UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Large Language Models and Causal Analysis: Zero-Shot Counterfactuals in Hate Speech Perception

Author:
Sergio HERNÁNDEZ

Supervisors:
Roger PROS
Jordi VITRIÀ

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

The code of this project is available at the following links:
GitHub: https://github.com/sergiohj93/Causal_NLP_TFM/tree/main
Google Drive: https://drive.google.com/drive/folders/1_KQ5MztWBpJ3RL7FBk5OY_nDFkx67sEp?usp=drive_link

June 30, 2024

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Large Language Models and Causal Analysis: Zero-Shot Counterfactuals in Hate Speech Perception

by Sergio HERNÁNDEZ

Detecting hate speech is crucial for maintaining the integrity of social media platforms, as it involves identifying content that denigrates individuals or groups based on their characteristics. However, the expression of hate can be different across different demographics and platforms, making its detection a complex task.

A significant factor in hate speech is the presence of offense, which alters the perception of hate without altering the core meaning of the text. This study aims to examine how offense affects the perception of hate speech in social media comments.

To achieve this, we employ two distinct causal inference methods to measure the impact of offensive language on the detection of hate speech. The first method utilizes the traditional backdoor criterion, which allows us to model the nodes of the causal graph as features in a machine learning model that predicts hate. This method is demanding from a modeling point of view, as it requires training a specific model for each node in the causal graph.

The second method leverages the capabilities of Large Language Models (LLMs) to generate textual counterfactuals in a zero-shot manner, i.e., without requiring any training or fine-tuning. These textual counterfactuals are then used to estimate causal effects.

Our findings reveal that the causal effect of offense on hate is higher with the LLM generated counterfactuals than with the methodology that follows the backdoor criterion. Additionally, we train a machine learning model to directly predict the causal effect from a comment.

Acknowledgements

I would like to thank my supervisor, Roger Pros, for his guidance, support and encouragement through my Master's thesis journey.

I am equally grateful to my co-supervisor, Jordi Vitrià, for his expert advice on critical aspects of the project.

I also thank my family and friends for their daily support.

Chapter 1

Introduction

1.1 Research objective

Social media sites have become global platforms where users can express and share their opinions freely. Unfortunately, some individuals exploit these spaces to share hateful content targeted toward individuals or groups based on attributes such as religion, gender or other characteristics, leading to the propagation of hate speech. Consequently, developing models to detect hate speech has become necessary to maintain the integrity of online platforms.

However, the hate-speech can differ a lot across demographics and platforms. Age, culture, and socio-economic factors can influence how hate is expressed, as well as the diverse forms of communication that each platform can have. All this makes hate detection a complex task.

A key aspect of hate speech is the presence of offense in the text. The offense of a sentence is what directly inflicts harm and promotes hostility towards specific individuals or groups. It is able of changing how hate is perceived in content without altering its core meaning.

This study aims to address the hate detection problem through a causal approach. Our objective is to investigate the impact of offensive language on the perception of hate speech in social media comments. To achieve this, we use different causal inference methods to quantify the effect of offensive language on the perception of hate speech.

We will limit ourselves to using machine learning models to estimate the probabilities of hate speech and of their underlying causes (which we identify in a causal graph). Specifically, we utilize deep learning models based on transformers (Islam et al., 2023).

Regarding the causal inference methods, we first use a traditional approach following the backdoor criterion (Pearl, 2009). It allows us to model the nodes of the causal graph as features in a machine learning model that predicts hate.

Moving forward, we will test a more contemporary approach that leverages the capabilities of Large Language Models (LLMs) (Bommasani et al., 2021) to generate textual counterfactuals (Lewis, 2013). These textual counterfactuals are then used to quantify causal effects.

1.2 Related work

There exist recent studies that combine the use of machine learning models or LLMs with causality.

In (Sheth et al., 2023), Sheth et al. propose a causality-guided framework for hate speech detection in order to adress the fact that hate-speech can differ a lot across distinct demographics and platforms. They took advantage of inherent causal cues, which can be leveraged to learn generalizable representations for detecting hate speech across different distribution shifts.

They employ roBERTa models (Liu et al., 2019) capable to predict these causal cues in order to integrate them into another roBERTa model for hate detection. To identify the causal cues, they utilize the ones verified by various studies in social sciences and psychology that can aid in detecting hate (Bauwelinck and Lefever, 2019; Craig, 2002; Krahé, 2020; Sengupta et al., 2022; Zhou et al., 2021). Some of these cues are the hater's prior history, the conversational thread, the overall sentiment and the offense in the text.

However, they only have access to the sentiment and the offense. In our study, we construct the causal graph of our problem in function of these causal cues (see Section 2.1.1). Furthermore, we utilize a simplified version of PEACE model, by adapting it to causal analysis (see Section 2.2.1).

Concerning large language models, in their study (Kıcıman et al., 2023), Kıcıman et al. analyze the causal capabilities of LLMs across different causal tasks. They find that LLMs are useful in capturing common sense and domain knowledge about causal mechanisms, as well to facilitate the translation between natural language and formal methods.

Similar to our approach, Gat et al.(Gat et al., 2023) generate textual counterfactuals with an LLM to estimate causal effects, achieving a good performance. The counterfactuals are created to change the value of a concept from a textual restaurant review, while maintaining the value of the other concepts (the concepts are food, service, ambiance and noise).

In (Li et al., 2023), Li et al. utilize an LLM (GPT-3.5 (OpenAI, 2023a)) to generate counterfactuals in order to perform data augmentation in different Natural Language Understanding (NLU) tasks (Chowdhary and Chowdhary, 2020). Their evaluation covers several factors to determine LLMs capability of generating counterfactuals, including prompt design and intrinsic properties of LLMs such as the model size. Their findings indicate that LLMs can produce satisfactory counterfactuals in most cases, though they also have found LLMs weaknesses when dealing with complex tasks like Relation Extraction (RE) (Nasar, Jaffry, and Malik, 2021).

Chapter 2

Methodology

2.1 Problem definition

2.1.1 Causal Graph

First of all, we use a causal graph (Pearl et al., 2000) to identify our accessible causal relationships that influence hate in comments. As shown in Figure 2.1, we consider that the hate depends on the offense, the sentiment and the meaning of the comment. The offense and the sentiment also depends on the meaning.

We need to keep in mind that there are inaccessible factors (such as the hater's prior history and the conversational thread) that can affect the hate as well.

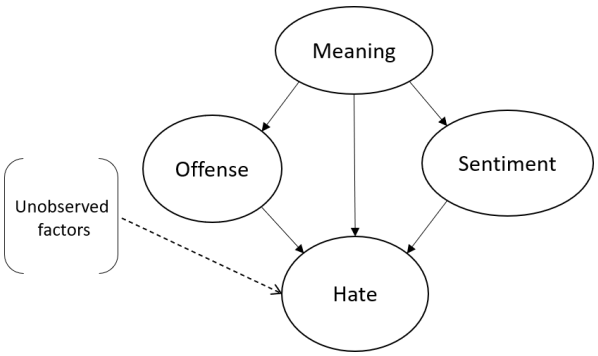


FIGURE 2.1: Causal Graph to identify causes of Hate.

Table 2.1 provides information about the variables in the causal graph nodes. In our causal inference approaches, we will obtain the offense, sentiment and meaning as embeddings produced by processing the texts with deep learning models. However, when necessary, offense and sentiment can be expressed on a discrete scale through classification, unlike the meaning.

TABLE 2.1: Variables of the Causal Graph

Variable	Type	Values	Causal role
Sentiment	Ordinal	Negative, Neutral, Positive	Covariate
Meaning	Embeddings	N/A	Covariate
Offense	Binary	Non-Offensive, Offensive	Treatment
Hate	Binary	Non-Hate, Hate	Outcome

2.1.2 Causal Inference

We aim to analyze how offense affects the perception of hate speech. To that end, we use a causal inference (Pearl, 2009) approach to quantify the effect of offensive language on the perception of hate. The causal role of each variable in our causal inference scenario appears in Table 2.1.

