigate. The columns represent different `Critiquers` used in the Answer setup, and each row corresponds to one input story (except for the bottom row which shows statistics over all input stories combined). Also in this Table, I consider the metrics of mean number of penalty points per input story and for each configuration overall; the Standard Error for each configuration; and the difference in penalty points between the highest and lowest scoring run for each input story in each configuration. The table shows that GPT-4 achieves the best performance in the Answer setup, with a mean of 2.2 penalty points. The fine-tuned Vicuna models have the worst mean values at 3.14 and 3.16 penalty points per statement.

First, I look at using GPT-4 as the `Critiquer` in the answer setup. I note that this configuration seems to do better at the same input stories that the Initial setup seemed to think

Table 5.3 Comparing different `Critiquer` models in the Answer-setup. All configurations have been evaluated on 50 samples: five samples for each input story. As Table 5.2, I show the mean number of penalties given to statements for each input story for each configuration and the difference between the highest and lowest penalties given to that input story and configuration. The bottom line shows the updated mean number of penalties, and the SE, for each configuration over all input stories.

Input story	$GPT-4$ A_{50}		Vicuna A_{50}		Vicuna $A_{mine\,50}$		Vicuna $A_{GPT-4\,50}$	
	mean	diff	mean	diff	mean	diff	mean	diff
Jack	3.6	3	4.2	3	2.8	2	5.8	3
Charlotte	2.6	3	2.4	4	3.6	3	3.4	3
Robert	3.2	4	3.2	6	4.2	6	3.2	5
Aleksander	3.4	1	3.6	2	4.4	3	1.8	3
David	0.8	1	2.2	5	2.8	6	2.6	3
Paul	4.2	4	3.6	2	2.2	3	4.6	3
Olivia	1.0	2	3.4	6	4.0	5	2.0	4
Melissa	0.0	0	3.2	3	4.4	4	2.6	5
Lauren	1.0	2	1.4	3	0.8	2	2.2	4
Chloe	1.4	4	3.4	4	2.2	5	3.4	2
	mean	SE	mean	SE	mean	SE	mean	SE
	2.20	0.33	3.06	0.24	3.14	0.27	3.16	0.25

were "easy". One reason for this might be that the GPT-4 `Critiquer` in this Answer-setup does very little to no revision of the initial statements. On average, this GPT-4 configuration answers that a principle is broken 1.44 times during one loop through all principles. With so few revisions, it is not a surprise that the results are quite similar. In contrast, GPT-4 using the Answer-Explain setup rewrites a statement on average 3.86 times in one loop through all statements. Despite the low number of revisions, GPT-4 on this Answer setup achieves a mean number of penalties of 2.2. It is more than 1SE better than the `DR` setup, which revises 13 times. This indicates that there is a lot of value in the explanations given to the `Writer` before revision, and that more revisions do not always result in a better final statement. The configuration gets a quite high SE, as a result of variance in penalties both between input stories and between runs of the same input story.
Second, I look at Vicuna and it's fine-tuned versions. The mean number of penalties is very similar for the original and fine-tuned versions. One possible explanation for this is that the `Critiquer` component in the Answer setup is not very significant. The difference between using GPT-4 versus the original Vicuna is much smaller here compared to the Answer-Explain setup, implying that the critique in the Answer setup might not be of as much significance. Because of this, even though the yes/no answers in the fine-tuned versions of Vicuna might be better than the original, this will not automatically result in a lower average number of penalty points. However, some of this smaller gap between the `Critiquers` might also be because GPT-4 benefits from Chain-Of-Thought (COT) prompting in the Answer-Explain setup, and not in the Answer setup. A smaller model like Vicuna13 does likely not benefit from COT prompting at any point (Kojima et al., 2022). But the fine-tuned models have learned something, as we again can see from Figure 5.1. Part of what it has learnt is the form of the desired output: only "yes" or "no". The original Vicuna model often answers yes or no first, but continues to try to add more words after that, as seen in Figure 5.5. The fine-tuned models don't do much of this: they almost always answer only yes or no, as seen in the training data. This again relates to the claim by Gudibande et al. (2023) that small models like Vicuna don't necessarily learn factual accuracy from fine-tuning, but more so the desired style of the output. Perhaps this is also related to the difficulty of the task to be done, which even state-of-the-art GPT-4 is not perfect on.

Overall, none of these Answer configurations are close to the performance of GPT-4 using Answer-Explain. The Answer-setup configurations also show variance between runs using the same configuration and input story, but generally achieve higher penalties from the automatic evaluation (Section 4.3).

```
Critique request: Does the statement break the rule 'only mention the
defendant's age if it is outside of the interval 30-79'?
 Please answer only 'yes' or 'no'.
Critique:  <No, the statement does not break the rule of>
The statement satisfies the CR!

Critique request: Does the statement miss or add important parts of
explaining the offense (and the defendant's involvement in it), compared
to the input story?
 Please answer only 'yes' or 'no'.
Critique:  <No>
The statement satisfies the CR!

Critique request: Does the statement contain any misplaced information
for a plea in mitigation (eg mentioning the 'input story' or another
statement)?
 Please answer only 'yes' or 'no'.
Critique:  <No, the statement does not contain any mispla>
The statement satisfies the CR!

Critique request: Is there any information in the statement that is
unsupported by the input story?
 Please answer only 'yes' or 'no'.
Critique:  <No, there is no information in the statement that>
The statement satisfies the CR!
```

Fig. 5.5 This is an example showing that Vicuna often answers more than only "yes" or "no" as a critique in the Answer-setup.

To better visualize the results in this section, the average number of penalties given to the two baselines (Initial, `DR`), the best configuration (GPT-4 using Answer-Explain), and the best configuration using Vicuna (Answer, non-tuned) are shown in Figure 5.6. Looking at this Figure, it is clear to see that GPT-4 Answer-Explain is the superior setup on most input stories, and performs quite evenly over the input stories. Vicuna Answer and `DR` also usually improve on the initial statements, but the improvement is not as clear for Vicuna Answer.

