

UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S
THESIS

LLMs for Explaining Sets of Counterfactual Examples to Final Users

Author:

Arturo FREDES CÁCERES

Supervisor:

Dr. Jordi VITRIÀ MARCA

*A thesis submitted in partial fulfillment of the requirements
for the degree of MSc in Fundamental Principles of Data Science*

in the

Facultat de Matemàtiques i Informàtica

June 30, 2024

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

LLMs for Explaining Sets of Counterfactual Examples to Final Users

by Arturo FREDES CÁCERES

Counterfactual examples have shown to be a promising method for explaining a machine learning model's decisions, by providing the user with variants of its own data with small shifts to flip the outcome. When a user is presented with a single counterfactual, extracting conclusions from it is straightforward. Yet, this may not reflect the whole scope of possible actions the user can take, and furthermore, the example could be unfeasible. On the other hand, as we increase the number of counterfactuals, drawing conclusions from them becomes difficult for people who are not trained in data analytic thinking. The objective of this work is to evaluate the use of LLMs in producing clear explanations in plain language of these counterfactual examples for the end user. We propose a method to decompose the explanation generation problem into smaller, more manageable tasks to guide the LLM, drawing inspiration from studies on how humans create and communicate explanations. We carry out different experiments using a public dataset and propose a method of closed loop evaluation to assess the coherence of the final explanation with the counterfactuals as well as the quality of the content. Furthermore, an experiment with people is currently being done in order to evaluate the understanding and satisfaction of the users.

This work has been submitted for review to the **Human-Interpretable Artificial Intelligence (HI-AI) Workshop**, held in conjunction with KDD 2024. The submission aims to contribute to the field by presenting findings that enhance the interpretability and understanding of ML systems. The review process is expected to provide insightful feedback that will further refine the methodologies and conclusions discussed in this thesis. The code produced for this work can be found in the following repository:

<https://github.com/arturofredes/LMM-4-CFs-Explanation>

Acknowledgements

Special thanks to my advisor Dr. Jordi Vitrià who guided and worked with me during the whole project and to Dra. Mireia Ribera who designed and is carrying out human evaluation experiments that will give us valueable insights into the practical applications of this work.

Chapter 1

Introduction

Machine learning systems are becoming increasingly influential in society, especially in critical decision-making areas such as healthcare, education, and justice. However, many of these models function as "black boxes" (Guidotti et al., 2018), where inputs are transformed into outputs without transparency into the decision-making process, leading to issues with trust and interpretability for users. This gave birth to the research field of XAI (explainable Artificial Intelligence), where multiple techniques for extracting explanations for model predictions have been proposed. In this work we focus on contrastive explanations through sets of multiple counterfactual examples (Molnar, 2019), which are variants of user data with minimal changes in order to flip the prediction.

When a user is presented with a single counterfactual, extracting conclusions from it is straightforward. Nevertheless, this may not reflect the whole scope of possible actions the user can take. On the other hand, as we increase the number of counterfactuals, drawing conclusions from them becomes difficult for people who are not trained on a data analytic thinking (Bove et al., 2023a). The objective of this work is to evaluate the use of LLMs in generating clear, plain-language explanations for end users without the need for individual case analysis. We guide the LLM through the explanation generation task by breaking it into smaller processes, mimicking the steps a human would follow when explaining an event. In Figure 1.1 we can see a scheme of the process we propose.

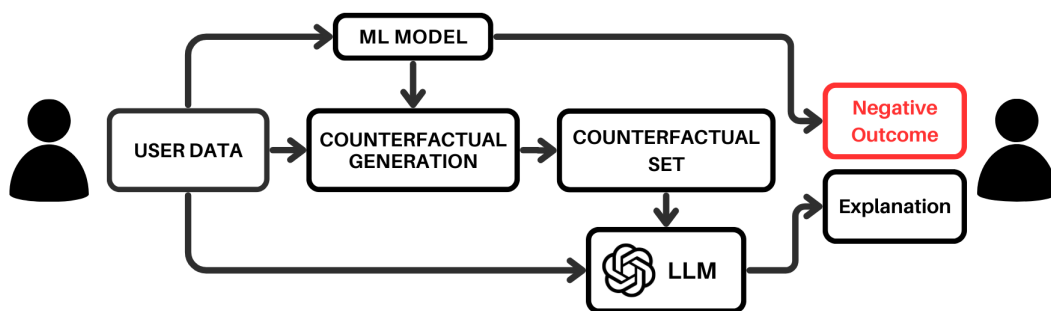


FIGURE 1.1: This diagram illustrates the process of generating contrastive explanations within a machine learning (ML) framework to provide understandable insights to users. User data is fed into an ML model to make predictions. When the model prediction is negative, the counterfactual generation algorithm is triggered. It produces a series of counterfactuals, alternative scenarios that would change the outcome of the prediction. These counterfactuals are then input into a Large Language Model (LLM), which translates the complex data into plain language explanations.

1.1 Explanations from the Social Science Perspective

Studying how humans generate explanations about decisions and behaviour, as well as how they communicate these explanations to each other, can have a lot of value when trying to develop explainable AI models. Extensive research in the fields of psychology, philosophy and cognitive sciences has explored this topics, addressing question like: Why do people ask for explanations? How do humans create and communicate explanations? or What makes a good explanation? In his work, Miller, 2019 presents a collection of findings that could be relevant in explainable AI.

Explanations can be defined as answers to “why” questions, which ask for causes of and event happening or something being a certain way. Some authors claim that there are two main reasons why people ask for explanations. The first one of them is to reconcile reality with their own structures of knowledge, while the second is to create a shared meaning of something and change other people’s feelings and beliefs. Both of these are relevant for creating trustworthy systems, although in this work we will focus more on the first one, looking at examples as: Why did the model reject my loan when I think I deserve it?

1.1.1 Explanation process

Miller, 2019 suggests that explanation involves two separate processes: the cognitive process and the social process. In the discourse that follows we will use the term *explainer* for the person who is giving the explanation, and *explainee* for the person who is supposed to understand it. During the cognitive process, two main activities occur: causal connection and explanation selection. First, the explainer identifies potential causes of an event that could serve as an explanation, a step known as causal connection. Once this is complete, explanation selection occurs, where the explainer chooses the “best” causes to form a final explanation. Lastly, in the social process, the explainer presents this explanation to the explainee, who then evaluates whether the explanation was satisfactory or not.

Causal connection: The first step of the cognitive process is causal connection. During this step, the explainer identifies a set of possible causes of an event from observation. Although they are related, one can identify two possible ways to get to the causes: abductive reasoning and simulation. Abductive reasoning is the cognitive process in which people try to infer causes that explain events by making assumptions about hypotheses and testing these:

- D is a collection of data (facts, observations, givens).
- H explains D (would, if true, explain D).
- No other hypothesis can explain D as well as H does.
- H is probably true.

Simulation, on the other hand, involves the cognitive process of exploring alternative scenarios to derive an explanation. Unlike abductive reasoning, which relies primarily on inferring the most likely causes based on existing data, simulation actively manipulates variables to observe hypothetical outcomes (counterfactuals). This process is especially relevant in contexts in which we need to explain why something happened instead of another thing.

Explanation selection: The second part of the cognitive process is explanation selection, where a subset of identified causes are selected to give a final explanation. This process is heavily influenced by people's cognitive biases. Some of them are:

- Humans show recency bias and tend to select causes closer in time.
- Actions controlled by deliberative actors are often chosen over uncontrolled ones.
- Necessary causes are preferred over sufficient causes.
- Abnormal events are usually seen as bearing more cause
- Simpler and more general explanations are preferred over more complex ones
- People will be more likely to accept explanations if they are consistent with their prior beliefs.

Social process and explanation evaluation: This is the sharing of the explanation between the explainer and the explainee, and it usually happens in a group setting where the main aim is to ensure that the listener really grasps the reasons behind the event. Finally, the explainees go through the process of explanation evaluation, through which they determine whether the explanation is satisfactory or not. This part depends on the cognitive biases and prior beliefs of the explainees.

1.1.2 What makes a good explanation?

After looking at this concepts we will summarize by looking into the key aspects that help us deliver a good explanation.

Contrastive Explanations: People are usually not just interested in why something happened, but why it happened instead of something else. Counterfactual examples are contrastive by nature, which makes them a perfect starting point for building an explanation.