Our key causal metrics to quantify the effect of the treatment on the outcome are the following:

$$\text{ITE} = p(H_i|\text{Off}_i = 1, S_i, M_i) - p(H_i|\text{Off}_i = 0, S_i, M_i) \tag{2.1}$$

$$\text{ATE} = \frac{1}{n}\sum_{i=1}^{n}(p(H_i|\text{Off}_i = 1, S_i, M_i) - p(H_i|\text{Off}_i = 0, S_i, M_i)) \tag{2.2}$$

In these equations, 'H' designs the hate, 'Off' the offense, 'S' the sentiment, 'M' the meaning and 'n' the total number of evaluated comments.

By using the Individual Treatment Effect (ITE) (Eq. 2.1), we assess the difference in hate probability for each comment based on the presence or absence of offense.The Average Treatment Effect (ATE) (Eq. 2.2) is then calculated by averaging these ITEs. However, we can only observe one of the two offense states for a given comment. To estimate the hate probability for the unobserved state, we need to intervene on the offense variable.

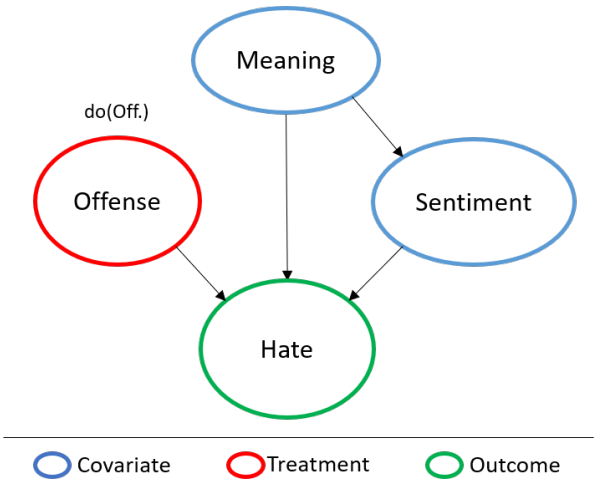


FIGURE 2.2: Causal Graph after the intervention on Offense. It also shows the causal role of each node

Figure 2.2 shows the causal graph after the intervention. As we can observe, the link from meaning to offense has disappeared because we are fixing the offense value and it doesn't depend anymore on the meaning.

Figure 2.3 illustrates the general methodology that we follow to perform causal inference. First, we intervene on the comment's offense to generate a new representation with its offense value altered. Next, we obtain the hate of these representations, to further calculate the causal metrics.

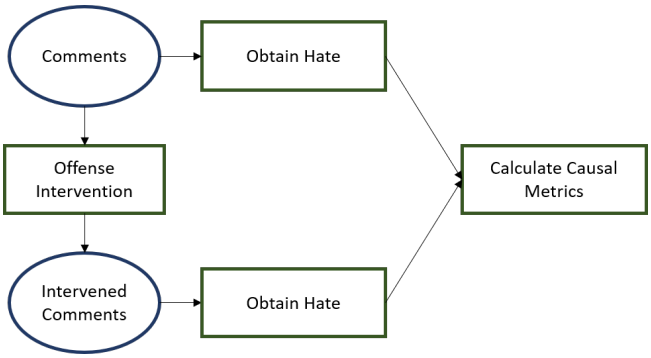


FIGURE 2.3: Methodology that we use to perform causal inference.

We employ two different causal inference methods which follow the described methodology. Each method has a distinct way of intervening on the offense.

The first method follows the traditional backdoor criterion (Pearl, 2009), and, by using a S-Learner (Künzel et al., 2019), will allow us to model the covariates and treatment as features in a machine learning or deep learning model that predicts hate. In this approach, the offense is represented by a single binary feature, so, we can intervene it by simply changing its value.

The second method leverages the capabilities of large language models (LLMs) (Bommasani et al., 2021) to generate textual counterfactuals. In this method, it's the LLM who intervenes on the offense by revising the text. The original and revised texts are then individually processed through a deep learning model to obtain its hate speech probability.

2.2 Backdoor Criterion

The Backdoor Criterion allows us to identify a set of variables Z that, when conditioned on, will block all the confounding paths, thus isolating the causal effect of X on Y. This makes it possible to estimate the causal effect by adjusting for Z. It's important to block confounding paths as far as they introduce bias and distortions into the relationship between the treatment and the outcome variables.

In this problem, X is the offense, Y is the hate and the adjusted concepts Z are the sentiment and the meaning.

A set of variables Z satisfies the backdoor criterion relative to (X,Y) in a directed acyclic graph G if:

1. No node in Z is a descendent of X.
2. Z blocks every path between X and Y that contains an arrow into X.

As we can see in Figure 2.2, these conditions are satisfied. Therefore, we can estimate the causal effect of offense on hate by using the variables in the causal graph.

2.2.1 S-Learner

We are going to use the S-Learner (Single-Learner) meta-algorithm (Künzel et al., 2019). The S-Learner approach integrates the treatment effect estimation directly

into a single predictive model. This is done by including the treatment variable as an additional feature along with the covariates. Then, we can use a machine learning model to estimate $p(H_i|Off_i = 1, S_i, M_i)$ and $p(H_i|Off_i = 0, S_i, M_i)$ by changing the offense binary value. To subsequently calculate the ITE (Eq. 2.1) and the ATE (Eq. 2.2).

We use an architecture that is partially based on the PEACE model (Sheth et al., 2023) (this model is explained in Section 1.2). Unlike PEACE, our goal is not to achieve the best predictive performance, but to conduct a causal analysis. Therefore, we adopt its causal architecture while omitting elements like the integration of causal modules using attention vectors.

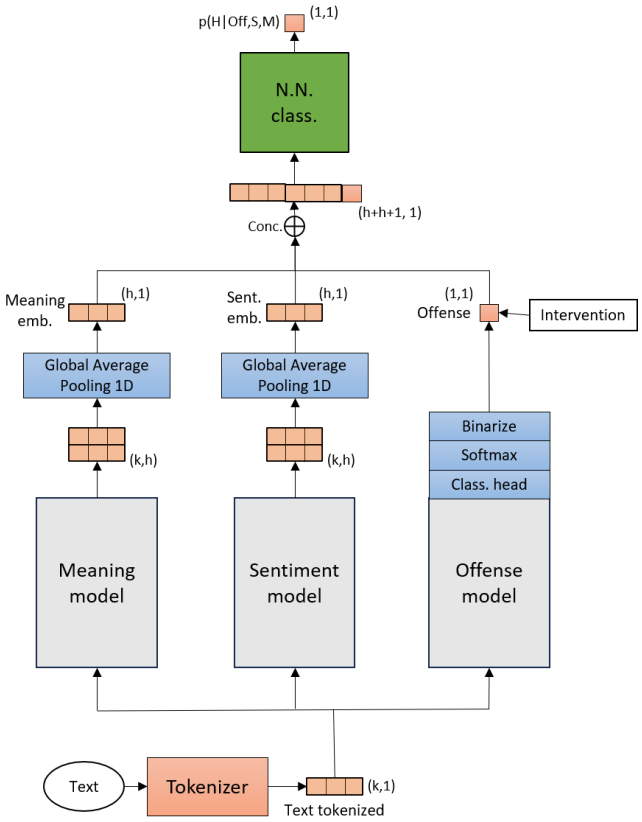


FIGURE 2.4: Our simplified PEACE model. 'k' indicates the length of the tokenized text and 'h' the size of the embeddings produced by the last hidden layer of the RoBERTa models for each token.

Figure 2.4 shows our simplified PEACE model. We utilize roBERTa models (Liu et al., 2019) to process the comments and obtain their values of sentiment, meaning and offense.

The sentiment model (Rosenthal, Farra, and Nakov, 2017) is a roBERTa-base model (Liu et al., 2019) trained on approximately 124M English tweets and fine-tuned for sentiment analysis to detect the sentiment values: 'negative', 'neutral' or 'positive'. The offense model (Zampieri et al., 2019) is another roBERTa-base model trained on around 58M English tweets and fine-tuned for offensive language identification to detect the values: 'non-offensive' or 'offensive'. Finally, the meaning

model is a roBERTa-base model without any additional training.

For the sentiment and meaning models, we extract the embeddings from the last hidden layer and apply a global average pooling operation (TensorFlow, 2024), discarding the classification head of the sentiment model. For the offense model, we obtain the binarized offense value from its classification head. We then concatenate these embeddings and the offense value to train a neural network classifier with two layers of 128 neurons for hate-speech detection.