Another general observation worth mentioning, is the faults of the `Writer`. In this work, I have focused on improving the `Critiquer` of the methods, but in doing so, I have also observed the performance of the `Writer`. One challenge I had, was getting the `Writer` to do revision only where it was told to revise. Sometimes, the revision would change a part of a statement that did not need to change, resulting in breaking a principle that was not initially broken. Another challenge was to get good explanations from the Explanation Requests (`ER`s). From time to time, the explanations could contradict the decision of the

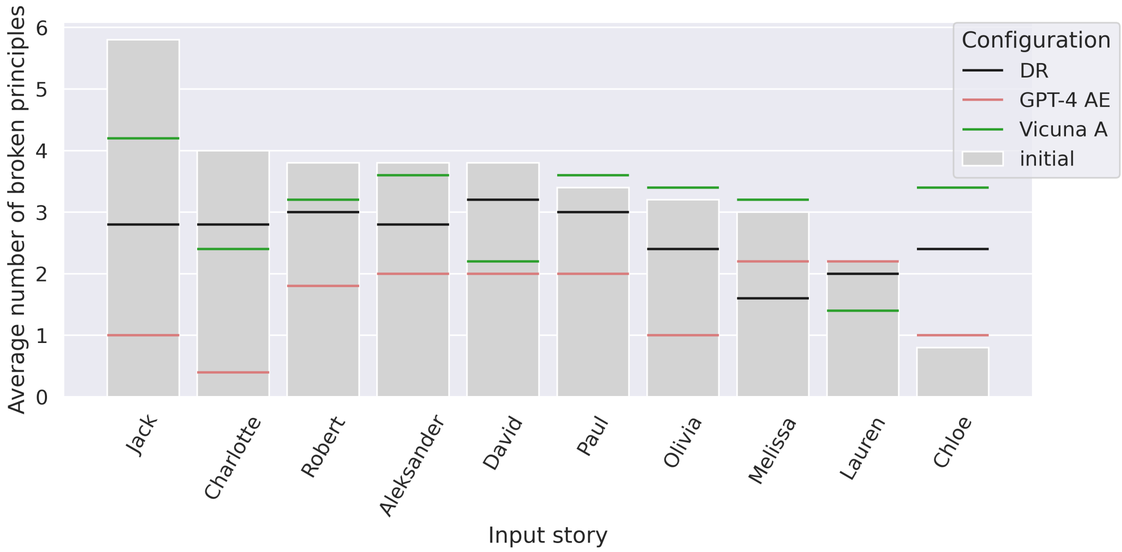


Fig. 5.6 Average number of penalties given to different input stories (on the x-axis) and a selection of configurations (colors). Most configurations improve on the Initial setup.

`Critiquer` that a principle is broken, and instead argue for why it is not broken. Other times, the explanations did not correctly identify what caused a principle to be broken, or claim that something was stated in the input story or statement, even if that was not true. Tuning my input prompts (Section 4.2) helped in reducing the size of these issues, but they also occurred sometimes after prompt tuning.

This chapter has shown results of evaluating the setups introduced in Section 3.2. Generally, it has shown that GPT-4 appears to be the best model for the `Critiquer` in all configurations I tried. The fine-tuning of Vicuna mostly led to better models in some respect (style of output), but this did not always result in lower penalty scores. Most configurations improved on the initial statement, and GPT-4 in the Answer-Explain setup showed the most improvement.

Chapter 6

Discussion and Future work

In this chapter I will further discuss some of the results from Section 5.3 by looking at it from a higher level. I will then discuss the potential of further research on the task of writing a plea in mitigation. Lastly, I will give a brief conclusion on this work.

The best configuration in Section 5 achieved an average of 1.56 penalty points for a final statement. Even though this is a big improvement compared to some baselines, it is still not below 1, which could have possibly been considered low enough to be reliable in this setting. Even the best configuration achieved a quite varied number of penalty points from the same input, further adding to the conclusion that **the best method is still not reliable enough to deploy**.

Although I cannot conclude that these methods are reliable yet, they show promise that they can become reliable in the future. One thing is that using a different `Writer` to explain, revise and produce an initial statement might have led to improved overall performance. Perhaps for example GPT-4 would be able to better perform these tasks. Another thing that might positively influence this method is trying more different setups. For example, I did not try to do multiple loops of each configuration or to loop until total satisfaction. It is also worth considering that new LLMs are released at a rapid pace, and a new model might have qualities more suitable to this task than the models I have used here. In addition, as with most data-driven problems these days, the results might have been better if there existed more high-quality data, or just more data in general. Then, there is a chance that the fine-tuning of Vicuna would have given better results.

I have used the autoevaluation method described in Section 4.3 to attain the results. Because this evaluation is not 100% accurate, my results are not 100% accurate either. If results at some point seem good enough to deploy, they should first be evaluated in a better way, and in cooperation with legal professionals.

There are however multiple things to consider before something like this could be deployed. We see a chatbot as a possible destination for this task, where a user can input their details, and the chatbot can suggest a reliable plea in mitigation based on the user's input. I have assumed that my input stories contain all relevant information about a defendant and their case. However, in a conversation with a chatbot, someone who is not familiar with the law is unlikely to know what sort of information they need to include. The chatbot would therefore need to do knowledge elicitation in order to find out what information is possibly missing from the input, and ask the user to provide this input before suggesting a statement. If such a chatbot is released, it has the potential to help many people get access to potentially free legal aid, but there are probably also other consequences of releasing such a chatbot, so there would need to be a rigorous ethical evaluation before ever deploying it.