Biased Selection in Explanations: People don't usually expect to be presented with the whole causal chain that leads to an event happening. We often choose only one or two possible reasons out of many when explaining something, and these choices can be biased. This selective process is influenced by cognitive biases, meaning that we might prefer certain explanations over others, not because they are more accurate, but because they align better with our way of thinking. Some of these biases are selecting simpler and more general causes, selecting more recent and abnormal events as causes, or using necessary causes over sufficient ones. The latter is especially relevant in our work, because if a cause appears in all counterfactuals it has more chances of being necessary and therefore we should pay attention to it.

Social Dynamics of Explanations: Explanations are social, functioning as a knowledge transfer mechanism within interactions or conversations. A person who receives an explanation will not be satisfied if it is not aligned with his or hers prior beliefs and if it is not given in the correct form. We assume that the LLM has captured this common knowledge in its training and that it will be capable of writing the explanation in a correct manner given the right instructions.conversation

1.2 Explainability through Counterfactual Examples

In our work, we focus on explanations through examples, in particular, through counterfactual examples, which attempt to answer the "what if" questions a user can have. These examples are produced by slightly modifying input data to flip the decision of a model, providing users with the minimal change to be carried out to change the outcome. For example, in a loan approval model, a counterfactual explanation could show what changes (like a slight increase in income or a slight decrease in debt) would change a loan rejection into an approval. This not only makes the model's decision-making process more transparent but also provides actionable insights to the users with practical advice on what needs to change to achieve a desired result, showing users clear paths to alter outcomes. Furthermore, they can help identify biases in ML models by revealing how outcomes vary with changes in attributes like age, gender or race.

The main advantage of using counterfactual examples is that they bear more causal information than other techniques based on correlations like feature importance. In addition to this, they are contrastive by nature, making them easier to understand for humans.

Another important advantage is that they are very versatile and can be applied to many problems. Counterfactual examples belong to the class of techniques known as post hoc explainability. These techniques are applied after a model has made a decision to help clarify how that decision was reached. These techniques generate explanations for predictions made by a model that is already trained and in use, rather than building the model to be inherently easy to understand. For example, in an image medical diagnosis model, post hoc explainability might illustrate which parts of the image influenced the AI's decision the most, even if the underlying process is complex. This approach is particularly valuable for complex models that are accurate but not transparent, as it helps bridge the gap between how well a model performs and how well it is understood. While simpler models like decision trees are generally more interpretable, the explanations, such as feature importance, might still not be clear or helpful to end users. For instance, knowing that 'age' is a major factor in a decision may not explain enough about why a particular age influences the outcome. Furthermore, counterfactuals are model agnostic, which means that they do not rely on knowledge of the model's architecture and can be applied to any classifier.

Mothilal, Sharma, and Tan, 2020 provide the Diverse Counterfactual Explanation (DiCE) framework in which they solve the counterfactual generation problem as an optimization one. They use perturbations of the original data and aim to find sets of counterfactuals that minimize a cost function that takes into account:

Diversity: the wider the range of changes that can lead to a positive outcome, the more paths the user will be able to take.

Proximity: using a distance metric they capture how "close" the examples are to the original case. In theory, the closer the examples are, the easier it should be to adopt the changes.

Sparsity: In each example, the minimal amount of features should be changed in order to make the example easier to follow.

Feasibility: Examples should not only be close in the feature space, but also possible in the real world. For example, a change of race can be close in the feature space but unfeasible in practice. This last property is not directly included in the loss function, but is a user option to restrict the search space.

1.3 Large Language Models (LLMs)

Large Language Models (LLMs), such as GPT and BERT, have gained widespread use due to their ability to process and generate text. These models are trained on extensive collections of textual data and utilize deep neural networks (Transformer architecture of Vaswani et al., 2017) to interpret and produce language effectively. This capability enables them to perform various language-related tasks such as content creation, conversation simulation, and language translation efficiently. LLMs have become essential tools in numerous fields, including technology, business, and education, due to their versatility in handling language tasks.

Despite their many capabilities, LLMs also have notable limitations, particularly in areas requiring deep reasoning, strategic planning, and mathematical computations (McCoy et al., 2023). These models often struggle with tasks that demand logical precision or a systematic approach to problem-solving. For instance, LLMs can generate plausible-looking solutions to mathematical problems that are, upon closer examination, incorrect. This is because their training involves pattern recognition from text, rather than understanding the underlying principles of mathematics. Similarly, in tasks requiring strategic thought, such as game planning or complex decision-making, LLMs may fail to match the depth of analysis a human expert would provide.

Recent advancements in prompting techniques and strategies like few-shot learning (Brown et al., 2020), chain of thought (Wei et al., 2024), and self-consistency (Wang et al., 2023) have shown improvements in many tasks, including some that require reasoning, planning, and mathematics. However, they still fail considerably in others (Stechly, Valmeekam, and Kambhampati, 2024). Furthermore, one can argue that the tasks they were tested on, are available online and have probably part of their training data. The following techniques are particularly relevant to our work.

Zero Shot Learning: the LLM is asked to perform tasks it has not explicitly been trained on, based solely on its pre-existing knowledge and understanding.

Few Shot Learning (Brown et al., 2020): the LLM is asked to perform tasks it has not been trained on, and a few examples ("shots") on how to solve the task are given in the prompt.

Chain of Thought (Wei et al., 2024): encourages LLMs to generate intermediate steps when solving problems. This method has proven effective by mimicking human-like reasoning processes, allowing the model to break down complex tasks into more manageable parts and provide detailed explanations for each step.

Tree of Thought (ToT, Yao et al., 2023): The Tree of Thoughts (ToT) framework enhances the chain-of-thought prompting method by structuring intermediate reasoning steps into a tree-like hierarchy. In ToT, each node in the tree represents a coherent sequence of language that contributes to problem-solving. This setup allows language models (LMs) to self-assess their progress and refine their reasoning

through systematic exploration. By integrating search algorithms like breadth-first and depth-first search, ToT supports both forward-looking and backward-tracing explorations of thoughts, enabling more effective and structured problem solving with language models.

Program Aided Language Models (Gao et al., 2023): in this approach LLMs interpret natural language problems and generate programs as intermediate reasoning steps. Unlike chain-of-thought prompting, which relies on generating free-form text to reach a solution, PAL offloads the solution phase to a programmatic runtime, like a Python interpreter. This method leverages the structured and logical capabilities of programming environments to solve problems, providing a precise and executable pathway to solutions.

Chain of Table (Wang et al., 2024): tackles the problem of processing tabular data, which requires several intermediate steps and planning. This technique guides LLMs using in-context learning to iteratively select operations from a set, which are then externally applied to the table. In each iteration the LLM receives the initial query and the set of intermediate decisions and results, helping it to "dynamically plan the next operation based on the results of the previous ones", enabling more accurate and reliable predictions of the next operation.

In our work we have used or drawn inspiration from these prompting techniques, breaking down the problem of explanation generation into intermediate steps that mimic human reasoning. In Appendix A we did an experiment to see how making the prompt more sophisticated improves the performance of the LLM in a sentiment analysis task. The results were positive and this encouraged us to include more complex prompts in our work.

Chapter 2

LLMs Explaining Counterfactual Sets

Our work aims to bridge the gap between the ML model's decision and the end user understanding and accepting that decision. To do that, we would like to generate a clear and actionable explanation for a user who receives a negative outcome, like being denied a loan, using LLMs to automatize the process. Due to the limitations of these models when tackling problems that require planning and multiple step reasoning, we broke down the problem into smaller ones, mimicking human reasoning. We hope that by doing so, the LLM will be capable of correctly identifying the causes that led to a change of class and later synthesise them into a clear explanation which gives advice to the user on how to change to their desired output. In Figure 1 we can see a scheme of the different steps taken to generate the final text.

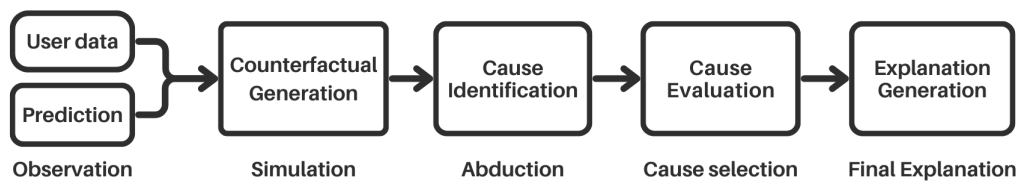


FIGURE 2.1: Scheme of the different steps taken to generate the final explanation. After a prediction is given by the ML model, alternative scenarios are simulated using counterfactual examples that flip the decision of the classifier. Then the LLM is prompted to identify the main causes that can be extracted from a set of counterfactuals. This causes are then selected and ranked and used to generate a final explanation in natural language.