2.2.2 TARNet

In our PEACE architecture, the neural network classifier receives the meaning and sentiment embeddings, each with a shape of (768,1), and a single binary value for the offense. Consequently, we are probably suffering the curse of dimensionality (Hastie et al., 2009), and the offense could have a minimal repercussion on the hate detection, making it challenging to estimate its causal impact.

To face this problem, we are also going to use a TARNet instead of the former neural network classifier. TARNet (Treatment-Agnostic Representation Network) (Shalit, Johansson, and Sontag, 2017) consists of two main components: a shared representation network and two separate heads for predicting outcomes for each value of the binary treatment.

The shared network learns a representation of the covariates independent of the treatment, which is then used by the treatment-specific heads to estimate outcomes.
During training, the network aims to minimize a loss function based on the factual outcomes observed in the data. Specifically, the loss is calculated using the head corresponding to the observed value of the treatment.

Then, by separating the representation of the covariates from the treatment effect estimation, TARNet ensures that the treatment value is considered despite the high dimensionality of the covariates.

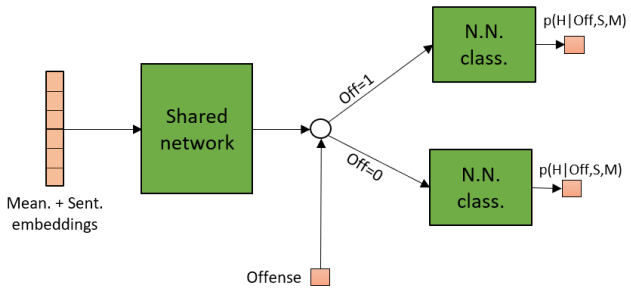


FIGURE 2.5: TARNet architecture.

Figure 2.5 shows the architecture of our TARNet. The shared network is a neural network composed of three layers, each with 200 neurons. Each treatment-specific head is a neural network with two layers, containing 100 neurons per layer.

2.3 Counterfactuals generated by an LLM

With this second causal inference method, we are going to intervene the offense by the creation of textual counterfactuals. Counterfactuals (Lewis, 2013) are hypothetical scenarios used to reason about causality by considering what would have happened under different circumstances. In other words, they allow us to explore the effects of changing a treatment variable while keeping everything else constant.

We achieve this more directly by using the backdoor criterion based method described in Section 2.2, where we intervene on the Offense at the feature level by modifying a single binary feature. However, we now aim to perform this intervention by revising the text of the comment. For this purpose, we leverage a Large Language Model (GPT-3.5 (Generative Pre-trained Transformer 3.5) (OpenAI, 2023a)) to generate textual counterfactuals.

Thanks to having been trained with large amounts of natural language data to learn language patterns, LLM models like GPT-3.5 can understand requests and answer with meaningful responses. We will leverage this capability in order to generate textual counterfactuals in a zero-shot manner, i.e., by simply prompting the model to do so and without having to train it specifically for this purpose. This approach is less demanding from a modeling perspective compared to the backdoor criterion method, which requires training a specific model for each node in the causal graph.

However, the quality of this LLM generated counterfactuals will depend on GPT-3.5's ability to modify the offense of the comment while keeping everything else unchanged just through text revision. This task appears to be more challenging for this method compared to the backdoor criterion, where, as previously mentioned, we can directly intervene on the offense just by modifying a binary feature.

2.3.1 Prompt design

We ask GPT-3.5 to generate counterfactuals from each comment of the inference dataset (explained in Section 3.1), depending on its binary offense value. These offense values are obtained by means of the roBERTa model for offensive language identification (Zampieri et al., 2019) (We have already obtained them in Section 2.2.1). If the comment is Offensive we ask the LLM to make it Non-Offensive, and vice versa. To achieve this, we need, for each one of these two tasks, to prompt the model.

We adopt some prompt engineering (Marvin et al., 2023) tactics from OpenAI documentation (OpenAI, 2023b), also used in recent papers for counterfactual generation (Gat et al., 2023; Li et al., 2024; Bhattacharjee et al., 2024; Kıcıman et al., 2023).

GPT-3.5 allows to define different types of messages to create the prompt. On the one hand, the system message helps set the behavior of the model by providing specific instructions. On the other hand, the user messages provide requests for the assistant to respond to.

First of all, we need to clearly indicate what we want. As a system message, we instruct the model on how to modify the treatment (the offense) without changing anything else, by asking it to increase or decrease the offensiveness without altering

the meaning of the sentence. With this, our aim is to preserve the sentiment, as well as other concepts included within the meaning. As a prompt engineering tactic, we include four examples in the system message to help GPT-3.5 understand what we want. This is known as 'few-shot' prompting.

In the user message, we ask the model to create the counterfactual using the same format as the examples. We also use delimiters to clearly indicate distinct parts of the input, such as XML tags to separate the examples, an emphasize important words for the task, such as the value of the treatment.

We have defined two prompts, each one with a different system message. One to make Offensive a comment (system message 0) and the other to make it Non-Offensive (system message 1). In Figure 2.6 we show the prompt to make Offensive a given Non-Offensive comment. The other prompt is similar and we can find it in Appendix A.1.

"SYSTEM message":

```
I will give you a sentence that I consider NON-OFFENSIVE. You must make it OFFENSIVE by
INCREASING its OFFENSIVENES without changing its MEANING. Here are a few examples of what I
would like:

<Example 1> Sentence: "Trust me, Mr Jackson, with this attitude, it is not ""your"" country."

Revised Sentence: "Listen up, Mr. Jackson, with that appalling attitude of yours, this country
couldn't be further from belonging to someone like you." </Example 1>

<Example 2> Sentence: "I do need help - help explaining to religious fanatics, that we live in
2014"

Revised Sentence: "I absolutely require assistance - assistance to elucidate to those ignorant
religious zealots, stuck in their archaic beliefs, that we are indeed living in the year 2014,
not the Dark Ages." </Example 2>

<Example 3> Sentence:  "John, you obviously don't understand our laws or our Constitution."

Revised Sentence: "John, it's painfully evident that your pea-sized brain can't understand our
laws or our Constitution." </Example 3>

<Example 4> Sentence: "We are ALL tired of waiting . . realize something"

Revised Sentence: "We're ALL damn sick and tired of waiting. Pull your head out of your rear
end and realize something!" </Example 4>
```

"USER message":

```
Sentence: <Input sentence>

Revised Sentence:
```

FIGURE 2.6: Prompt messages to make a comment Offensive.

Figure 2.7 presents a diagram illustrating the counterfactual generation methodology.

2.3.2 Evaluation of the counterfactuals

In Section 3.2 we will test the quality of the generated counterfactuals by comparing them to the original comments (we call these samples 'factuals'). We want to evaluate if the LLM is able to modify the offense without significantly altering the sentiment and the meaning.

To achieve this, we utilize the roBERTa models from (Zampieri et al., 2019; Rosenthal, Farra, and Nakov, 2017) (previously used in Section 2.2.1) to predict the offense

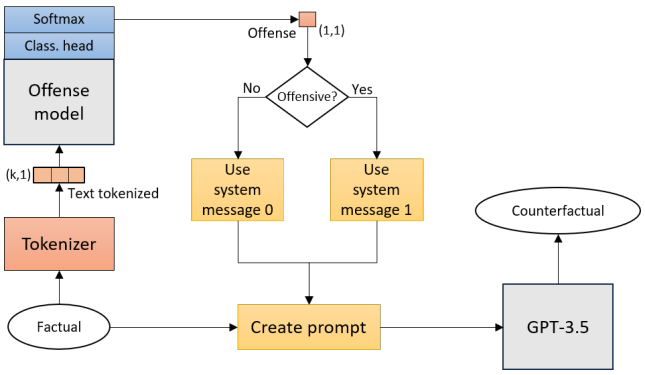


FIGURE 2.7: Diagram of the counterfactual generation methodology.

and sentiment values for each pair of factual/counterfactual and compare them. Additionally, we will use another deep learning model to compare the semantic similarity.

Finally, when calculating the causal metrics, we will also compare the ones from this approach to the obtained with the backdoor criterion method.

2.3.3 Hate models

To obtain the hate probability from the comments and the intervened comments, the current LLM counterfactual approach does not limit us to an architecture that allows the modification of the treatment value. Therefore, we can use other hate prediction models that don't require training a specific model for each node in the causal graph. We also use our simplified PEACE model, for comparison purposes. We test three models.