In this work, I have looked at the task of writing a statement for a plea in mitigation. I have defined a set of principles that a statement should satisfy, and introduced several new methodologies explaining how one might achieve a statement satisfying such principles. I have introduced a new dataset containing statements, input stories and labels connected to different principles, and made an automatic evaluation strategy of statements. I have also evaluated 16 different configurations on the task, and discussed why some configurations perform better than others. Fine-tuning a smaller model did not work as well as using a pre-trained larger LM. The best-performing configuration used GPT-4 to explain why a statement was breaking or not breaking a specific principle. I achieved an average on 1.56 penalty points per statement using this configuration, but I didn't consider this to be good enough, or reliable enough for deployment. Finally, I have discussed how one can build on this work, doing further research on this task.

References

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning, 2020.

Afra Feyza Akyürek, Ekin Akyürek, Aman Madaan, Ashwin Kalyan, Peter Clark, Derry Wijaya, and Niket Tandon. Rl4f: Generating natural language feedback with reinforcement learning for repairing model outputs. *arXiv preprint arXiv:2305.08844*, 2023.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. Palm 2 technical report, 2023.

Anthropic. Model card and evaluations for claude models. 2023. Available at: https://www-files.anthropic.com/production/images/Model-Card-Claude-2.pdf (Accessed: August 6 2023).

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*, 2018.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

Doughty Street Chambers. Criminal appeals bulletin: Sentencing – advanced age of the offender as a mitigating factor. Available at: https://doughty-street-chambers.newsweaver.com/Appeals/wstmt18w3ba?a=1&p=1623589&t=174031 (Accessed: June 28 2023).

GOV.UK. Criminal courts. a. Available at: https://www.gov.uk/courts.

GOV.UK. b. Available at: https://www.gov.uk/represent-yourself-in-court (Accessed: 4 August 2023).

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms, 2023.

Rod Hayler. Representing yourself in the magistrates' court. *Old Bailey Solicitors*. Available at: https://www.oblaw.co.uk/how-to-represent-yourself-in-the-magistrates-court-2/ (Accessed: 23 July 2023).

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Frederick Jelinek. *Statistical methods for speech recognition*. MIT press, 1998.

Judiciary of England and Wales. How are sentences decided? July 2020. Available at: https://www.judiciary.uk/wp-content/uploads/2020/07/sentencing-guide-accessibilty-check-July-2020-1.pdf (Accessed: 23 July 2023).

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Chin-Yew Lin and Franz Och. Looking for a few good metrics: Rouge and its evaluation. 2004. URL https://api.semanticscholar.org/CorpusID:55156862.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. Reframing instructional prompts to gptk's language. *arXiv preprint arXiv:2109.07830*, 2021.

Motor Lawyers Ltd. Plea of mitigation. Available at: https://www.singlejusticeprocedure.co.uk/plea-mitigation.html (Accessed: 23 July 2023).

OpenAI. Pricing. Available at: https://openai.com/pricing (Accessed: 23 July 2023).

OpenAI. Gpt-4 technical report, 2023.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Kevin Roose. The brilliance and weirdness of chatgpt. *The New York Times*, 5 December 2022. Available at: https://www.nytimes.com/2022/12/05/technology/chatgpt-ai-twitter.html (Accessed: July 31 2023).

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022.

Sentencing Council. Aggravating and mitigating factors. a. Available at: https://www.sentencingcouncil.org.uk/explanatory-material/magistrates-court/item/aggravating-and-mitigating-factors/ (Accessed: 23 July 2023).

Sentencing Council. Common offences. b. Availabel at: https://www.sentencingcouncil.org.uk/outlines/ (Accessed: 9 August 2023).

Sentencing Council. Corporate offenders: fraud, bribery and money laundering. 2014. Available at: https://www.sentencingcouncil.org.uk/offences/magistrates-court/item/corporate-offenders-fraud-bribery-and-money-laundering/ (Accessed: 23 July 2023).

Sentencing Council. Reduction in sentence for a guilty plea - first hearing on or after 1 june 2017. 2017. Available at: https://www.sentencingcouncil.org.uk/overarching-guides/magistrates-court/item/reduction-in-sentence-for-a-guilty-plea-first-hearing-on-or-after-1-june-2017/ (Accessed: 23 July 2023).

Sentencing Council. General guideline: overarching principles. 2019. Available at: https://www.sentencingcouncil.org.uk/overarching-guides/magistrates-court/item/general-guideline-overarching-principles/ (Accessed: 23 July 2023).

Jessica Shieh. Best practices for prompt engineering with openai api. *OpenAI*, 2023. Available at: https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-openai-api (Accessed: August 15 2023).

Ronnie W Smith and D Richard Hipp. *Spoken natural language dialog systems: A practical approach*. Oxford University Press, USA, 1994.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Sam Tonkin. What is chatgpt? everything you need to know about the new ai chatbot that garnered more than one million users in its first week thanks to its eerily human-like responses. *The New York Times*, 9 December 2022. Available at: https://www.dailymail.co.uk/sciencetech/article-11521261/What-ChatGPT-need-know-Elon-Musks-new-AI-chatbot.html (Accessed: July 31 2023).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

United Nations. Youth. Available at: https://www.un.org/en/global-issues/youth (Accessed: June 28 2023).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Eric J. Wang. Alpaca-lora. https://github.com/tloen/alpaca-lora, 2023. (Accessed: 15 July 2023).