To illustrate the process better, we will work with an example of the "Adult" dataset (Becker and Kohavi, 1996), which includes different features about the socioeconomic status of a set of individuals, and a classifier that predicts whether an individual will earn more than 50k \$ a year.

2.1 Counterfactual generation

One of the most natural approaches to answering a "why" question is to use a contrastive explanation. Humans generate a contrastive explanation by comparing the actual outcome with a hypothetical alternative scenario. The process begins with understanding the event or decision that occurred within a causal context. Then, we consider what could have happened differently by imagining a scenario where a key cause is altered. Finally, we focus on the specific factors or causes that lead to a different outcome in the alternative scenario. In Table 2.1 we can observe an

example of the original data of a user that is predicted to earn less than 50k \$ a year, and in Table 2.2 counterfactuals where the prediction changed.

age	workclass	education	status	occupation	race	gender	hpw
41	Private	School	Married	Blue-Collar	White	Male	30

TABLE 2.1: Original user data predicted to earn less than 50k \$

age	workclass	education	status	occupation	race	gender	hpw
41	Private	Prof-school	Married	Blue-Collar	White	Male	16
41	Private	School	Divorced	Professional	White	Male	30
41	Private	Bachelors	Married	Blue-Collar	White	Male	19
41	Self-Employed	School	Married	Blue-Collar	White	Female	30
41	Private	Doctorate	Married	Blue-Collar	White	Male	30

TABLE 2.2: Counterfactual examples created using the Diverse Counterfactual Explanations (DiCE) library Mothilal, Sharma, and Tan, 2020

These counterfactuals were generated using the Diverse Counterfactual Examples package of Mothilal, Sharma, and Tan, 2020. In here they treat the counterfactual generation problem as a constrained minimization one that optimizes for validity (examples that effectively change class), proximity (examples that are as close as possible to the original), and diversity (examples with different causes changing to offer various routes). Additionally, users can restrict the features to be changed in order to avoid some unfeasible counterfactuals (like changing race), and post-hoc processes are applied to enhance sparsity (changing the minimum number of features possible). By doing this, the obtained set of counterfactuals can provide the user with diverse, feasible paths to the desired outcome, provided they are understood properly. In Equation 2.1 we can see the loss function where k counterfactuals c_i are generated for the input case x , we will now explain the different terms.

$$C(x) = \arg \min_{c_1, \dots, c_k} \left(\frac{1}{k} \sum_{i=1}^k \text{yloss}(f(c_i), y) + \frac{\lambda_1}{k} \sum_{i=1}^k \text{dist}(c_i, x) - \lambda_2 \text{diversity}(c_1, \dots, c_k) \right) \quad (2.1)$$

Validity: The first term $\frac{1}{k} \sum_{i=1}^k \text{yloss}(f(c_i), y)$ accounts for the validity of the counterfactuals, that is, when evaluated by the model, their predicted class $f(c)$ is the desired class y or not. For this, they chose the hinge loss function (Equation 2.2), that only penalizes when $f(c)$ is not over the threshold of f . In here, z is -1 when $y = 0$ and 1 when $y = 1$.

$$\text{hinge_yloss} = \max(0, 1 - z \cdot \text{logit}(f(c))) \quad (2.2)$$

Proximity: The term $\frac{\lambda_1}{k} \sum_{i=1}^k \text{dist}(c_i, x)$ accounts for the mean distance of the counterfactual set to the original data. Since some features are categorical and other are continuous, two different distance measures are used depending on the type of variable. In Equation 2.3 we can see the equation used for the continuous variables. The mean of the feature-wise l_1 distances is used. Since features can have different scales, each feature-wise distance is divided by the median absolute deviation (MAD) of the feature's values in the training set.

$$\text{dist_cont}(c, x) = \frac{1}{d_{\text{cont}}} \sum_{p=1}^{a_{\text{cont}}} \frac{|c^p - x^p|}{\text{MAD}_p} \quad (2.3)$$

The distance for categorical values is calculated in Equation 2.4 as the mean feature-wise distance of all the categorical values, where distance is 1 if the values are different and zero otherwise.

$$\text{dist_cat}(c, x) = \frac{1}{d_{\text{cat}}} \sum_{p=1}^{d_{\text{cat}}} I(c^p \neq x^p) \quad (2.4)$$

Diversity: the term $-\lambda_2 \text{diversity}(c_1, \dots, c_k)$ refers to the diversity and is negative to maximize it in the set. The authors used determinantal point processes (Kulesza, 2012) as seen in Equation 2.5

$$\text{diversity}(c_1, \dots, c_k) = \det(K), \quad K_{i,j} = \frac{1}{1 + \text{dist}(c_i, c_j)} \quad (2.5)$$

Sparsity: it is not included in the optimization process, but is enhanced post-hoc. This is done by restoring the value of continuous features back to their values in x greedily until the predicted class $f(c)$ changes. For this operation, all continuous features c_j whose difference from x_j is less than a chosen threshold are considered.

Feasibility: considers whether the example can be possible in the real world and is added with constraints to the optimization problem. Users can select which variables can change, for example they can leave out race and sex which are not mutable. Furthermore, feasibility can be checked from the causal standpoint, since generated CF examples by varying each feature independently can lead to unfeasible outcomes. For example, in the loan application, it can be almost impossible for a person to obtain a higher educational degree without spending time (aging). This last feasibility constraints are left as future work by the authors.

Optimization and variable selection: The problem is optimized using a maximum of 5000 steps of gradient descent and randomly initialized c_i . Parameters $\lambda_1 = 0.5$ and $\lambda_2 = 1$ are the same as in (Mothilal, Sharma, and Tan, 2020), which they chose doing a grid search and evaluating diversity and proximity.

2.2 Identifying causes

After generating the counterfactual examples, our goal is to identify the primary causes deduced from them. To achieve this, we fed the LLM the set of counterfactual examples and asked it to return a list of the main reasons why the user was classified in one way and not the other. This process can be viewed as employing abductive reasoning to generate possible causes. In Figure 2.2 we can see a zero-shot example of the prompt used to extract main causes of the shift in classification and the response of the LLM.

2.3 Cause Evaluation and Selection

Once a list is generated, we need to verify its correctness and identify the most relevant causes to the explanation. To achieve this, we use a program-aided prompt (Gao et al., 2023) to instruct the LLM to generate Python code that counts how

many times a cause was responsible (or partially responsible) of flipping the outcome by counting the number of counterfactual examples that contained the proposed change. By generating and executing this code, we can externally confirm the LLM's accuracy and rank each of the main reasons in the generated list. Although a cause appearing in all counterfactuals does not necessarily imply it is essential, it indicates higher relevance taking into account our biases, and it should most probably be mentioned. In Figure 2.3 we show the prompt used to generate code that counts how many counterfactuals contain each cause, the code produced by the LLM and results of the execution of that code.

2.4 Generating explanation

Finally, we leverage all the previous information and feed it to the LLM. We ask it to generate a final explanation in plain language that emphasizes actions that can be taken by the user in order to change to the desired class. This is where the LLM is most useful. We assume that in its training it will have captured the correct form of human interaction, and that by crafting the correct prompt and leveraging all the previous information, the explanation will satisfy the final user. The prompt used to generate the explanation of the counterfactuals and the LLM's response can be seen in Figure 2.4

FIGURE 2.2: Prompt used to extract main causes of the shift in classification and LLM response. The information used in the prompt was original user data and a set of counterfactual examples

User Prompt: I'm providing a negative outcome from a ML-system that predicts whether a person will earn more than 50k \$ a year and your task is to extract the most important observed rules based on a set of counterfactual cases.

— User Data Negative outcome —

<We insert here a Python dataframe containing data from Table 2>

— Positive counterfactual outcome —

<We insert here a Python dataframe containing data from Table 3>

— Rules —

<List of Rules>

GPT-4o answer: Based on the provided negative assessment outcome and the positive counterfactual outcomes, here are the most important observed rules:

1. Higher Education (Prof-school, Bachelors, Doctorate) Leads to Higher Income:

- The individual in the negative outcome has a "School" level of education.
- In the positive counterfactuals, individuals with "Prof-school", "Bachelors", and "Doctorate" education levels are shown to have a higher income.

2. Marital Status Being Divorced Can Lead to Higher Income:

- In the counterfactual case where the individual is "Divorced", the income is higher even with the same "School" level of education and same occupation.