Regarding the PEACE model, predictions from the offense module are no longer required. Then, we can discard this offense module and use only the globally averaged embeddings of the covariates to train the neural network classifier. No further modification is required. The new architecture is shown in Figure 2.8.

The second hate model is HateBERT (Caselli et al., 2021). It is a re-trained BERT model (Devlin et al., 2018) for abusive language detection obtained by further training the English BERT base uncased model (Devlin et al., 2018) with more than 1 million posts from banned Reddit communities.

It hasn't been trained to directly detect hate speech, but rather with a Masked Language Model objective (Salazar et al., 2019). Therefore, we extract the globally averaged embeddings from the model to further train a neural network classifier with a few layers (in the same way as we do with our PEACE model). The model usage is shown in Figure 2.9a.

The last model (Antypas and Camacho-Collados, 2023) is a roBERTa-base model (Liu et al., 2019) trained on approximately 58M tweets and fine-tuned for binary hate-speech classification on a combination of 13 different hate-speech datasets in the English language.

Therefore, we can use it without requiring any additional training. We will test it in this manner but, also, by extracting the globally averaged embeddings from the last hidden layer of this roBERTa-base model (discarding its classification head) to

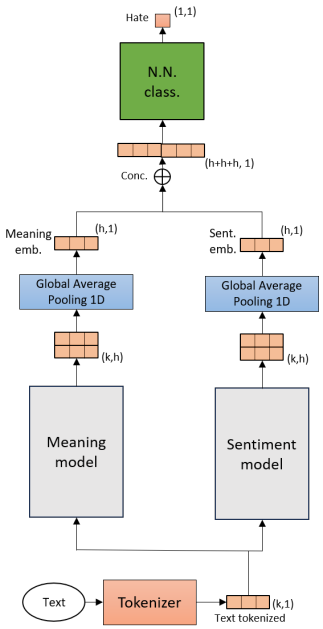
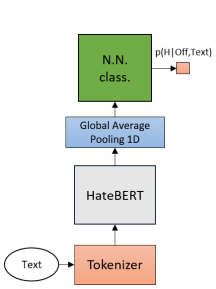
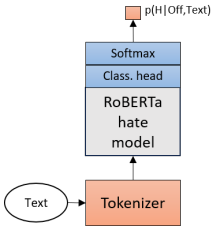


FIGURE 2.8: Architecture of the PEACE model that we use to evaluate the LLM generated counterfactuals. 'k' indicates the length of the tokenized text and 'h' the size of the embeddings produced by the last hidden layer of the RoBERTa models for each token.

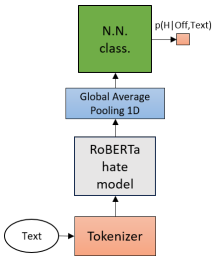
further train a neural network classifier with a few layers. Again, for comparison purposes. The model usage is shown in Figure 2.9b and 2.9c.



(A) HateBERT



(B) RoBERTa hate



(C) RoBERTa hate with our classifier

FIGURE 2.9: Architectures to use the HateBERT and the RoBERTa hate models.

Chapter 3

Experiments

3.1 Datasets

3.1.1 Original datasets

We construct our dataset with data from two widely used hate benchmark datasets which contain user comments from different platforms: Facebook, Twitter, Reddit and Youtube.

The first one is the FRENK dataset (Ljubešić, Fišer, and Erjavec, 2021). It contains comments to Facebook posts on the topics of migrants and LGBT. They can be written in Croatian, English or Slovenian, but we are only interested in the English subset (Ljubešić, Fišer, and Erjavec, 2019), which consists of 10,705 data samples.

Each comment is annotated by the topic and its target (as ordinal variables) and by a binary hate speech variable (which indicates if the comment can be considered hate speech or not). We only take the text of the comment and the binary hate speech variable.

The second hate dataset is Twitter-Reddit-Youtube (Kennedy et al., 2020). It consists of a collection of comments in English from those three platforms annotated by 7,912 annotators. It has more than 100 variables, but the "hate speech score" is the primary outcome. It is a continuous score created from the combination of 10 ordinal labels (sentiment, (dis)respect, insult, humiliation,violence,etc...).

We binarize this data such that any comment with a hate speech score less than 0.5 is considered non-hateful, and vice versa. As with the FRENK dataset, we take only the text and the binary hate, from 10,705 samples.

3.1.2 Our datasets

As far as we now have the same variables from both datasets (the comment as a string and the binary hate-speech variable), we can simply concatenate them, obtaining 21,410 rows. In Table 3.1 we show a few samples with short texts.

TABLE 3.1: A few samples from the dataset

Text	Label
"This is not just a migration, it's planned Islamic invasion."	1
"Oh don't worry. We are all Summers here!"	0
"i almost puke seeing this transvestite"	1

TABLE 3.2: A few discarded texts from the inference dataset.

"Penny Baker"
"....."
"Mmmm"
"http://www.abc.net.au/news/2012-03-06/badgett-good-things-about-gay-marriage/3870750"

The length of the comments ranges from 1 to 7322 characters. But, as we can see in the bar chart from Figure 3.1, most have fewer than 200 characters. After this value, the length starts to decrease very quickly, to the point where very few texts have more than 600 characters.

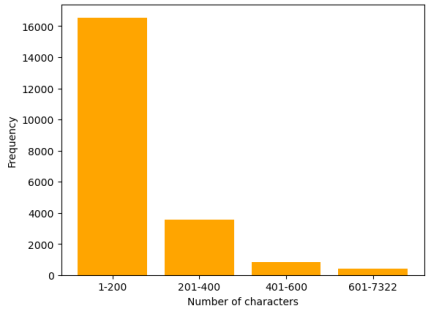


FIGURE 3.1: Frequency of the number of characters of the comments.

We split our data into two datasets. The first dataset will be used for causal inference and analysis, i.e., to obtain the offensiveness of the comment, intervene it and to predict the hate, to later calculate the causal metrics. With the second dataset we will train the models that predict hate.

Initially, 4,602 samples go to the inference set and 16,808 to the training set.

Inference set (Factuals)

The texts from the inference set are the ones that will be used to generate the LLM counterfactuals. Then, to make the task easier for the LLM, we are going to filter the texts.

We remove too short and too long texts. In general, the too short comments of our data don't have so much meaning, finding cases in which only appear the name of a person or just one word such as "hi", "Lol", or "No". And, as we have already seen in Figure3.1, most texts have less than 200 characters. Then, we discard comments with less than 20 characters (338 texts) and with more than 200 characters (1137 texts).

We can also find comments between those length limits that only have an URL. It may be that, with this kind of texts, GPT-3.5 cannot be able to create a counterfactual. As far as, after the length filtering, the number of texts which contains and URL is just 70, we can discard all of them. In Table 3.2 we show a few discarded samples.

We end up with 3057 samples in the inference set, which will be a good number for us when creating the LLM counterfactuals, due to our limited resources. As we can see in the histogram from Figure 3.2, the number of characters frequency also

tends to decrease in this range between 20 and 200 characters.

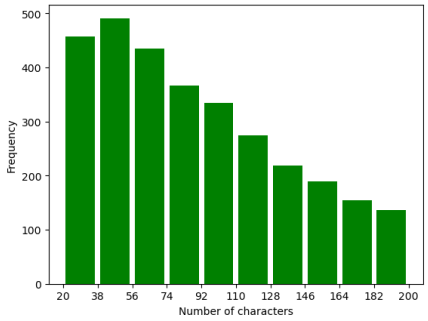


FIGURE 3.2: Frequency of the number of characters of the comments from the inference set.

Hate training set

We don't feel necessary to make any filtering to the set for training the hate models, unlike with the inference set. We want to consider all the comments to train a more robust model, and we also think that texts like the previously discarded can be now useful. For instance, the model can be able to predict short texts better. With this idea, we add to the training set the texts discarded from the inference set, obtaining 18,353.

Now, we are going to consider the class distribution of the two datasets. As we can see in Figure 3.3, we find a very similar distribution in both datasets, with a higher number of non-hate labels. Therefore, we are in an imbalanced classes scenario (Haixiang et al., 2017) and we will take this into account when training the model and when evaluating its prediction.

(A) Inference set

(B) Hate training set

FIGURE 3.3: Class distributions of the inference and training sets.

A summary of the datasets can be found in Table 3.3.