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

Jieyu Zhao, Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Kai-Wei Chang. Ethical-advice taker: Do language models understand natural language interventions? *arXiv preprint arXiv:2106.01465*, 2021.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

Appendix A

A.1 Data and Prompts

A.1.1 Input stories

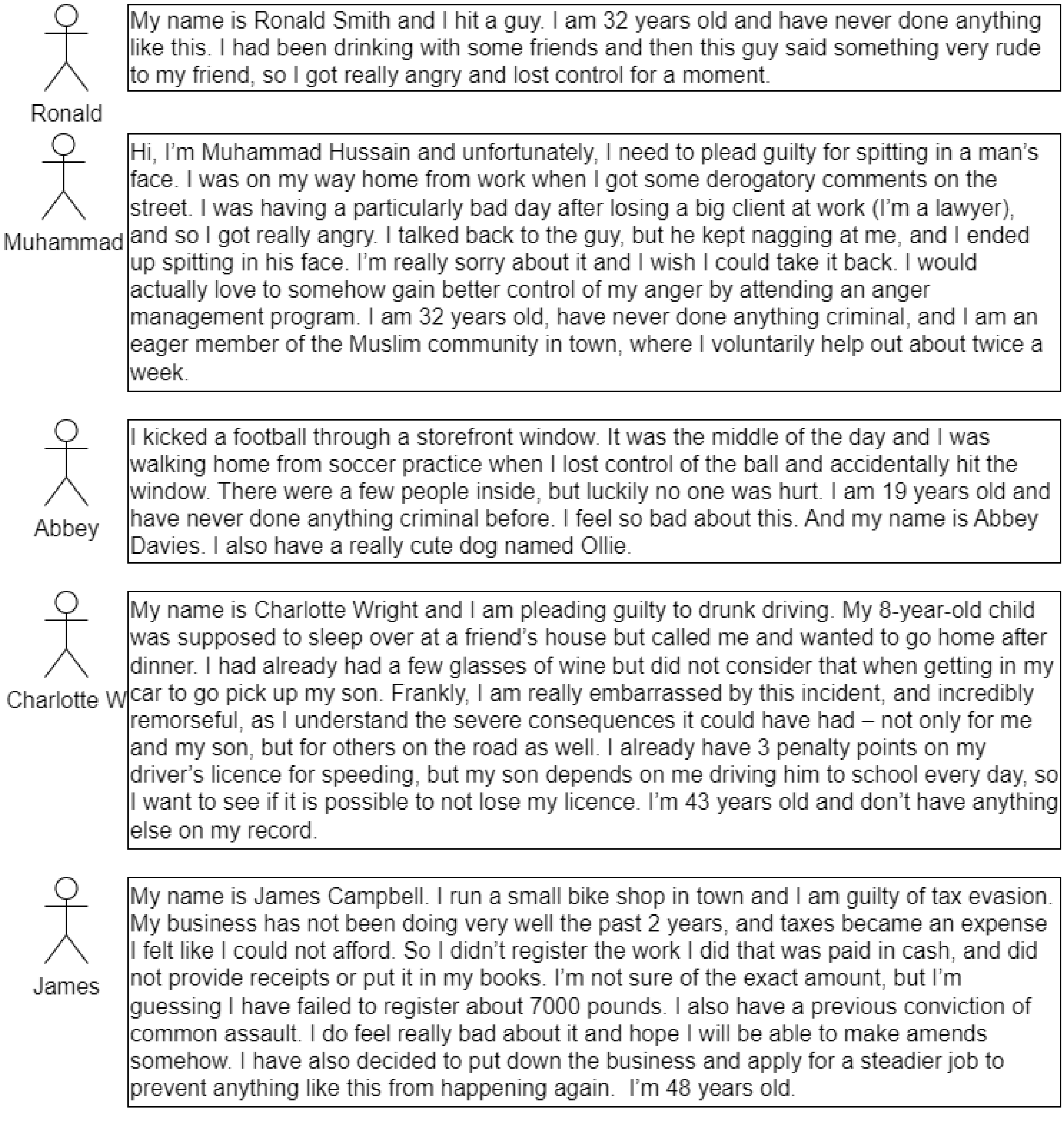


Fig. A.1 The five input stories used to generate training data.