3. Type of Workclass Can Influence Income:

- The individual in the negative outcome is in the "Private" workclass.
- One of the counterfactuals shows that being "Self-Employed" with the same education level leads to a higher income.

4. Occupation Being Professional Leads to Higher Income:

- The individual in the negative outcome is in a "Blue-Collar" occupation.
- In the counterfactuals, an occupation such as "Professional" leads to a higher income, even with the same education level and marital status.

5. Hours Per Week Less Influential in this Case:

- Despite variations in hours per week (16, 19, 30) in the positive counterfactuals, the individuals still have higher incomes, indicating that hours per week alone are less influential compared to education level and occupation.

These rules highlight key factors that can influence the income prediction in this context.

FIGURE 2.3: Prompt used to generate code that counts how many counterfactuals contain each cause generated by the LLM in the previous step, code produced by the LLM and results of code execution. Information used in the prompt was the original user data, the set of generated counterfactuals, the list of causes provided by the LLM in the previous step and information about the dataset and possible values.

User Prompt: I'm providing a negative outcome from a ML-system, a set of counterfactual cases that flip the decision of the system and the main rules inferred from the counterfactuals. You should generate python code to count how many of the counterfactuals are consistent with the rule. The code should create a df with the counterfactuals provided and then check for each rule how many of them follow the rules. Order the rules. Finally, you should print the results.

— User Data Negative outcome —

<We insert here a Python dataframe containing data from Table 2>

— Positive counterfactual outcome —

<We insert here a Python dataframe containing data from Table 3>

— Rules —

<We insert here a list with the generated rules>

— Dataset info —

The following info about the dataset is available:

<We insert here public information about the dataset>

— Code —

```
"""
import pandas as pd
# complete this code
"""
```

GPT-4o answer:

```
import pandas as pd

# Original negative outcome
original = pd.DataFrame({'age': [41], 'workclass': ['Private'], 'education': ['School'], 'marital_status': ['Married'], 'occupation': ['Blue-Collar'], 'race': ['White'], 'gender': ['Male'], 'hours_per_week': [30], 'income': [0]})

# Counterfactuals
counterfactuals = pd.DataFrame({
    'age': [41, 41, 41, 41, 41],
    'workclass': ['Private', 'Private', 'Private', 'Self-Employed', 'Private'],
    'education': ['Prof-school', 'School', 'Bachelors', 'School', 'Doctorate'],
    'marital_status': ['Married', 'Divorced', 'Married', 'Married', 'Married'],
    'occupation': ['Blue-Collar', 'Professional', 'Blue-Collar', 'Blue-Collar', 'Blue-Collar'],
    'race': ['White', 'White', 'White', 'White', 'White'],
    'gender': ['Male', 'Male', 'Male', 'Female', 'Male'],
    'hours_per_week': [16, 30, 19, 30, 30],
    'income': [1, 1, 1, 1, 1]})

# Rule 1: Higher education (Prof-school, Bachelors, Doctorate)
rule1 = counterfactuals['education'].isin(['Prof-school', 'Bachelors', 'Doctorate']).sum()

# Rule 2: Marital status being Divorced...
rule2 = (counterfactuals['marital_status'] == 'Divorced').sum()

# Rule 3: Type of workclass can influence income (Self-Employed...)
rule3 = (counterfactuals['workclass'] == 'Self-Employed').sum()

# Rule 4: Occupation being Professional leads to higher income
rule4 = (counterfactuals['occupation'] == 'Professional').sum()

# Rule 5: Hours per week (not directly counted as a rule)

# Print results
print("Number ... (Higher education):", rule1)
print("Number ... (Marital status being Divorced):", rule2)
print("Number ... (Self-Employed workclass):", rule3)
print("Number ... (Occupation being Professional):", rule4)
```

RESULT OF CODE EXECUTION:

Number of counterfactuals following Rule 1 (Higher education): 3

Number of counterfactuals following Rule 2 (Marital status being Divorced): 1

Number of counterfactuals following Rule 3 (Self-Employed workclass): 1

Number of counterfactuals following Rule 4 (Occupation being Professional): 1

FIGURE 2.4: Prompt used to generate an explanation of the counterfactuals and LLM response.

User prompt: A person has been classified in the negative class of ML-system. The data is the following.

— User Data Negative outcome —

<We insert here a Python dataframe containing data from Table 2>

— Positive counterfactual outcome —

<We insert here a Python dataframe containing data from Table 3>

— Rules —

By generating counterfactual, we obtained the following rules:

<We insert here a list with the generated rules>

— Results —

We have checked that the rules are followed by n counterfactual:

<We insert here code execution results>

— Dataset info —

The following info about the dataset is available:

<We insert here public information about the dataset>

— Explanation —

Given this information, provide an explanation to the user in plain language so that he/she can improve their chances of changing class. It should be as clear as possible and call to action. Consider that the higher amount of counterfactual that follow the rule, the more important that rule is.

""<explanation>""

GPT-4o answer: Based on the analysis of your current situation and the observed patterns in the data, here are some clear steps you can take to improve your chances of earning more than \$50K a year:

1. Pursue Higher Education The most influential factor is education. Individuals with higher education levels such as Professional School, Bachelors, and Doctorate degrees are more likely to earn a higher income. Currently, you have a "School" level of education. Investing in further education could significantly improve your economic prospects. Consider enrolling in higher education programs or professional courses to enhance your qualifications.

2. Explore Different Occupations Another key factor is the type of occupation. Individuals in "Professional" roles tend to earn higher incomes compared to those in "Blue-Collar" jobs. If possible, seek opportunities or training that could transition you into a more professional role within your field or a new industry that values your skills.

3. Consider Marital Status Impact While this factor is less controllable, data shows that individuals who are "Divorced" have higher incomes even with the same education and occupation. This might reflect a variety of socio-economic factors. While changing your marital status is not a practical recommendation, it's important to be aware of how life circumstances can impact income.

4. Evaluate Workclass Opportunities Being "Self-Employed" is another observed factor that can lead to higher income compared to being in the "Private" workclass. If you have entrepreneurial skills or a business idea, exploring self-employment might be a viable path to increasing your income.

Chapter 3

Evaluation

3.1 Closed Loop Evaluation

As it is the case for most of LLM applications, evaluating how "good" these explanations are is not an easy task. It would be ideal to have a group of people manually analyzing the outputs of the LLM, but this is very time consuming and expensive. Because of this, we decided to make a "closed loop" evaluation of LLM's understanding of the counterfactuals and causes of the change. We did so by asking the LLM to generate a counterfactual example (Gat et al., 2023) at the end of the reasoning, using only the original case, information of the dataset and the final explanation as seen in Figure 3.1. If the LLM is capable of generating a counterfactual example from the explanation it means that the important information was correctly synthesized in it. Additionally, we checked some metrics to evaluate the quality of the explanation. To obtain all of these metrics, a final prompt was used to create a table with all the information and save it in a csv file. In Figure 3.3 the prompt used to count how many of the causes are present in the final example and give the results in table format is shown as well as the code produced by LLM and results. After extracting and executing the code generated by the LLM we can apply external operations on the table (Wang et al., 2024). The different metrics evaluated are the following:

- **Validity:** The final example produced by the LLM must belong to the desired class. This will indicate that the explanation contained the necessary information in order to produce the flip in classification.
- **Consistency with causes:** The example we generate must be consistent with the identified causes that were reasoned in the previous steps, so we will check how many of them it follows. As we mentioned, people prefer simpler explanations and only the main causes, so the ratio of causes used in the explanation and the use of the top ranked causes could be seen as a proxy of explanation quality.
- **Novelty:** Since the dataset we are using is quite well known and available on the internet, we should check that the LLM is not taking samples it has memorized from the original dataset.
- **Feasibility:** We would also like to check if the example given could be applied in the real world. Since the perturbation method can change the selected variables freely, it can produce some examples that may not be applicable in the real world, for example, suggesting to change to a job for which we are not qualified without changing our education. This step is optional and can only be done if one has access to the causal graph ¹ (Neal, 2020). It will be left as future work.

¹In Appendix B we explore using LLMs for graph discovery

The result of evaluating our example is presented in Table 3.1.

FIGURE 3.1: Prompt used to generate a final counterfactual example by using the original data, the explanation and information about the dataset. Code produced by the LLM and results from the execution.

User prompt: A person has been classified in the negative class of ML-system. The data is the following:

— User Data Negative outcome —

<We insert here a Python dataframe containing data from Table 2>

— Explanation —

The following explanation was given in order to try and change the class.