3.2 Experiment 1: Evaluation of the LLM counterfactuals

In this first experiment, our objective is to assess the quality of the LLM generated counterfactuals, by comparing them to the factual instances. We test if the LLM can

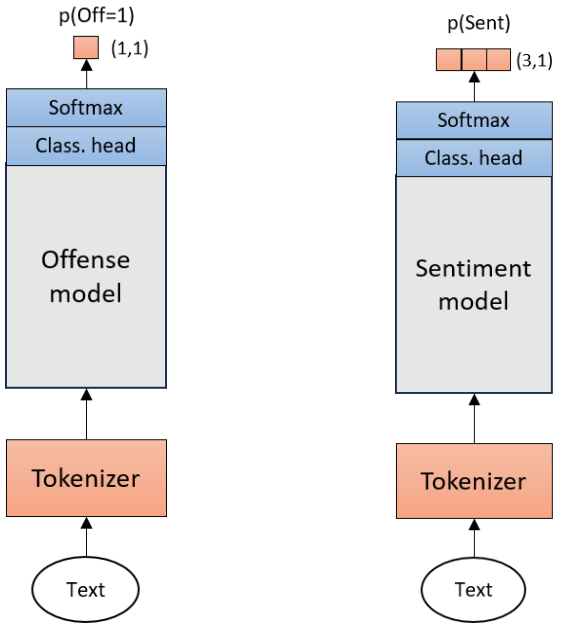
TABLE 3.3: Datasets summary.

Dataset	Number of comments	Hateful comments
Inference	3,057	986
Hate training	18,353	5,769

modify the offense without significantly altering the sentiment and the meaning.

3.2.1 Offense and Sentiment comparison

To predict the offense and sentiment for each factual and counterfactual, we use the roBERTa models from (Zampieri et al., 2019; Rosenthal, Farra, and Nakov, 2017) (previously used in Section 2.2.1). On the one hand, as illustrated in Figure 3.4a, by processing a text through the offense model, we obtain the probability of being offensive. On the other hand, as shown in Figure 3.4b, processing the text through the sentiment model provides the probability for each one of the possible values (negative, neutral or positive) as a vector.



(A) Offense model (B) Sentiment model

FIGURE 3.4: Usage of the offense and sentiment models to predict the probabilities.

For each factual and counterfactual pair, we calculate the offense distance $p(\text{cf\_Off} = 1) - p(\text{fact\_Off} = 1)$ if the factual is non-offensive and the offense distance $p(\text{fact\_Off} = 1) - p(\text{cf\_Off} = 1)$ if the factual is offensive. Then, we average these distances.

Regarding the sentiment, we calculate the euclidean distance between the sentiment probability vectors of each factual and counterfactual pair, and then we average these distances.

Additionally, we discretize both the offense and the sentiment. Then, we count the number of factual and counterfactual pairs where the binarized offense is different between the factual and the counterfactual. This count is then divided by the total number of pairs to compute the accuracy.

We compute another accuracy by counting the number of pairs where the LLM is able to maintain the same discretized sentiment value between the factual and the counterfactual.

As an additional metric, we calculate the accuracy of offense improvement. This measures whether the LLM increases the offensive probability when the factual is non-offensive and decreases it when the factual is offensive.

3.2.2 Semantic similarity

To compare the meaning of the factual and counterfactual texts, we could use embeddings generated by a roBERTa model (Liu et al., 2019). However, roBERTa primarily produces embeddings for individual tokens rather than entire sentences or texts. While it is possible to obtain a single vector by averaging token embeddings (like we do in Section 2.2.1), this kind of approaches often fail to effectively capture the full semantic meaning of the sentence (Reimers and Gurevych, 2019).

Therefore, we use the 'all-MiniLM-L6-v2' model from the Sentence Transformers library to obtain the embeddings (Reimers and Gurevych, 2020). This model maps texts to a 384 dimensional dense vector space and can be used for tasks such as semantic similarity (Agirre et al., 2012).

We compare the embeddings of the factual to the counterfactual ones by the use of the semantic similarity. Specifically, we calculate the cosine similarity (Salton, Wong, and Yang, 1975), i.e., the cosine of the angle between the two vectors, by using Eq. 3.1.

$$\text{Cosine similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{3.1}$$

This value measures the similarity focusing on the vector direction in the embedding space rather than their absolute magnitude. This is crucial because the direction often captures semantic meaning more effectively than absolute distances. (Reimers and Gurevych, 2019). As a metric, we average the semantic similarities of all the factual and counterfactual pairs.

3.2.3 Results

In the call to the OpenAI API, we ask GPT-3.5 to generate 5 responses for each request. Consequently, for each of the 3,057 factual instances in the inference dataset, 5 corresponding counterfactuals are generated, resulting in a total of 15,285 factual-counterfactual pairs. Five factual-counterfactual pairs corresponding to the same factual are shown in Table 3.4. This factual is considered offensive by the offense model, then, the GPT-3.5 task was to make it non-offensive.

TABLE 3.4: Factual-counterfactual pairs corresponding to the same factual.

Factual	Counterfactual
“This is a really stupid comment to be fair”	"This comment seems to lack intelligence, to be honest."
“This is a really stupid comment to be fair”	"This comment seems rather unfounded, to be honest."
“This is a really stupid comment to be fair”	"This comment seems a bit uninformed, to be honest."
“This is a really stupid comment to be fair”	"This is not a very insightful comment, to be honest."
“This is a really stupid comment to be fair”	"This comment isn’t very thoughtful, to be honest."

TABLE 3.5: Metrics to compare the factuals to the LLM generated counterfactuals.

Metric	Value
Average of offense distances	0.39
Average of sentiment distances	0.44
Accuracy of distinct offense	0.71
Accuracy of same sentiment	0.61
Accuracy of offense improvement	0.94
Average of semantic similarities	0.68

For each factual-counterfactual pair, we obtain the sentiment and offense probabilities, as well as the semantic embeddings. To next calculate the metrics, as described in the previous two sections. Table 3.5 shows the metrics.

We find that the average offense distance effectively modifies the binary offense in 71% of cases. Additionally, GPT-3.5 achieves a high accuracy of 0.94 in successfully increasing or decreasing offensiveness. Concerning sentiment, the average sentiment distance suggests that sentiment is altered to some extent when the LLM attempts to modify the offense, although the discretized sentiment remains preserved in 61% of the cases, which is not that bad.

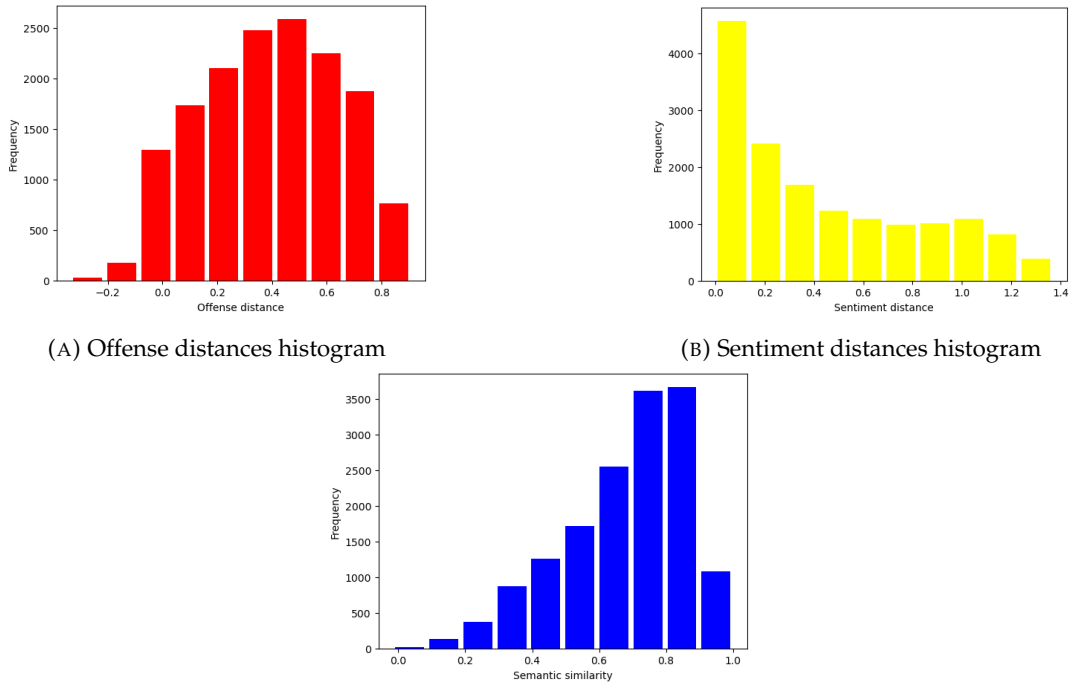
However, we find the average semantic similarity of around 0.7 reasonably satisfactory in order to assess if the LLM can modify the offense without altering too much the semantic meaning.