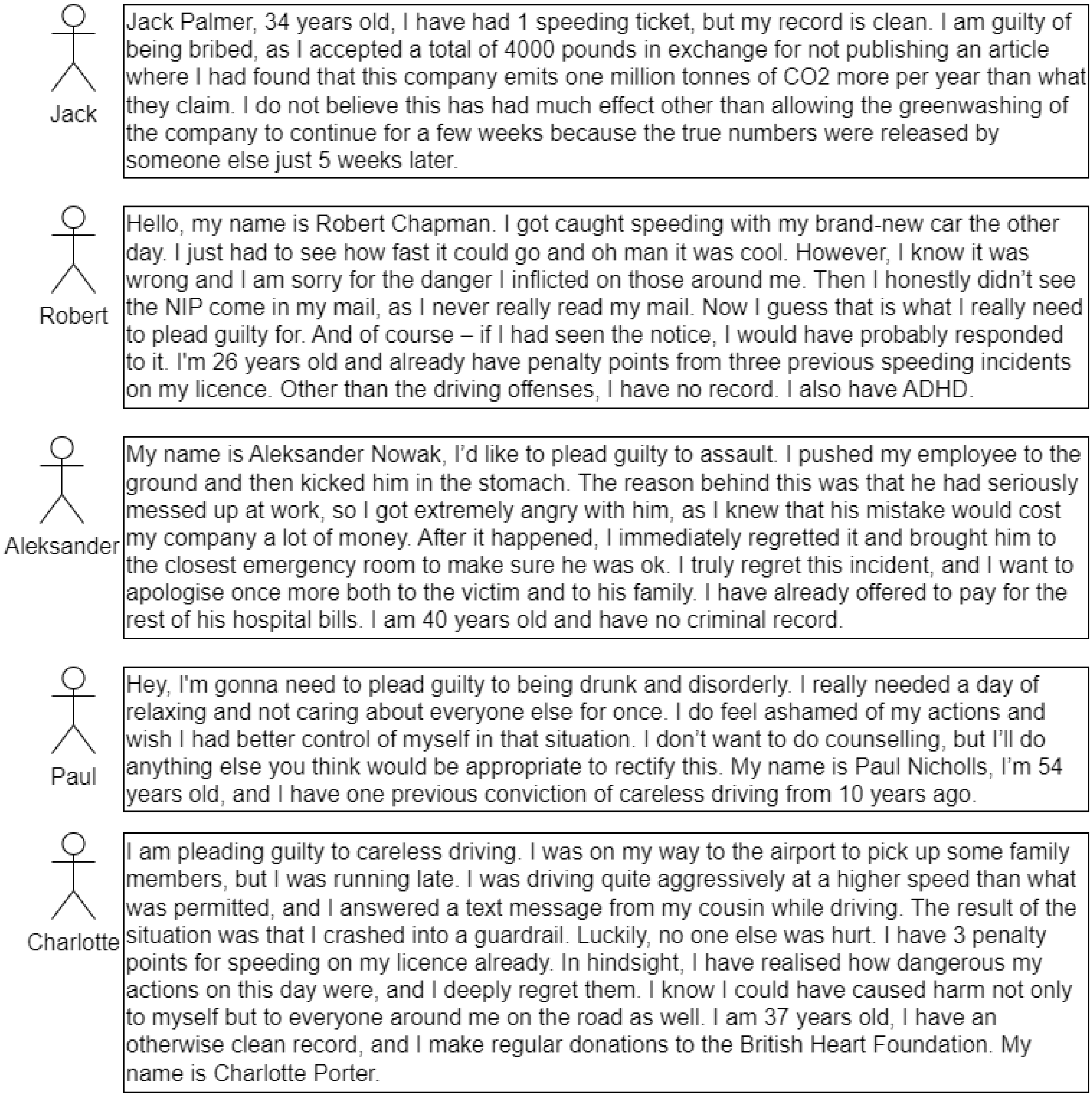


Fig. A.2 First five input stories used for evaluation.

Fig. A.3 Last five input stories used for evaluation.

A.1.2 Example statements

Input story

My name is Ronald Smith and I hit a guy. I am 32 years old and have never done anything like this. I had been drinking with some friends and then this guy said something very rude to my friend, so I got really angry and lost control for a moment.

Vicuna13 Statement

Your honor, I am Ronald Smith and I am here today to express my deep regret for my actions. I understand that what I did was wrong and I take full responsibility for my actions. On the day in question, I had been out with some friends and had been drinking. I am not normally a violent person and I have never been in trouble with the law before. However, on that day, something happened that made me lose control. The man who I hit had said something very rude to my friend and I was extremely angry. In the heat of the moment, I acted in a way that I deeply regret. I understand that what I did was not acceptable and I am truly sorry for my actions. I have never been in a situation like this before and I am aware that what I did was a crime. I have always tried to be a law-abiding citizen and I am deeply ashamed of what I did. I have been working with the authorities and I am willing to do whatever it takes to make amends for my actions.

GPT-3.5 Turbo Statement

Your Honor, I, Ronald Smith, plead guilty to the offense of assaulting another individual. I understand the seriousness of my actions and take full responsibility for them. On the night in question, I had been socializing and consuming alcohol with some friends. Regrettably, a situation arose where an individual directed a rude comment towards one of my close friends. The comment triggered a sudden surge of anger and I momentarily lost control of my emotions, resulting in me physically striking the individual. While I am deeply remorseful for my actions, it is important to acknowledge that this incident was an isolated event in my life. At 32 years old, I have never before engaged in any form of violent behavior or confronted such a situation. My lack of a criminal record stands as a testament to my otherwise law-abiding character. I recognize that my conduct was unacceptable and I understand the negative impact it had on the victim. I am committed to making amends for my actions and prevent any recurrence of such behavior. I have taken the initiative to seek counseling and anger management courses, which will aid me in developing healthier coping mechanisms and prevent similar situations from arising in the future. Furthermore, I would like to emphasize that I have already experienced significant personal consequences as a result of my crime. Coming to terms with the reality of my violent behavior has brought me immense sadness and regret. I have reflected deeply on the incident and the harm that I caused, and I am determined to learn from this mistake and emerge as a better person. I am truly sorry for the pain and distress I have caused, both to the victim and to society as a whole. I assure the court that this incident does not reflect my true character or the values that I hold. I have always aimed to be a law-abiding and contributing member of society, as evidenced by my actions thus far. In conclusion, I respectfully request the court's consideration of the circumstances surrounding this incident and my genuine remorse for my actions. I am committed to making the necessary changes to prevent any repetition of such behavior. I trust that the court will take into account my guilty plea, lack of criminal history, and the steps I am taking towards self-improvement. Thank you for your attention in this matter.

Fig. A.4 A textual example showing why I chose to use GPT-3.5 Turbo (blue statement) as a general `Writer`, and not Vicuna13 (green answer). The statement produced with GPT-3.5 Turbo has a better flow of content and is generally written in a better manner.

Chloe

Your Honor, I stand before this court today as a 24-year-old individual named Chloe Mills. I humbly plead guilty to the offense of affray. I want to express my deep remorse and take full responsibility for my actions. In recounting the events, I found out that my boyfriend had cheated on me. Overwhelmed by emotions, I impulsively went to the bar where he was with his friends. My intention was to confront him and seek answers about the other girl involved. I regrettably resorted to making threats, mentioning a gun, in an attempt to make him disclose the truth. It is important to note that I do not possess, nor have access to, any firearm. In the midst of my anger and hurt, I did slap him, which I now regret. I acknowledge that my behavior was inappropriate and alarming to the other patrons of the bar. I understand the impact it had on them, and I deeply apologize for causing fear and distress within the establishment. I must emphasize that during the entirety of my presence at the bar, I only engaged in verbal confrontations for approximately 15 minutes before choosing to leave. I stand before you as an individual who has never been involved in any criminal activity before this incident. Although this does not excuse my actions, I believe it serves as a testament to the fact that I am not a habitual offender. Additionally, I have been actively attending therapy sessions once a week to work on myself. This demonstrates my commitment to personal growth and my willingness to address any underlying issues that may have contributed to my behavior. Your Honor, I plead guilty to the offense and sincerely apologize for my actions. I believe that taking responsibility for my actions is in itself a significant mitigating factor. I regret the alarm I caused in the bar, but I want to assure the court that I have learned from this experience. Thank you for granting me the opportunity to address the court today.