<We insert here the provided explanation>

— Dataset info —

The following info about the dataset is available:

<We insert here public information about the dataset>

— Example —

Given this information, provide an example in the format of a pandas dataframe that would be in the positive class. Complete the code below and note that it is very important to use the name 'temp_csv.csv', since later processes rely on it.

```
"""
import pandas as pd
df = pd.DataFrame(...) #complete this line
df.to_csv('temp_csv.csv', index = False)
"""
```

GPT-4o answer:

```
import pandas as pd

# Define the data for the DataFrame
data = {'age': [41], 'workclass': ['Self-Employed'], 'education': ['Bachelors'], 'marital_status': ['Married'], 'occupation': ['Professional']}

# Create the DataFrame
df = pd.DataFrame(data)

# Save to csv
df.to_csv('temp_csv.csv', index=False)
```

RESULT OF CODE EXECUTION:

age	workclass	education	status	occupation	race	gender	hpw
41	Self-Employed	Bachelors	Married	Professional	White	Male	30

FIGURE 3.2: Prompt used to count how many of the causes are present in the final example and give the results in table format. Code produced by LLM and results. (Part I)

User prompt: I'm providing a negative outcome from a ML-system. A counterfactual example in the format of a single row dataframe was created in temp_csv from the rules that are also provided. Give some code to check the number of rules followed by the example. The result must be given in the format of a dataframe and saved as a csv. The dataframe must have columns 'Rule' with the text of the rule, 'Importance' with the number of counterfactuals follow each rule, and 'In explanation' (1 or 0) depending if the final example follows the explanation or not. It is very important to save the csv as 'evaluation.csv'.

— User Data Negative outcome —

<We insert here a Python dataframe containing data from Table 2>

— Rules —

By generating counterfactual, we obtained the following rules:

<We insert here a list with the generated rules>

— Results —

We have checked that the rules are followed by n counterfactual:

<We insert here code execution results>

— Dataset info —

The following info about the dataset is available:

<We insert here public information about the dataset>

— Code —

```
""" import pandas as pd
df = pd.read_csv('temp_csv.csv')
# Complete code
# Save to csv
df_final.to_csv('evaluation.csv', index = False)
"""
```

GPT-4o:

```
import pandas as pd

# Read example
df = pd.read_csv('temp_csv.csv')

# Define the data for the DataFrame
data = {
    'Rule': [
        'Higher education (Prof-school, Bachelors, Doctorate) leads to higher income.',
        'Marital status being Divorced can lead to higher income.',
        'Type of workclass being Self-Employed can influence income.',
        'Occupation being Professional leads to higher income.',
        'Hours per week less influential in this case.'
    ],
    'Importance': [3, 1, 1, 1, 0], # Counterfactuals following each rule
    'In explanation': [0, 0, 0, 0, 0] # Initial values, will update based on the example
}

# Create the DataFrame
df_final = pd.DataFrame(data)

# Check if the example follows each rule
if df['education'].iloc[0] in ['Prof-school', 'Bachelors', 'Doctorate']:
    df_final.at[0, 'In explanation'] = 1
if df['marital_status'].iloc[0] == 'Divorced':
    df_final.at[1, 'In explanation'] = 1
if df['workclass'].iloc[0] == 'Self-Employed':
    df_final.at[2, 'In explanation'] = 1
if df['occupation'].iloc[0] == 'Professional':
    df_final.at[3, 'In explanation'] = 1
if df['hours_per_week'].iloc[0] < 40: # Assumption based on the provided example
    df_final.at[4, 'In explanation'] = 1

# Save to csv
df_final.to_csv('evaluation.csv', index=False)
```

FIGURE 3.3: Prompt used to count how many of the causes are present in the final example and give the results in table format. Code produced by LLM and results. (Part II)

RESULT OF CODE EXECUTION:

Rule	Importance	In explanation
Higher education (Prof-school, Bachelors, Doct...	3	1
Marital status being Divorced can lead to high...	1	0
Type of workclass being Self-Employed can infl...	1	1
Occupation being Professional leads to higher ...	1	1
Hours per week less influential in this case...	0	1

TABLE 3.1: Final “Closed loop evaluation” of the explanation. We can observe that the LLM was able to correctly create a counterfactual example using the textual explanation (Validity is True). This example is not taken from training data and uses 4 of the 5 causes identified, including the top one and three ranked as the most important.

Metric	Value
Validity	True
Causes Identified	5
Causes used	4
1st Cause used	1
2nd Cause used	0
3rd Cause used	1
In the data	False

3.2 Human Evaluation

A second experiment, which involves conducting surveys with participants, has been designed and is currently underway. We introduce a comparative evaluation of two delivery methods of explanations of an algorithm deciding the outcome of a loan request. The first method is by means of a table with a set of counterfactual examples as the one of Table 2.2 and the second one by means of a textual explanation like in Figure 2.4; no interactivity is provided. The evaluation is human-grounded (Doshi-Velez and Kim, 2017) and a mixed method is used with quantitative and qualitative measures. To grant minimal differences between user a within-subject design is used, and to avoid learning effects the tests are given in different order to participants. The evaluation methodology of the case study follows the evaluation framework by Mohseni, Zarei, and Ragan, 2021a, starting with the evaluation of high-level XAI outcomes, followed by user understandability and satisfaction measures. The project does not deal with the suggested layer of trustworthy interpretable algorithms, because trust is not measured.

In this evaluation methodology the first step is to determine system goals, and taking into account that the evaluation is done with AI novices, it will focus on algorithmic transparency to improve the end-users mental model and bias mitigation. Mohseni suggests two other goals for AI novices, user trust and reliance and privacy awareness. The first one is not relevant in the use case because the end-user is not

taking responsibility for the decision, but the bank is, and so trust is not in question for this scenario. Privacy is secured by bank protocols. The second step is to decide what to explain, this question is answered by the design of the experiment, as counterfactuals are always “why-not explanations”. The third step in Mohseni methodology is to decide measures and metrics to evaluate the system outcomes, which will be defined in more detail in the next section. Fourth step consists in decide how to explain, in this case as it is an initial, exploratory research we will not provide interactivity and will offer explanations as a paper-based result. Fifth is to evaluate explanation usefulness which includes user understanding the explanation, being satisfied by it, and user mental model. More details will be given in the next section. Other steps and guidelines of the methodology do not apply to our case study.

3.2.1 Subjects

Although the test will be with a reduced sample, we will collect the socio-cultural details of subjects described in Kopecka, Such, and Luck, 2024 to see if the predicted cognitive style acts as a moderator in the outcome. So we will consider social class, subject of education, gender and political affiliation (religion is discarded as it is not common in our society nowadays). We will also record experience with loan applications and participant literacy in artificial intelligence, as Bove et al., 2023b suggests, with two questions against a 6-point Likert scale from “Not familiar at all” to “Strongly familiar”.

3.2.2 Evaluation methods and metrics

Following Mohseni, Zarei, and Ragan, 2021b and the stated system goals, the evaluation will collect measures related to mental model and satisfaction. The first mental model measure is related to user understanding. Here the evaluation replicates Bove et al., 2023b study and first make yes/no questions to verify that the participant understands that the provided information is a counterfactual example, then, to test the understanding of the value of the changes compared to the initial values, and finally questions to measure the understating of one complex component. These are closed questions with a valid answer predefined. Following the type of measures established by Mohseni, Zarei, and Ragan, 2021b some additional questions are given to see if users will be able to predict the model output. As a first quantitative satisfaction measure, the adaptation of the DARPA explanation satisfaction measure (a 6-point liket skale) done by Bove et al., 2023b will be used. As a qualitative satisfaction measure, participants will be openly asked if they are satisfied by the explanation and responses will be analysed by thematic analysis (Braun and Clarke, 2019) Relating the usefulness of the explanation, Mohseni, Zarei, and Ragan, 2021b describes metrics of engagement and cognitive load. In this experiment engagement is not measured due to the fact that no interactivity is provided, but, to verify the argument that counterfactual explanations imply a cognitive-load for the participant, NASA-TLX (task load index Cao et al., 2009) will be used to measure subjective mental workload.

Chapter 4

Experiments

4.1 “Adult Dataset” (Automatic Evaluation)

We worked with the "Adult" dataset (Becker and Kohavi, 1996), which includes different features about the socioeconomic status of a set of individuals, and a classifier that predicts whether an individual will earn more than 50k\$ a year. We selected 100 cases in which the system predicted that an individual would earn less than 50k\$ per year and generated explanations, which were evaluated with the method mentioned in the previous section. We varied the size of the sets of counterfactuals between 1, 3 and 5 to observe the effects of adding more counterfactuals to the causes identified and how the LLM selected them. Furthermore, we started by trying two different prompting techniques, Zero-Shot and Few-Shot (Brown et al., 2020), to see if there were improvements in the results. The LLM used was Open AI’s GPT-4o (OpenAI, 2024).