Figure 3.5 presents histograms of offense and sentiment distances, as well as semantic similarities. The offense distances present the higher frequencies around 0.5, with the frequencies decreasing for both lower and higher values. It looks promising, although there are also many samples close to a distance of 0.

Regarding sentiment distances, there are a lot of samples near 0, with frequency gradually decreasing as the distance increases. It’s a positive outcome to have too many samples near 0.

For semantic similarity, there are few samples near 0, and frequencies gradually increase up to about 0.9, after which they drop significantly. This distribution is favorable, as we have higher frequencies near a high similarity value.

In summary, the LLM seems capable of modifying the offensiveness of comments, although the sentiment and semantic meaning are being altered to some extent. Nevertheless, the higher frequencies near low sentiment distances and near high similarity values are a positive result.



(A) Offense distances histogram

(B) Sentiment distances histogram

(C) Semantic similarities histogram

FIGURE 3.5: Histograms of offense and sentiment distances, as well as semantic similarities.

3.3 Experiment 2: Causal inference analysis

In this experiment, we perform causal inference and analyze the resulting metrics for the tested methods, aiming to compare them. We consider the various architectures used in each method to estimate the probability of hate speech. For both causal inference methodologies, we follow the general steps from Figure 2.3.

For evaluating the hate speech predictions made by the different models, we calculate the accuracy and F1-score (Powers, 2020). While our goal is not to achieve the best predictive performance, these metrics serve as useful benchmarks, for example, for the parameter tunings that we make. The F1-score is chosen to address the class imbalance issue found in Section 2.2. Additionally, we train with class weights (Scikit-Learn, 2024).

3.3.1 Settings

Backdoor criterion methods

First, we follow the methodology explained in Section 2.2 to calculate the ITE (Eq. 2.1 and ATE (Eq. 2.2) by processing the texts from the inference dataset. For both PEACE 2.4 and TARNet 2.5 models.

The neural network classifier of the PEACE model consists of two layers, each one with 128 neurons. Regarding the TARNet, the shared network is a neural network with three layers, each containing 200 neurons. Each treatment-specific head

consists of a neural network with two layers, each having 100 neurons.

We train both the neural network of PEACE and the TARNet model using the 'hate training' dataset. For the PEACE neural network, we employ the Adam optimizer, a learning rate of 0.001, a batch size of 32 and we train for 10 epochs. The neural network architecture and training parameters have been selected after a parameter tuning.

Given the variability observed in the metrics across different training runs of the neural network model, we decided to split off a 10% validation set from the 'hate training' dataset. This validation set is used to select the model from the training run that achieves the best F1-score. We will apply this approach to all the models that predict hate. For the PEACE neural network, we performed 20 training runs.

Regarding TARNet training parameters, we utilize the Stochastic Gradient Descent optimizer, a learning rate of 1e-5, a batch size of 64, train with a maximum of 300 epochs and run 20 TARNet trainings.

Hate models to evaluate the LLM generated counterfactuals

Now, to calculate the ITEs (Eq. 2.1) and ATE (Eq. 2.2) of the factual/counterfactual pairs, we follow the methodology explained in Section 2.3. For all the hate prediction models tested, which have been trained on the 'hate training' dataset.

All neural network classifiers used with the hate prediction models present the same neural network architecture and training parameters. For the architecture, two layers, each one with 128 neurons. Regarding the training parameters, we utilize the Adam optimizer, a learning rate of 0.001, a batch size of 32, we train for 10 epochs and run 20 trainings. After some parameter tuning, we found that these settings, also used with the PEACE model from the backdoor criterion method, provided satisfactory performance for all models.

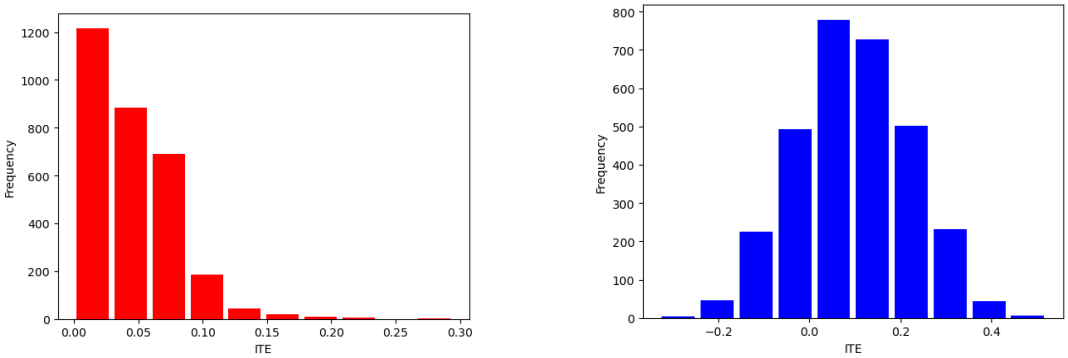
3.3.2 Results

Methods based on the backdoor criterion

Regarding the methods which follow the backdoor criterion, we obtain an ATE of 0.04 for the S-Learner approach with the PEACE model. If we change the neural network classifier by the TARNet, now we get an ATE of 0.10. The first ATE seems to be very small, but, in order to confirm that the curse of dimensionality is affecting the PEACE model, let's check the histograms of the ITEs, to observe their frequence distribution.

As shown in Figure 3.6, the higher frequencies concentrate on low ITE values for the PEACE model, and higher values get very few frequencies in comparison. In contrast, for the TARNet, we find high frequencies on a wider range of values. This suggests that, as we expected, the higher dimensionality of the covariates is limiting PEACE model's ability to estimate effectively the causal impact of the binary treatment.

Consequently, for the comparison with the method based in LLM generated counterfactuals, we are going to consider only the causal metrics of the TARNet.



(A) ITE PEACE histogram

(B) ITE TARNet histogram

FIGURE 3.6: Histograms of the ITE obtained with the methods that follow the backdoor criterion.

TARNet vs. LLM generated counterfactuals

Table 3.6 presents the ATE, Accuracy and F1-score obtained with the TARNet and with the hate models used in the LLM counterfactuals based method. These last methods achieve similar ATE values, which are considerably higher than the one from TARNet.

TABLE 3.6: Average Treatment Effect (ATE), Accuracy and F1-score of the tested models which predict hate.

Model	ATE	Acc.	F1-score
TARNet	0.10	0.76	0.61
PEACE for LLM cf eval.	0.30	0.73	0.66
HateBERT	0.29	0.73	0.63
RoBERTa hate	0.30	0.73	0.58
RoBERTa hate w/ our classifier	0.28	0.75	0.65

Regarding Accuracy and F1-score, we have obtained considerably close values between all models, with PEACE and the 'roBERTa hate with our classifier' achieving the higher F1-scores. The 'roBERTa hate' model obtains the lowest F1-score, likely due to the absence of an additional neural network trained on our hate training dataset.

We delve deeper into the comparison of the causal metrics by the use of scatter plots. We plot the ITEs from TARNet vs. the ITEs from the LLM counterfactuals based method. We need to remind that, from TARNet, we have as many ITEs as samples on the inference dataset. But, from the LLM counterfactuals method, we have 5 times more ITEs (since each factual has 5 ITEs associated, corresponding to the 5 counterfactuals generated). To compare them, we select, for each factual, the ITE closest to the TARNet ITE.