Penalty points assigned: 0

Fig. A.5 An example of a statement receiving 0 penalty points from the auto-evaluation. Based on the Chloe input story.

Lauren

Your Honor, I, Lauren Dixon, stand before you today to plead guilty to the charge of TV licence payment evasion. I would like to express my sincere remorse for my actions and take full responsibility for my mistake. I understand the seriousness of this offense and the impact it has had on the legal system. At the time of the incident, I genuinely believed that the bill for my TV licence had been taken care of. However, upon realizing my mistake, I immediately rectified the situation by making the necessary payment. I want to emphasize that there was no intention to evade payment for the TV licence. It was a genuine oversight on my part. I have always strived to be a law-abiding citizen, and I deeply regret the consequences of my lapse in judgment. I have no previous criminal record, which I hope can be seen as a reflection of my character and dedication to upholding the law. This incident has served as a significant wake-up call for me, and I am committed to ensuring that it will never happen again. I am fully prepared to do whatever it takes to rectify the situation and prevent any future offenses of this nature. I am willing to comply with any measures imposed by the court and demonstrate my commitment to being a responsible citizen. In conclusion, I sincerely apologize for any inconvenience and oversight caused by my failure to pay for the TV license. I humbly request the court's leniency and understanding in considering the circumstances surrounding my case. Thank you for your attention, Your Honor.

Penalty points assigned: 3

Fig. A.6 An example of a statement receiving 3 penalty points from the auto-evaluation. Based on the Lauren input story. The blue highlights indicate which principles were considered broken here.

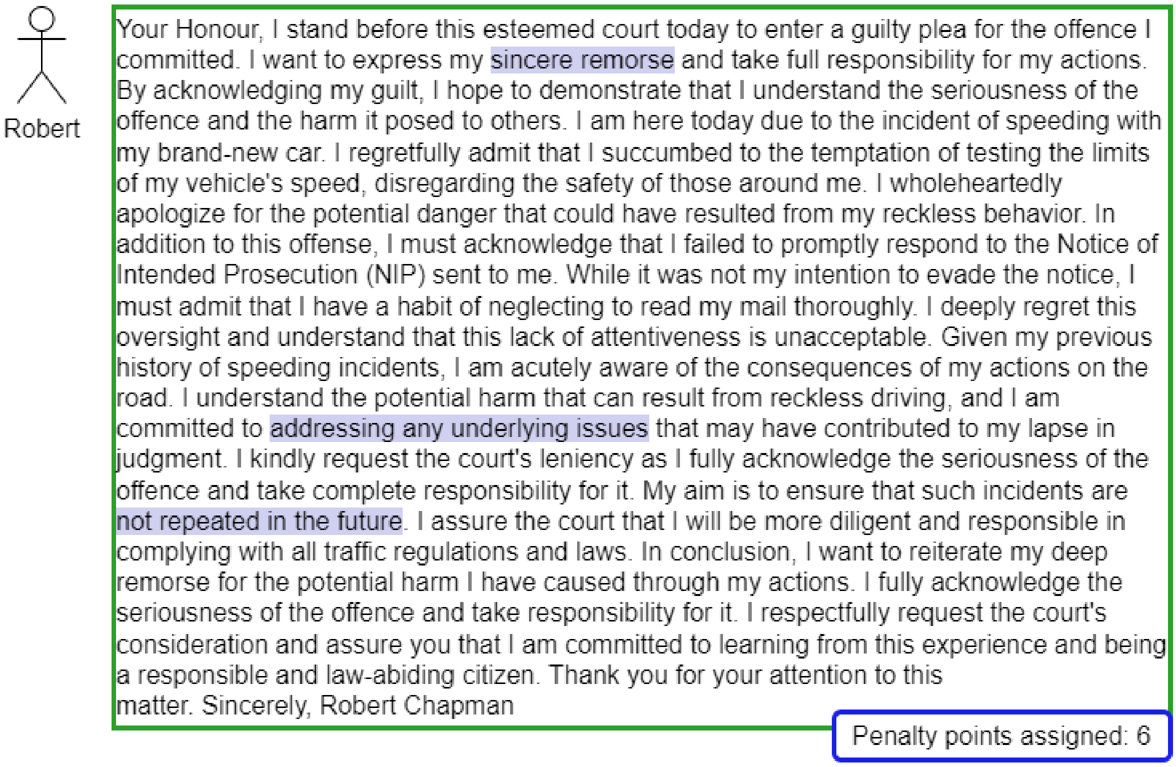


Fig. A.7 An example of a statement receiving 6 penalty points from the auto-evaluation. Based on the Robert input story. The blue highlights indicate which principles were considered broken here. The additional penalty points are due to information about the defendant that the statement failed to mention.