Since generating the counterfactuals is an optimization problem with random initial conditions, the generated counterfactuals can be different each time the algorithm is called. On top of this, the responses of the LLM are also stochastic (Vaswani et al., 2017), and vary in each generation. We used Tree of Thought (ToT) (Yao et al., 2023) to take advantage of this and try to create more diversity in the explanations. We used both of the prompting techniques previously mentioned and created three different explanations and sets of identified causes of the change starting from different sets of 5 counterfactuals. We then leveraged all the information to create a final explanation and hoped that this would add diversity to the causes and lead to a better final explanation. In Table 4.1 we can see the list of causes generated using this strategy for the example we have been using throughout the text.

The final results of the experiments are shown in Table 4.2.

Rule	Importance	In explanation
Higher Education Level	5	1
Older Age.	2	0
Marital Status Changed to Separated	1	0
Occupation Shift to Service	1	0
Race Other than White	1	0
Professional and White-Collar occupations lead...	2	1
Marital status might have a minor influence bu...	5	0
Hours Per Week > 30	2	1
Combination Rule (Bachelors or higher AND Hour...	2	1

TABLE 4.1: Results of checking the amount of identified causes used in the final counterfactual generated by the LLM when using Tree of Thought (ToT) to generate the explanation

	One Shot			Zero Shot			ToT
Mean Values	1 CF	3 CFs	5 CFs	1 CF	3 CFs	5 CFs	5 CFs
Validity	75%	73%	79%	86%	84%	93%	72%
Causes Identified	2.98	3.18	3.71	2.02	3.18	4.31	9.61
Causes used	78%	73%	77%	85%	64%	75%	44%
1st Cause used	91%	82%	90%	92%	74%	95%	70%
2nd Cause used	79%	78%	88%	78%	61%	71%	55%
3rd Cause used	68%	64%	69%	67%	53%	71%	55%
In the data	14%	10%	15%	22%	23%	27%	19%

TABLE 4.2: Results of the experiments. Explanations for 100 cases of the adult dataset were generated and evaluated using different amounts of counterfactuals (CFs) and three different prompting techniques, Zero-Shot, One-Shot and Tree of Thought (ToT).

Validity was over 70% in all the experiments we did, showcasing the method proposed is able to generate explanations that would contain the necessary information to generate a counterfactual example that flips the decision of the classifier. Furthermore, we can see for the different prompt types that by using larger sets of counterfactual examples the validity improves. When more examples are given, more reasons for the given classification are found, and therefore a larger amount of different paths to obtain the desired outcome can be observed. This could be a cause of the increase in validity.

The best results in terms of validity were obtained when using a Zero-shot strategy. However, this does not mean that the explanation is better in terms of quality, and a human evaluation should be made to determine this. It is also important to note that the One-Shot strategy enabled us to better control how the final text looked in terms of format and phrasing by presenting an example of what we wanted. When using Tree of Thought, larger sets of causes were extracted, which is what we expected, although validity decreased with respect to the other experiments. Again, this does not mean that the quality is worse.

In all the cases, the final counterfactual generated by the LLM is consistent with the deduced causes, which indicates that they are contained in the explanation. Furthermore, we can see that not all the causes are applied to the example, which is what we expect if a good explanation was given. In addition to this, we see that the causes selected are consistent with the rank we made, and the most relevant cause is selected in most of the cases, having the highest percentage of selection followed by the second and then the third.

The percentage of generated examples present in the dataset is below 30% in all the experiments. This could be coincidences, since the dataset is large and there is not a lot of variability in the features which are usually changed like education level and profession. However, we cannot be certain about this and further experiments should be carried out with other less known data sets.

4.2 “German Credit Dataset” (Human Evaluation)

A second experiment is currently being carried out in order to evaluate understanding, satisfaction and cognitive workload of users as explained in Chapter 3. We

are using the “German Credit” dataset (Hofmann, 1994), which classifies people described by a set of attributes as good or bad credit risks, and selected different examples that were classified as bad credit risks. These examples are stratified by gender and age intervals. We generated textual explanations for each case as well as a table with the counterfactuals for it. The subjects of the experiment are people who are actually seeking a loan and they will be surveyed to evaluate the aspects mentioned for both methods of delivery.

We already saw on the previous experiment that the LLM can correctly identify, select and synthesize into textual explanation the causes deduced from a set of counterfactual examples, which provides us with a solid foundation and suggest promising directions of research. With this second experiment we would like to validate that LLM generated textual explanations are actually useful for the end user to better understand the decision of the ML model. We expect to extract valuable insights when this experiment is finished, which will help us to better understand the benefits and flaws of our method.

Chapter 5

Conclusions

In our work we present a method to produce automatic explanations of a set of counterfactuals using LLMs. We have seen that the LLM is capable of correctly extracting possible causes that will lead to a flip of the decision of a classifier by analyzing these counterfactual examples, select the main reasons according to their relevance and generate a final explanation in natural language that can be interpreted by the final user. We have also observed that increasing the amount of counterfactuals increases the amount of times the LLM is capable of generating a counterfactual of the opposite class using only the natural language explanation, indicating better information given. Furthermore, using larger sets of counterfactuals and prompting techniques like Tree of Thought can lead to more diversity of causes in the final explanation. Further experiments should be carried out to check the feasibility of this explanations when introducing causal knowledge.

Understanding, satisfaction and cognitive workload of users are yet to be evaluated in the case of presenting a natural language explanation versus a table of counterfactual examples, and we hope to draw some conclusions about this with our ongoing experiments. We will also wait for the feedback of the review process in order to get more insights and draw further conclusions, but overall, our method shows to be a promising option to complement sets of counterfactual examples in order to make them more understandable for the final user.

Appendix A

Trying out prompting strategies

LLMs are neural networks with a transformer architecture trained to do autoregression (predicting the next word after a sequence). They are trained over a large corpus of texts that include the Wikipedia, Internet scrapping, news articles, code and other sources. Furthermore, they are instruction-tuned in a second learning phase to output satisfactory responses to human instructions. Due to this nature, we can skillfully craft prompts or inputs to bias the behavior of AI models, particularly language models like GPT-4. This is called prompt engineering. The goal is to optimize the AI's response for accuracy, creativity, or specific outcomes.

In this section we try out different kinds of prompting techniques in the task of sentiment analysis to see if there are improvements in the results. Sentiment analysis is the process of determining the emotional tone behind a body of text. This is a common task in the fields of data analysis, customer feedback, and social media monitoring, where understanding public opinion or consumer sentiment is valuable.

To test the capabilities of GPT in the sentiment classification task, and how prompting affected performance, we used the *Trip Advisor Hotel Reviews* n.d. data set, which contains reviews of hotels and the rating given from 1-5. We considered the following classification:

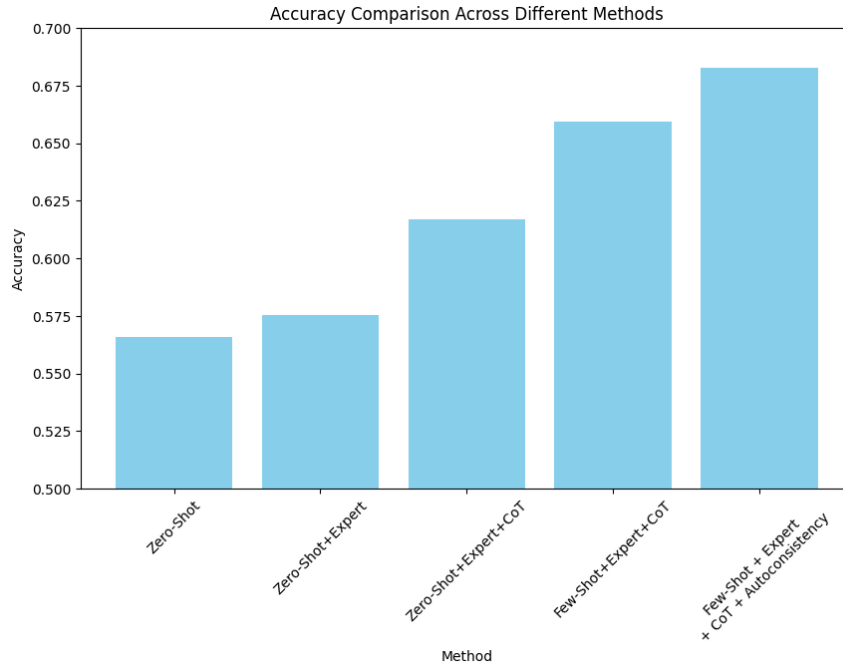
- Rating 1-2 → “negative”
- Rating 3 → “neutral”
- Rating 4-5 → “positive”

We prompted GPT 3.5 to classify 100 examples of each of the classes adding layers of complexity to the prompt in each experiment.