As we can observe in Figure 3.7, the scatter plots from all models, except for 'roBERTa hate', present a similar shape. While the values from TARNet are generally confined to the range of $[-0.25, 0.5]$, the values from the LLM counterfactuals

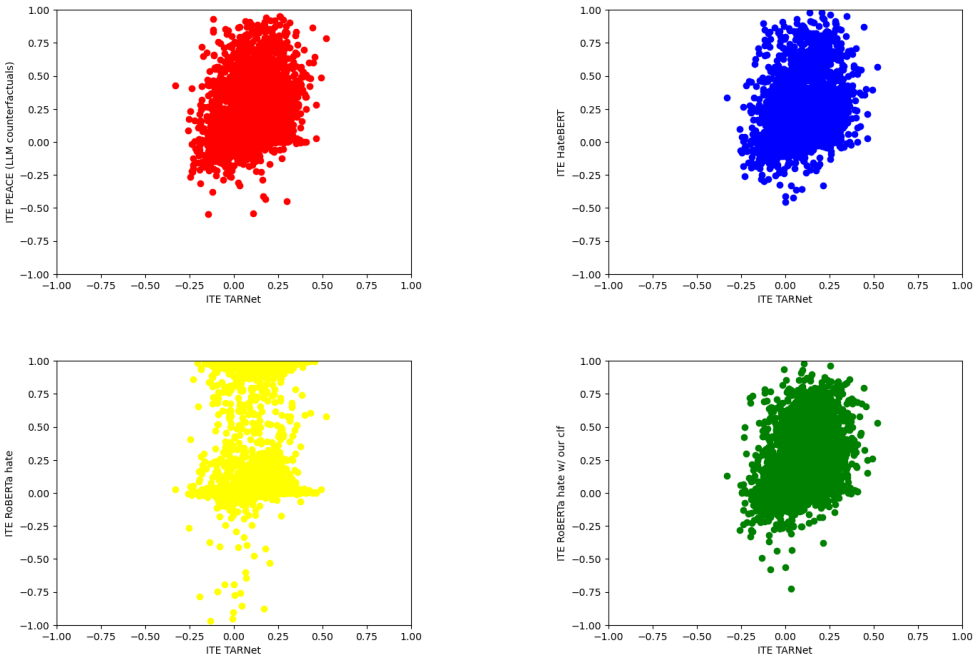


FIGURE 3.7: Scatter plots of TARNet ITEs vs. LLM counterfactuals ITEs.

models reach more extreme values (especially positive ones), with some ITEs reaching values close to 1.

The scatter plot of TARNet vs. 'roBERTa hate' indicates that 'roBERTa hate' reaches more extreme values than the other models, including some extreme negative ones. Again, the main difference of this model is its lack of an additional neural network trained on our hate training dataset.

In conclusion, the causal effect of offense on hate is higher with the LLM generated counterfactuals than with the methodology that follows the backdoor criterion. By modifying the offense by text revision, the comment is changed in a way that tends to increase the hate more than simply changing the value of a binary feature.

Chapter 4

Predicting ITE with a Machine Learning model

4.1 ITE model

As an additional approach to determining the causal impact of a comment, we are going to train a machine learning model to directly predict the ITE from a given factual comment, without needing to intervene it. The model will learn the following function to estimate the ITE given the covariates:

$$\mu \sim \mathbb{E}[\text{ITE}|S, M] \tag{4.1}$$

Figure 4.1 illustrates the architecture of the ITE model. We utilize the embeddings obtained from the meaning and sentiment models (Liu et al., 2019; Rosenthal, Farra, and Nakov, 2017) to next concatenate them and train a neural network regressor to estimate the ITE.

The ITEs are the target of the model. We use those computed from the hate predictions obtained with the PEACE model 2.8 (after processing the factual/counterfactual pairs through this PEACE model). We want to do the prediction only from the factual comment, however, we have obtained 5 ITEs for each factual (as far as we have generated 5 counterfactuals for each factual). We utilize all these ITEs, resulting in 5 different ITE values for the same factual.

But, for the training purpose, we can use these textual counterfactuals as factuals as well, and also train the model with them. Therefore, we have in the training set as many samples as two times the number of factual/counterfactual pairs, which gives a total number of 30,570 samples. Each sample consists of the sentiment and meaning embeddings as features and the ITE as the target. In order to evaluate the training, we split of a 30% test set from the training set.

The neural network consists of two layers, with 128 and 64 layers, respectively. Regarding the training parameters, we utilize the Adam optimizer, a learning rate of 0.001, a batch size of 32 and train for 20 epochs. Since the ITE is a continuous value, this neural network is trained for regression, with a mean squared error (MSE) loss function (Hastie et al., 2009).

As done in Section 3.3, we split off a 10% validation set from the training set, in order to select the model from the training run that achieves the lowest mean

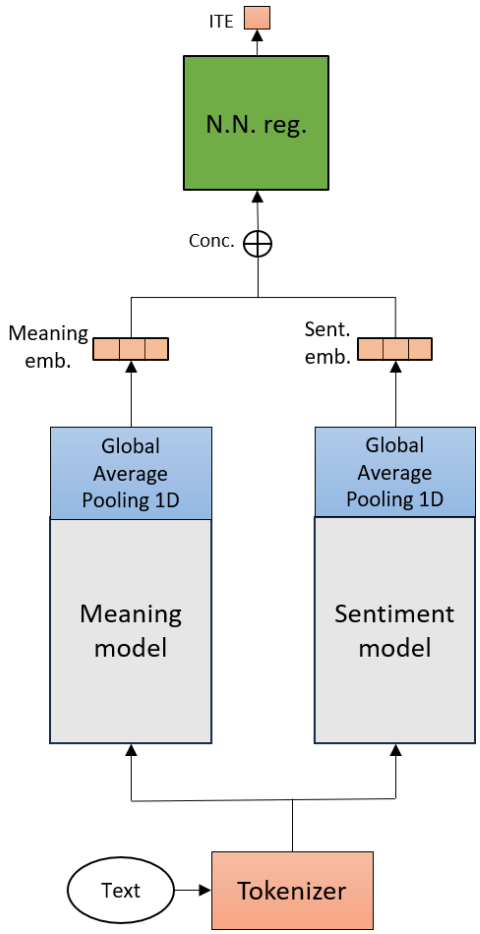


FIGURE 4.1: Architecture of the ITE model.

TABLE 4.1: Metrics to evaluate the ITE model.

Dataset	MSE	ATE
Training	0.033	0.30
Test	0.054	0.30

squared error. We performed 10 training runs.

For the evaluated metrics, we compute the mean square error for both training and test sets. Additionally, we calculate the ATE by averaging the predicted ITEs. As we can observe in Table 4.1, the resulting ATEs, for both training and test sets, match the ATE of 0.30 obtained with the PEACE model in Section 3.3. We consider this consistency, as well as the moderately low MSEs, a positive outcome.

Now, we can train the ITE model using both training and test sets, for being able to predict in a future the ITE of new comments.

Chapter 5

Conclusion

In this study, the hate speech detection problem has been addressed through a causal approach. We have investigated how offense influences the perception of hate speech in social media comments.

First, we constructed a causal graph to identify our accessible causal relationships that influence hate in comments. Then, the causal objective metrics, to quantify the impact of the treatment on the outcome, were defined, as well as the general methodology that we followed to perform causal inference with the distinct methods.

We selected and filtered two sets of data from existing hate benchmark datasets. The first set was used for causal inference and analysis, while the second set was utilized to train models that predict hate.

Concerning the first causal inference method, we defined how the backdoor criterion allows us to estimate the causal effect. Then, with the S-Learner meta-algorithm, we developed a deep learning architecture (inspired by the PEACE model from Sheth et al. (Sheth et al., 2023)) to predict the hate from the variables in the causal graph. By the additional use of a TARNet, we achieved to face the curse of dimensionality problem that we encountered when using the S-Learner.

Next, we explored another method to perform causal inference by intervening on the treatment variable using textual counterfactuals generated by GPT-3.5. We adopted some prompt engineering tactics to prompt the model and being able of generating the counterfactuals in a zero-shot manner, without requiring any training or fine-tuning. Then, the causal metrics were computed based on the hate values predicted by different models and architectures.

The counterfactuals generated by the LLM were compared to the factual comments to evaluate if GPT-3.5 could modify the offensiveness without significantly altering the sentiment and the meaning of the comment. Our findings suggest that the LLM is capable of modifying the offensiveness of comments, although the sentiment and semantic meaning are also altered to some extent.

Our causal inference experiments led us to the conclusion that, by modifying the offense by text revision with an LLM, the comment is changed in a way that tends to increase the hate more than with the traditional methodology, where we simply change the value of a binary feature.

Finally, as an additional approach to determine the causal effect of a comment, we trained a machine learning model to directly predict the Individual Treatment Effect (ITE) from a comment. With this model, we will be able to predict in a future the ITE of new comments.