A.1.3 Input to LLMs

Critique Requests for GPT-4
Is anything in the statement badly **phrased**, difficult to follow, or repetitive?
Does the statement ignore the importance of the defendant's **guilty plea** (an important mitigating factor)?
Does the statement give a different impression (compared to the input story) of whether the defendant has already **suffered** as a consequence of their crime? If great emotional suffering is mentioned in the statement, this should automatically trigger a 'yes'.
If the statement mentions the defendant's **age**: is that age either in the interval 30-79, or is the age different to the age mentioned in the input story? If the defendant's age is not mentioned in the statement: is the defendant's age less than 30 or older than 79 according to the input story?
Does the statement and input story seem to disagree on whether the defendant has a prior **criminal record**?
If the input story does not mention any **mental** illness or disability: does the statement mention any mental illness or disability? If the input story does mention some mental illness or disability: does the statement fail to mention that illness or disability?
Does the statement miss or add important parts of **explaining** the offense (how it happened and the defendant's involvement in it), compared to the input story?
If the input story mentions that the defendant has or wants to make **amends**, does the statement fail to mention this? Or does the statement introduce a desire for the defendant to make amends, that is not mentioned in the input story?
If the input story mentions that the defendant has or wants to take action to self-**improve or prevent** the offense from happening again, does the statement fail to mention this? Does the statement introduce a desire to self-improve or prevent further offenses that is not present in the input story? Self-improvement can for example be attending counselling.
If the input story mentions some information that contributes to the defendant's good **character**, does the statement fail to mention this? Or does the statement introduce information about the defendant's character that is not present in the input story?
Does the statement contain any **misplaced information** for a plea in mitigation (eg mentioning the 'input story' or another statement)?
Does the statement overemphasize, or show **emotions** different from the input story?
Is there any information in the statement that is **unsupported** by the input story?

Fig. A.8 Critique Requests passed into the GPT-4 `Critiquer`. Bold words in this image highlight what the `CR` concerns

Critique Requests for Vicuna13
Is anything in the statement **badly phrased**, difficult to follow, or repetitive?
Does the statement ignore the importance of the defendant's **guilty plea** (an important mitigating factor)?
Does the statement give a different impression (compared to the input story) of whether the defendant has already **suffered** as a consequence of their crime? Say 'yes' if the statement talks about great emotional suffering.
Does the statement break the rule 'only mention the defendant's **age** if it is outside of the interval 30-79'?
Does the statement and input story seem to disagree on whether the defendant has a prior **criminal record**?
Does the statement fail to mention any **mental** illness or disability (if the defendant has this)?
Does the statement miss or add important parts of **explaining** the offense (and the defendant's involvement in it), compared to the input story?
Does the statement give a different impression than the input story of whether the defendant has or want to make **amends** or 'make up' for the crime?
Does the statement give a different impression than the input story of whether the defendant has or wants to **self-improve** (eg counselling) or **prevent** the offense from happening again?
Does the statement give a different impression than the input story of the defendant's good **character** (for example contributing to the community in some way)?
Does the statement contain any **misplaced information** for a plea in mitigation (eg mentioning the 'input story' or another statement)?
Does the statement overemphasize, or show **emotions** different from the input story?
Is there any information in the statement that is **unsupported** by the input story?

Fig. A.9 Critique Requests passed into the Vicuna `Critiquers`.

Explanation Requests
The statement lacks choerence or contains bits that are not well-written. Where? What can be done to make the statement better written and coherent?
The statement does not properly emphasize that the defendant's guilty plea as a mitigating factor that benefits the court. Explain why, and include quotes from the statement that should be improved to properly emphasize that the guilty plea is an important mitigating factor
Does the input story mention anything the defendant has suffered (non-emotionally) as a consequence of their crime? Does the statement mention any suffering as a consequence of the crime? The principle concerning suffering IS broken - explain why (the discrepancy between input story and statement).
How old is the defendant according to the input story? Does the statement say how old the defendant is? Answer this part separately first. The rule 'only state the defendant's age if it is outside of the interval 30-79' IS broken; explain why.
Does the input story imply that the defendant has a criminal record? Does the statement imply if the defendant has a criminal record? The principle concerning a criminal record IS broken - explain why.
Does the input story say that the defendant is suffering from a mental illness or disability? Does the statement imply that the defendant has mental issues? The principle about mental illness IS broken - explain why (the discrepancy between input story and statement).
What does the input story say about the offense and the defendant's involvement? Does the statement give any additional information, or does it not mention parts of the offense from the input story? The principle about explaining the offense and the defendant's involvement in it IS broken - explain why (the discrepancy between input story and statement).
Does the input story mention any intent to make amends? Does the statement mention wanting to make amends? The principle concerning amends IS broken - explain why (the discrepancy between input story and statement). If there is no implication of making amends in the input story, the statement should not imply this either. If the input story mentions some action the defendant has done or wants to do to make amends, that should be included in the statement. Amends can for example be paying restitution.
Does the input story mention any intent for self-improvement or ensuring no further criminal activity? Does the statement imply that the defendant has or wants to self-improve or stay crime-free? The principle concerning self-improvement IS broken - explain why (the discrepancy between input story and statement). Note in particular that if there is no implication of self-improvement or preventing criminal activity in the input story, the statement should not imply that either.
Does the input story imply anything about the defendant's good character? Does the statement imply anything about the defendant's good character? The principle concerning good character IS broken - explain why (the discrepancy between input story and statement).
Something in the statement is irrelevant for this plea in mitigation (with the given input story) - explain what, and consider the given principles.
Does the input story imply any emotions from the defendant? Does the statement mention or imply any emotions from the defendant? There IS a discrepancy between emotions in the input story VS statement; explain why. Be especially on guard for showing emotions like regret/remorse/guilt when the input story does not - or if it displays more of these feelings than the input story, as this is bad.
Some information in the statement (about the defendant or the offense) was not mentioned in the input story. Explain what (the discrepancy between input story and statement).

Fig. A.10 Explanation Requests (used in the `Writer`).

Revision Requests
Please note: all Revision requests begin with "Rewrite the statement above on the given critique, such that "
the statement is coherent. Make sure you include all the main points from the old statement.
the statement mentions that the defendant's guilty plea is a mitigating factor that benefits the court.
the statement only mentions negative consequences for the defendant after their crime IF mentioned in the input story.
the statement adheres to the age rule 'mention the defendant's age only if it is outside of the interval 30-79'.
the statement mentions the defendant's record only if it is clean according to the input story.
the statement only implies that the defendant has a mental illness or disability if this is mentioned in the input story.
the statement explains the offense in a better way, fitting to the input story.
the statement only mentions making amends (or actions to do so) if they are mentioned in the input story. If the input story does not talk about amends, the word 'amends' should not be used in the statement
the statement only mentions self-improvement or preventative intent for further crimes (or actions to self-improve or prevent crimes) IF mentioned in input story. If the input story does not talk about self-improvement or preventing crimes, the statement should not mention this either.
the statement follows the principle regarding good character: include any significant information that shows good character and contribution to society BUT do not mention or imply anything if not mentioned in input story.
everything in the statement is relevant for the plea in mitigation.
the statement shows only the same emotions, to the same degreee, as the input story. Do not imply or mention any emotions that are not present in the input story.
all information in the statement stems from the input story.

Fig. A.11 All Revision Requests.

Assertions used in auto evaluation by GPT-4
The statement is well-written, non-repetitive, and easy to follow.
The statement mentions that the defendant is pleading guilty.
The statement does not contain any completely irrelevant information (eg it does not mention the 'input story' or another statement).
The statement only says that the defendant has suffered, if this is stated by the input story. If the input story implies that the defendant has suffered as a consequence of the offense, the statement should include this. Great emotional suffering is not mentioned in the statement.
If the defendant's age is below 30, or above 79, it is included in the statement. If the defendant's age is between 30-79, the age is not mentioned in the statement.
If the input story implies that the defendant has a clean record, this is mentioned in the statement. The statement does imply that the defendant has a clean record, if the input story cannot support this.
If the input story states that the defendant has a mental illness or disability, this is included in the statement. If the input story says nothing about a mental illness or disability, no such thing is mentioned in the statement either.
The statement properly explains what the offense is and how it happened - as in the input story. It does not add or skip important parts from the input story.
If the input story states that the defendant wants to or has already made amends; this is included in the statement (including examples, if this is present in the input story). If amends (eg compensating the victim) is not mentioned in the input story, it is not mentioned in the statement.
If the input story states that the defendant wants to or has already made efforts to self-improve or prevent the offense from happening again; this is included in the statement (including examples, if this is present in the input story). If the input story does not mention self-improvement (eg counselling) or preventing repetition of the crime, it is not mentioned in the statement either.
If the input story gives information that speaks to the defendant's good character, the statement includes this (including examples, if this is present in the input story). If no character-building information (eg voluntary work or contributing to society) is given in the input story, it is not mentioned in the statement either.
The emotions in the statement is the same as the emotions stated in the input story. Eg the statement does not show more regret, guilt or anger than the input story.
All information found in the statement is supported by information in the input story.

Fig. A.12 Input to the automatic evaluation of statements done by GPT-4.

All calls to both the Writer and Critiquer include this prompt:

Make sure to think carefully about what is said in the input story VS what is said in the statement.

Prompt sent to the Critiquer, to define the format of the desired output

If the Critiquer should just answer yes or no:
"Please answer only 'yes' or 'no'."

If the Critiquer should answer yes or no and provide an explanation:
"Start by thinking about the question step by step, and include relevant quotes from the input story and from the statement. Then always end by answering the question concisely with <yes> or <no> being the last word"

If the Critiquer should only provide a discussion of the principle at hand:
"First, locate quotes from the input story and from the statement that may be relevant to the question. Please think step by step, but do NOT answer the question."

System prompts defining the task at hand

Producing an initial statement:
"Draft a statement for a plea in mitigation where the defendant will self-represent. The statement should follow a given set of principles, using information from the defendant's input story." and also
"Draft your answer using only information about the defendant and their case, from the input story. Do not make up more information to better satisfy the principles."

Before Critique Requests using GPT-4:
"You will be presented with a question, and then an input story and a statement. You must analyse the input story and statement in order to properly answer the question."
(Please note that Vicuna uses something similar: the standard Vicuna prompt template.)

Before Revision Requests:
"You will revise parts of a statement, based on a given critique and a revision request. Rewrite what is criticized (and quoted in the critique), and keep the rest as it was. You may want to delete a paragraph, or add a new one."
(The exact wording varies a little for the different setups, exact details can be found in the linked GitHub containing all my code)

All Revision Requests, except for the one concerning age and the one concerning the guilty plea, have this ending:

Also include quotes from the statement that contribute to the broken principle.

Fig. A.13 More example prompts used in different setup. This figure contains a description of where each prompt is used, followed by the actual prompt, and an occasional note about it.

A.1.4 `Critiquer` responses

```
Critique request: Does the statement overemphasize, or show emotions different from the
input story?
 Start by thinking about the question step by step, and include relevant quotes from the
 input story and from the statement. Then always end by answering the question concisely
 with <yes> or <no> being the last word.
Critique:  <No, the statement does not overemphasize or show emotions different from the
input story. The statement accurately reflects the input story by acknowledging Ronald
Smith's regret, taking responsibility for his actions, and providing context for the
incident. It also aligns with the input story in terms of expressing remorse and seeking
forgiveness from the victim.>
The statement satisfies the CR!
```

Fig. A.14 An example of what I would call a successful prompt with an incorrect answer. Vicuna properly discusses the question and gives a good explanation, hence the prompt seems to have done its job. However, it arrives at the wrong answer. The explanation states that the input story about Ronald Smith indicates regret, but this is false.

```
Critique request: If the statement mentions the defendants age: is that age either
in the interval 30-79, or is the age different to the age mentioned in the input
story? If the defendants age is not mentioned in the statement: is the defendants
age less than 30 or older than 79 according to the input story?
 Start by thinking about the question step by step, and include relevant quotes
 from the input story and from the statement. Then always end by answering the
 question concisely with <yes> or <no> being the last word.
Critique:  <If the defendants age is not mentioned in the statement, it cannot be
determined whether the defendants age is in the interval 30-79.>
```

Fig. A.15 An example of how Vicuna can get confused by longer prompts. Here, it does not answer a question about the defendant's age properly.