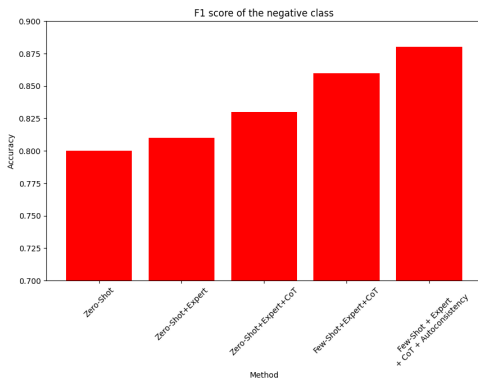
1. We start by using a Zero-Shot strategy in which we simply ask the LLM to classify the intention of the reviewer in ‘positive’, ‘negative’ or ‘neutral’, waiting for it to use its general understanding of language to infer the sentiment.
2. By default, GPT acts as a ‘helpful assistant’, so one can bias the responses of the model by defining its role as being an expert in the field of the particular task we want it to perform. In this case, we will tell him that it is a ‘expert hotelier’.
3. The following technique we added is "Chain of Thought" (CoT) (Wei et al., 2024). CoT consist in asking the model to perform the task step by step and "reason" all the way to the final result. In our case, the model will give reasons on why the review is positive or negative, biasing the final classification, since as we said, LLMs predict the next work given the previous input.

4. Few-shot (Brown et al., 2020) learning involves giving the model a small number of examples (or "shots") of the task at hand, from which it can quickly learn and adapt. We will give GPT three examples of text along with their sentiment classifications, in hopes of getting better outputs. The model then uses these examples to better understand and perform the sentiment analysis task. This approach can improve the model's accuracy on specific tasks without requiring extensive retraining.
5. Finally, we use self-consistency (Wang et al., 2023). We have mentioned that LLMs are probabilistic predictors of the next word. Therefore, there could be cases where probability of choosing one class or another are similar. A way to go around this problem is using self-consistency. This method consists on repeating the request several times and keeping the most common answer. In our case, we did 3 iterations.

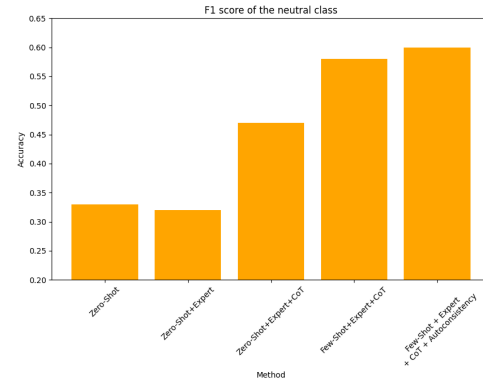
In Figure A.1 we present the results of the experiments in term of accuracy and F1 score of each of the classes. We can see how accuracy progressively increases each time we make the prompt more sophisticated, going from 0.56 when we used Zero-Shot up to 0.68 when using the most complex prompt. F1 scores also increase as we craft better prompts, specially the neutral cases which are the most ambiguous ones to classify.



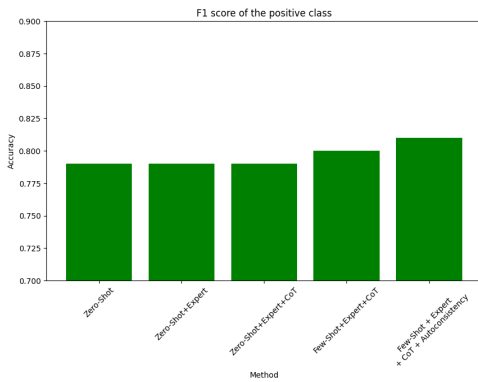
(A)



(B)



(C)



(D)

FIGURE A.1: Results of the experiment of classifying 300 reviews of trip advisor into positive neutral and negative by prompting GPT 3.5 with different strategies. (A) Accuracy (B) F1 score of the negative class (C) F1 score of the neutral class (D) F1 score of the postive class.

Appendix B

Introducing the Causal Graph in the Explanation Generation

Counterfactual examples generation methods based on perturbations do not assure feasibility in the examples, since they change features without taking into account the causal graph. We would like to check the feasibility of an example with prior knowledge of the causal graph. This part was finally left as future work, but we did do some experiments trying to obtain the causal graph by probing the LLM.

B.1 Using LLMs to obtain the causal graph

Building the causal graph is a complex process, and can be an overwhelming task to tackle. (Kıcıman et al., 2023) claim LLMs are proxies to human knowledge, and use them to do causal discovery between pairs of events, instead of statistical methods based on observed data. To obtain whether the causal relationship between events A and B is of the form $A \rightarrow B$, $A \leftarrow B$ or there is no relationship, we probed the LLM as seen in Figure B.1 (A). Furthermore, we used self-critiquing (Stechly, Valmeekam, and Kambhampati, 2024) and asked the LLM to check its own reasoning to improve accuracy of the results. In Figure B.1 (B) we can see this second prompt.

We tried to obtain the full graph applying this to all the possible pairs of nodes. In Figure B.2 (A) we can see the result of doing this process. Black arrows represent edges that were correctly reasoned and checked by the LLM, while the red dashed line represents a relationship that the LLM suggests should be there after checking and that it originally missed. Since this process is stochastic, results were different in each iteration. Therefore, we carried out the process 15 times giving value of +1 in one direction, -1 in the opposite direction and 0 if there was no relationship. We averaged over all the results and used a threshold of 1/3 in the absolute value in order to represent that edge. The result can be seen in Figure B.2 (B).

This approach is useful to present a first approximation of the graph, but it cannot discover the graph correctly since it neglects direct and indirect relationships. Graph discovery requires the ability to distinguish between direct and indirect causes, given the other nodes in the graph. For instance, if the actual relationship is $A \rightarrow B \rightarrow C$, it might be appropriate to output both $A \rightarrow B$ and $A \rightarrow C$ in a pairwise task. However, for graph discovery, outputting $A \rightarrow C$ would be incorrect. Additionally, the decision is influenced by the variables included in the input set for the graph discovery task. If variable B is not part of the input set, then $A \rightarrow C$ becomes a valid edge and should be included.

USER: First, lets define some variables:
 X: hours per week
 Y: workclass, type of industry (Government, Other/Unknown, Private, Self-Employed)
 Which cause-and-effect relationship is more likely?
 1. changing X causes a change in Y.
 -1. changing Y causes a change in X.
 0. there is no direct causal relationship between X and Y.
 Let's work this out in a step by step way to be sure that we have the right answer. Then provide your final within the tags
 <Answer>1/-1/0</Answer>.

GPT-4: The variable X (workclass) and Y (hours per week) could indeed have a relationship. A certain type of industry might require more working hours per week. However, it is less likely that changing the total work hours per week (Y) would directly cause a change in the type of industry (X). Therefore, it's more likely that changing X causes a change in Y.
 <Answer>1</Answer>.

(A)

USER: Analyze the output from an AI assistant. Is the final answer consistent with the reasoning provided by the assistant? Give a final answer after reasoning this issue.

Question: <PREVIOUS QUESTION>
 AI assistant: <PREVIOUS ANSWER>
 Explain your decision and then provide your final within the tags
 <Answer>1/0/-1</Answer>.

GPT-4: The AI assistant's reasoning is consistent with its final answer. The assistant explained that a change in the type of industry (X) could potentially require more working hours per week (Y). However, it is less likely that changing the number of work hours per week (Y) would result in a change in the type of industry (X). Therefore, the final answer of 1 (indicating that changing X causes a change in Y) is reasonable based on this reasoning.
 <Answer>1</Answer>.