As future directions, we could test other LLM's (such as GPT-4) in order to see if the results are consistent across different models.

We could also experiment with the prompt design by trying different number of shots for the few-shot setting (i.e., different number of examples). Or try to provide more specific task guidelines. To see how it affects the generated counterfactuals and the causal metrics.

Appendix A

GPT-3.5 counterfactuals

A.1 Prompt to make Non-Offensive a comment

"SYSTEM message":

I will give you a sentence that I consider OFFENSIVE. You must make it NON-OFFENSIVE by DECREASING its OFFENSIVENES without changing its MEANING. Here are a few examples of what I would like:

<Example 1> Sentence: Fu*king bastards. Stop mention God! Stupids! Stoooooop!

Revised Sentence: "Please refrain from mentioning God, anyone." </Example 1>

<Example 2> Sentence: "Military has and will always have GAYS. LOOK AT ancient Greece. You're a misanthropic bigot"

Revised Sentence: "Military has and will always have homosexuals. Look at ancient Greece." </Example 2>

<Example 3> Sentence: "I think your role will be to shut up and do as you are told. If not don't sign up."

Revised Sentence: "Your role will likely involve following instructions. If that's not something you're comfortable with, this might not be the right fit for you." </Example 3>

<Example 4> Sentence: "They ARE fighting for our country you fekkin pillock."

Revised Sentence: "They are fighting for our country!" </Example 4>

"USER message":

Sentence: <Input sentence>

Revised Sentence:

FIGURE A.1: Prompt messages to make a comment Non-Offensive.

Bibliography

Agirre, Eneko et al. (2012). "SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity.* SEM 2012: The First Joint Conference on Lexical and Computational Semantics—". In: *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), Montréal, QC, Canada*, pp. 7–8.

Antypas, Dimosthenis and Jose Camacho-Collados (July 2023). "Robust Hate Speech Detection in Social Media: A Cross-Dataset Empirical Evaluation". In: *The 7th Workshop on Online Abuse and Harms (WOAH)*. Toronto, Canada: Association for Computational Linguistics, pp. 231–242. URL: https://aclanthology.org/2023.woah-1.25.

Bauwelinck, Nina and Els Lefever (2019). "Measuring the impact of sentiment for hate speech detection on Twitter". In: *Proceedings of HUSO*, pp. 17–22.

Bhattacharjee, Amrita et al. (2024). "Zero-shot LLM-guided Counterfactual Generation for Text". In: *arXiv preprint arXiv:2405.04793*.

Bommasani, Rishi et al. (2021). "On the opportunities and risks of foundation models". In: *arXiv preprint arXiv:2108.07258*.

Caselli, Tommaso et al. (Aug. 2021). "HateBERT: Retraining BERT for Abusive Language Detection in English". In: *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*.

Chowdhary, KR1442 and KR Chowdhary (2020). "Natural language processing". In: *Fundamentals of artificial intelligence*, pp. 603–649.

Craig, Kellina M (2002). "Examining hate-motivated aggression: A review of the social psychological literature on hate crimes as a distinct form of aggression". In: *Aggression and Violent Behavior* 7.1, pp. 85–101.

Devlin, Jacob et al. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Gat, Yair et al. (2023). "Faithful explanations of black-box nlp models using llm-generated counterfactuals". In: *arXiv preprint arXiv:2310.00603*.

Haixiang, Guo et al. (2017). "Learning from class-imbalanced data: Review of methods and applications". In: *Expert systems with applications* 73, pp. 220–239.

Hastie, Trevor et al. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer.

Islam, Saidul et al. (2023). "A comprehensive survey on applications of transformers for deep learning tasks". In: *Expert Systems with Applications*, p. 122666.

Kennedy, Chris J et al. (2020). "Constructing interval variables via faceted Rasch measurement and multitask deep learning: a hate speech application". In: *arXiv preprint arXiv:2009.10277*.

Kıcıman, Emre et al. (2023). "Causal reasoning and large language models: Opening a new frontier for causality". In: *arXiv preprint arXiv:2305.00050*.

Krahé, Barbara (2020). *The social psychology of aggression*. Routledge.

Künzel, Sören R et al. (2019). "Metalearners for estimating heterogeneous treatment effects using machine learning". In: *Proceedings of the National Academy of Sciences* 116.10, pp. 4156–4165.

Lewis, David (2013). *Counterfactuals*. John Wiley & Sons.

Li, Yongqi et al. (2023). "Prompting large language models for counterfactual generation: An empirical study". In: *arXiv preprint arXiv:2305.14791*.

— (2024). "Prompting Large Language Models for Counterfactual Generation: An Empirical Study". In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 13201–13221.

Liu, Yinhan et al. (2019). "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692*.

Ljubešić, Nikola, Darja Fišer, and Tomaž Erjavec (2021). *Offensive language dataset of Croatian, English and Slovenian comments FRENK 1.0*. Slovenian language resource repository CLARIN.SI. URL: http://hdl.handle.net/11356/1433.

Ljubešić, Nikola, Darja Fišer, and Tomaž Erjavec (2019). *The FRENK Datasets of Socially Unacceptable Discourse in Slovene and English*. arXiv: 1906.02045 [cs.CL]. URL: https://arxiv.org/abs/1906.02045.

Marvin, Ggaliwango et al. (2023). "Prompt Engineering in Large Language Models". In: *International Conference on Data Intelligence and Cognitive Informatics*. Springer, pp. 387–402.

Nasar, Zara, Syed Waqar Jaffry, and Muhammad Kamran Malik (2021). "Named entity recognition and relation extraction: State-of-the-art". In: *ACM Computing Surveys (CSUR)* 54.1, pp. 1–39.

OpenAI (2023a). *GPT-3.5: OpenAI's Generative Pre-trained Transformer 3.5*. Accessed: 2024-06-16. URL: https://platform.openai.com/docs/models/gpt-3-5.

— (2023b). *Prompt Engineering*. Accessed: 2024-06-17. URL: https://platform.openai.com/docs/guides/prompt-engineering.

Pearl, Judea (2009). *Causality*. Cambridge university press.

Pearl, Judea et al. (2000). "Models, reasoning and inference". In: *Cambridge, UK: CambridgeUniversityPress* 19.2, p. 3.

Powers, David MW (2020). "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation". In: *arXiv preprint arXiv:2010.16061*.

Reimers, Nils and Iryna Gurevych (2019). "Sentence-bert: Sentence embeddings using siamese bert-networks". In: *arXiv preprint arXiv:1908.10084*.

— (2020). *SentenceTransformers: Multilingual Sentence, Paragraph, and Image Embeddings using BERT Co.* Version 2.2.0. URL: https://www.sbert.net.

Rosenthal, Sara, Noura Farra, and Preslav Nakov (2017). "SemEval-2017 task 4: Sentiment analysis in Twitter". In: *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pp. 502–518.

Salazar, Julian et al. (2019). "Masked language model scoring". In: *arXiv preprint arXiv:1910.14659*.

Salton, Gerard, Anita Wong, and Chung-Shu Yang (1975). "A vector space model for automatic indexing". In: *Communications of the ACM* 18.11, pp. 613–620.

Scikit-Learn, Documentation (2024). *Imbalanced data: using sample weights*. URL: https://scikit-learn.org/stable/auto_examples/svm/plot_weighted_samples.html.

Sengupta, Ayan et al. (2022). "Does aggression lead to hate? Detecting and reasoning offensive traits in hinglish code-mixed texts". In: *Neurocomputing* 488, pp. 598–617.

Shalit, Uri, Fredrik D Johansson, and David Sontag (2017). "Estimating individual treatment effect: generalization bounds and algorithms". In: *International conference on machine learning*. PMLR, pp. 3076–3085.

Sheth, Paaras et al. (2023). "Peace: Cross-platform hate speech detection-a causality-guided framework". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 559–575.

TensorFlow (2024). *tf.keras.layers.GlobalAveragePooling1D*. https://www.tensorflow.org/api_docs/python/tf/keras/layers/GlobalAveragePooling1D. Accessed: 2024-06-20.

Zampieri, Marcos et al. (2019). "SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval)". In: *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 75–86.

Zhou, Xianbing et al. (2021). "Hate speech detection based on sentiment knowledge sharing". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7158–7166.