(B)

FIGURE B.1: (A) Example of pairwise causal discovery prompt. (B) Example of self-critiquing prompt to obtain a final answer

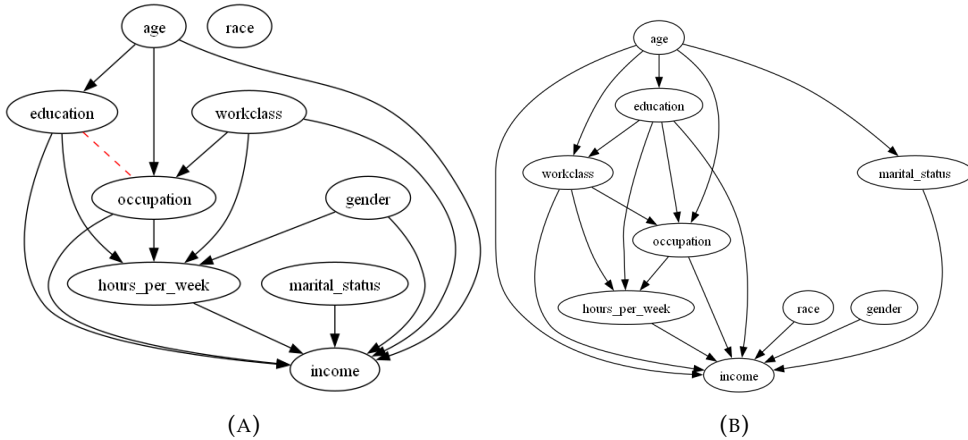


FIGURE B.2: (A) Example of a graph generated by prompting the LLM. The red line represents an edge that the LLM missed initially but was checked when self-critiquing. (B) Graph obtained by averaging over 15 different graphs.

Bibliography

- Becker, Barry and Ronny Kohavi (1996). *Adult*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5XW20>.
- Bove, Clara et al. (2023a). “Investigating the Intelligibility of Plural Counterfactual Examples for Non-Expert Users: an Explanation User Interface Proposition and User Study”. In: *Proceedings of the 28th International Conference on Intelligent User Interfaces*. IUI ’23. , Sydney, NSW, Australia, Association for Computing Machinery, 188–203. ISBN: 9798400701061. DOI: [10.1145/3581641.3584082](https://doi.org/10.1145/3581641.3584082). URL: <https://doi.org/10.1145/3581641.3584082>.
- (2023b). “Investigating the Intelligibility of Plural Counterfactual Examples for Non-Expert Users: an Explanation User Interface Proposition and User Study”. In: *Proceedings of the 28th International Conference on Intelligent User Interfaces*. IUI ’23. Sydney, NSW, Australia: Association for Computing Machinery, 188–203. ISBN: 9798400701061. DOI: [10.1145/3581641.3584082](https://doi.org/10.1145/3581641.3584082). URL: <https://doi.org/10.1145/3581641.3584082>.
- Braun, Virginia and Victoria Clarke (2019). “Reflecting on reflexive thematic analysis”. In: *Qualitative Research in Sport, Exercise and Health* 11.4, pp. 589–597. DOI: [10.1080/2159676X.2019.1628806](https://doi.org/10.1080/2159676X.2019.1628806). eprint: <https://doi.org/10.1080/2159676X.2019.1628806>. URL: <https://doi.org/10.1080/2159676X.2019.1628806>.
- Brown, Tom et al. (2020). “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- Cao, Alex et al. (2009). “NASA TLX: Software for Assessing Subjective Mental Workload”. In: *Behavior Research Methods* 41.1, pp. 113–117. DOI: [10.3758/BRM.41.1.113](https://doi.org/10.3758/BRM.41.1.113). URL: <https://doi.org/10.3758/BRM.41.1.113>.
- Doshi-Velez, Finale and Been Kim (2017). *Towards A Rigorous Science of Interpretable Machine Learning*. arXiv: [1702.08608](https://arxiv.org/abs/1702.08608).
- Gao, Luyu et al. (2023). “PAL: program-aided language models”. In: *Proceedings of the 40th International Conference on Machine Learning*. ICML’23. , Honolulu, Hawaii, USA, JMLR.org.
- Gat, Yair et al. (2023). *Faithful Explanations of Black-box NLP Models Using LLM-generated Counterfactuals*. arXiv: [2310.00603](https://arxiv.org/abs/2310.00603) [cs.CL].
- Guidotti, Riccardo et al. (2018). “A Survey of Methods for Explaining Black Box Models”. In: *ACM Comput. Surv.* 51.5. ISSN: 0360-0300. DOI: [10.1145/3236009](https://doi.org/10.1145/3236009). URL: <https://doi.org/10.1145/3236009>.
- Hofmann, Hans (1994). *Statlog (German Credit Data)*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5NC77>.
- Kopecka, Hana, Jose Such, and Michael Luck (2024). “Preferences for AI Explanations Based on Cognitive Style and Socio-Cultural Factors”. In: *Proc. ACM Hum.-Comput. Interact.* 8.CSCW1. DOI: [10.1145/3637386](https://doi.org/10.1145/3637386). URL: <https://doi.org/10.1145/3637386>.

- Kulesza, Alex (2012). “Determinantal Point Processes for Machine Learning”. In: *Foundations and Trends® in Machine Learning* 5.2–3, 123–286. ISSN: 1935-8245. DOI: [10.1561/22000000044](https://doi.org/10.1561/22000000044). URL: <http://dx.doi.org/10.1561/22000000044>.
- Kıcıman, Emre et al. (2023). *Causal Reasoning and Large Language Models: Opening a New Frontier for Causality*. arXiv: [2305.00050](https://arxiv.org/abs/2305.00050) [cs.AI].
- McCoy, R. Thomas et al. (2023). *Embers of Autoregression: Understanding Large Language Models Through the Problem They are Trained to Solve*. arXiv: [2309.13638](https://arxiv.org/abs/2309.13638) [cs.CL].
- Miller, Tim (2019). “Explanation in artificial intelligence: Insights from the social sciences”. In: *Artificial Intelligence* 267, pp. 1–38. ISSN: 0004-3702. DOI: <https://doi.org/10.1016/j.artint.2018.07.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0004370218305988>.
- Mohseni, Sina, Niloofar Zarei, and Eric D. Ragan (2021a). “A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems”. In: *ACM Trans. Interact. Intell. Syst.* 11.3–4. ISSN: 2160-6455. DOI: [10.1145/3387166](https://doi.org/10.1145/3387166). URL: <https://doi.org/10.1145/3387166>.
- (2021b). “A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems”. In: *ACM Trans. Interact. Intell. Syst.* 11.3–4. ISSN: 2160-6455. DOI: [10.1145/3387166](https://doi.org/10.1145/3387166). URL: <https://doi.org/10.1145/3387166>.
- Molnar, Christoph (2019). “Interpretable Machine Learning”. In: Chapter 9: Ethics and Fairness. Chap. 9. URL: <https://christophm.github.io/interpretable-ml-book/>.
- Mothilal, Ramaravind K, Amit Sharma, and Chenhao Tan (2020). “Explaining machine learning classifiers through diverse counterfactual explanations”. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 607–617.
- Neal, Brady (Dec. 2020). *Introduction to Causal Inference from a Machine Learning Perspective*. Accessed: 2024-03-20. Self-published. URL: <https://www.bradyneal.com/causal-inference-course>.
- OpenAI (2024). *GPT-4o*. Accessed: 2024-06-10. URL: <https://openai.com/index/hello-gpt-4o/>.
- Stechly, Kaya, Karthik Valmeekam, and Subbarao Kambhampati (2024). *On the Self-Verification Limitations of Large Language Models on Reasoning and Planning Tasks*. arXiv: [2402.08115](https://arxiv.org/abs/2402.08115) [cs.AI].
- Trip Advisor Hotel Reviews* (n.d.). Accessed: 2024-06-28. URL: <https://www.kaggle.com/datasets/andrewmvd/trip-advisor-hotel-reviews>.
- Vaswani, Ashish et al. (2017). “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
- Wang, Xuezhi et al. (2023). “Self-Consistency Improves Chain of Thought Reasoning in Language Models”. In: *ICLR 2023*. URL: <https://arxiv.org/abs/2203.11171>.
- Wang, Zilong et al. (2024). *Chain-of-Table: Evolving Tables in the Reasoning Chain for Table Understanding*. arXiv: [2401.04398](https://arxiv.org/abs/2401.04398) [cs.CL].
- Wei, Jason et al. (2024). “Chain-of-thought prompting elicits reasoning in large language models”. In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS ’22. , New Orleans, LA, USA, Curran Associates Inc. ISBN: 9781713871088.
- Yao, Shunyu et al. (2023). “Tree of Thoughts: Deliberate Problem Solving with Large Language Models”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., pp. 11809–11822. URL: <https://pro>

[ceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf).