University of Bergen
Department of Informatics

Investigating Biases in Rules Extracted from Language Models

Author: Sophie Martina Blum
Supervisors: Ana Ozaki, Samia Touileb

UNIVERSITETET I BERGEN
Det matematisk-naturvitenskapelige fakultet

June, 2023

Abstract

We investigate an approach for extracting occupational gender bias in the form of logical rules from Large Language Model (LLM)s based on Angluin's exact learning model with membership and equivalence queries to an oracle. In our approach, the oracle is a LLM and we show the changes that are necessary to use Angluin's algorithm with such an oracle. In our experiments, we extract occupational gender bias with the adapted algorithm from BERT and roBERTa models and compare our results to an established bias extraction method, which is template-based probing. Our goal is to use a new method to combine multiple attributes in a template sentence and to study their relationship to the gender in a sentence. We achieve this by using our rule extraction approach with a variable template containing multiple attributes. The extracted rules show a similar bias as previous bias extraction methods but also give insight into more complex relationships between attributes.

Acknowledgements

I would like to thank my supervisor Ana Ozaki for her support and guidance while writing this work. She always provided me with insight and mentorship throughout the whole writing process. I would also like to thank my co-supervisor Samia Touileb, who gave me many helpful tips on the way.

This work would not have been possible without the great support of my peers, who worked alongside me every week and motivated me to work. My biggest gratitude goes to my partner Ruben, who supported me, helped me raise our daughter on the way, and gave me the time I needed to write this thesis.

Sophie Martina Blum
Friday 30th June, 2023

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Artificial Intelligence (AI) models are widely used across many domains. Especially language models started to be available to everyone with LLMs and the recent developments towards conversational language models like ChatGPT from OpenAI, which is based on a LLM architecture. Most of these models are black boxes, and despite efforts to develop them unbiasedly, these systems can encode some harmful bias, often stemming from the training data [16]. When being used by a wide range of users, the biases in language models, and systems based on them, can become harmful and should be revealed and addressed.

Stereotypical biases in language models can often be exposed with a template-based probing approach. The templates are pre-defined sentence structures that combine a predicate with an attribute, depending on the task and the targeted bias [30, 10, 24, 7]. For the probing, the token of interest in the sentence is masked and then predicted by a model. An example template is "`[predicate] works as [description]`" [33] and with a pronoun or noun as a predicate and an occupation as the description, masking and predicting the pronoun is a way to detect occupational gender bias. It has been shown that these templates can be sensitive to grammatical changes, for example, a change in the grammatical tense [32], which motivates the development of more stable methods.

In an effort to contribute to explainable AI, we want to unravel the knowledge hidden in LLMs and find biases encoded in the models. Parts of this work are also submitted in an article [5]. Based on Anlguin's *exact learning model*, we investigate an approach to using it to extract knowledge from a LLM in the form of logical rules.

The exact learning model describes identifying an abstract target concept via interaction with a teacher (*oracle*) through queries. More specifically, a *minimally adequate teacher*

is an oracle that answers *membership* and *equivalence queries* about the target concept. For our case, a membership query is a variable assignment the learner gives that the oracle has to classify as either satisfied or not satisfied by the target. An equivalence query takes a hypothesis given by the learner as input, and the oracle decides whether it is equivalent to the target. The answer to the query is then the answer to the equivalence query and, in the case of a "no", an example of the difference. One specific application of exact learning is the HORN algorithm presented by Angluin et al. [3], which can learn rules as a Horn formula in polynomial time from a minimally adequate teacher. The algorithm starts with an empty Horn formula as a hypothesis and adds Horn clauses through negative examples. Those examples in the hypothesis that are not implied by the target Horn formula are removed through positive examples.
The HORN algorithm with a machine learning model acting as the oracle is the basis for our method [27, 28]. We extend it by using specifically a LLM as an oracle and apply it to extract rules that reflect occupational gender biases encoded in the model. This method is then compared to a template-based probing approach using the same set-up of attributes and the same template. We analyze the shared biases and highlight the rule extraction method's differences, benefits, and downsides.

In general, we want to achieve three goals. The **first** goal is to use the HORN algorithm to extract meaningful, logical rules from LLMs and, with that, find rules that describe the decision process of the underlying black-box machine learning model. With this method, we want to fulfill our **second** goal of introducing a different approach to template-based bias extraction, addressing the sensitivity of template-based probing approaches to changes in the template. The **last** goal is to use this method with a flexible template containing multiple attributes to compare the relationship of different attributes to the gender of an entity in a sentence. We want to find out if "gender" is generally more often explained by specific attributes and to find relationships including multiple attributes to describe an entity's gender.

Our contribution is a new method that combines Anlguin's HORN algorithm with LLMs to extract logical rules. These rules reflect the biases encoded in the language models. We combine multiple attributes to study their relationship to the gender in a sentence and therefore find a method to extract complex relationships between attributes in a sentence.

In the following, we will first give the necessary background information to understand our method in Chapter 2. We will cover the HORN algorithm (algorithm 1) and its extensions and bias in machine learning, specifically bias in Natural Language Processing (NLP). In Chapter 3, we further describe the idea behind our method, including problems of using the HORN algorithm for this specific task and their solutions. Detailed descriptions of all experiments are given in Chapter 4. We first develop a dataset for a template-based probing approach and a base for developing the attributes used for our rule extraction approach. Then we show the results of the template-based probing approach with our template and set of attributes, followed by the rule extraction approach using the HORN algorithm. We state all results and comment on them. In Chapter 5, we show related works on exact learning and bias in language models. In the end, we give a conclusion in Chapter 6 that summarizes the experiment results and evaluates the goals of this work. We also provide suggestions for extensions and improvements.

Chapter 2

Background

In the following, we define relevant concepts and notions to understand the fundamentals of the approach we present in this work. Some more common concepts are defined shortly, as we assume the reader to be familiar with the concepts (for example neural networks).

2.1 Neural Language Models

2.1.1 Neural Networks

Generally speaking, a neural network is a set of simple nodes that are connected over (multiple) layers and that process data to learn some task [13]. A *fully connected layer* is some input $x = [x_1, ..., x_n]$ and output $o = [o_1, ..., o_m]$ where the calculation of each output o_j depends on all inputs x_i [39]. This is shown in figure 2.1.

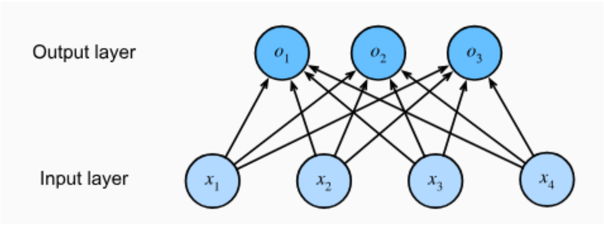


Figure 2.1: Visualization of one hidden layer with 4 input and 3 output nodes by Zhang et al. [39]

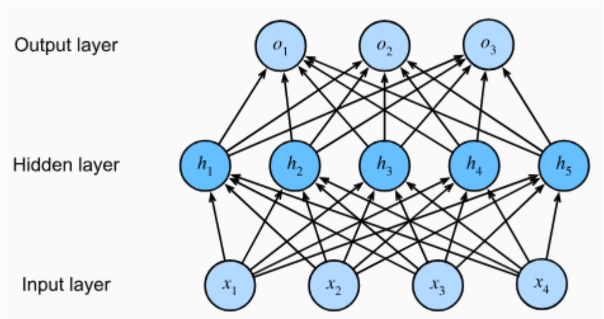


Figure 2.2: A Multi Layer Perceptron (MLP) with one hidden layer as visualized by Zhang et al. [39]

A MLP is a stack of multiple fully connected layers on top of each other, shown in figure 2.2. The outputs of a hidden layer are called *hidden representations* and are a linear combination of the previous layer's hidden representations. To use the full potential of deep architectures, each layer has a non-linear *activation function* [39].
A variation to this architecture is *residual connections*, where the input of one layer, in addition to being propagated through a block of layers, is also added to the output of the block layers. This allows a network architecture to learn the identity function easier [39].

There are many different techniques to regularize neural network architectures. The one that is relevant here is called *layer normalization*, which is applied to one representation (hidden representation or input) x at a time by normalizing its entries [39].

2.1.2 Language Models

The core objective of a language model is to estimate the joint probability of a sequence $x_1, x_2, ..., x_T$ of T tokens $P(x_1, x_2, ..., x_T)$. A sequence of tokens can be any sequence; in the context of natural language processing, this is often the mapping of tokens to words or characters [39]. This can be very useful in a perfect scenario because a language model could generate natural text or meaningful dialogues [39]. If the model that estimates the probability of a sequence is a neural network (of some kind), it is called neural language modeling. This work focuses on transformer-based language models, specifically the encoder-only language models BERT and RoBERTa.

The input and output of a language model can take different shapes. They are considered to be *aligned* if input and output show some kind of step-by-step correspondence, and *unaligned* if they don't.

2.1.3 Transformer Architecture

A core building block of the *transformer* architecture is the *attention mechanism* [35], which is a function that maps a *query* q and a set of *key-value pairs* $D = \{(k_1, v_1), ..., (k_m, v_m)\}$ to an output, which can generally be expressed as [39]

$$Attention(q, D) = \sum_{i=1}^{m} \alpha(q, k_i) v_i \tag{2.1}$$

For the transformer architecture [35], the attention function α is called *Scaled Dot-Product Attention* and is calculated simultaneously for a seq of queries. With query matrix $Q \in \mathbb{R}^{\ltimes \times \neg}$, key matrix $K \in \mathbb{R}^{m \times d_k}$ and value matrix $V \in \mathbb{R}^{m \times d_v}$, the scaled dot-product attention is defined as

$$Attention(Q, K, V) = softmax\left(\frac{QK^\top}{\sqrt{d_k}}\right) V \tag{2.2}$$

This attention function is used as *self-attention*, where queries, keys, and values all come from the same sequence. Another method is called *multi-head attention*, where the queries, keys, and values are transformed h times with different, learned linear projections. Then there are h parallel attention computations, whose results are concatenated and projected onto the final values [35].

Scaled dot-product and multi-head attention are visualized by Vaswani et al. [35] and shown in figure 2.3.

The transformer architecture [35] is an example of an encoder-decoder architecture dealing with unaligned inputs and outputs of varying lengths. The *encoder* transforms a variable-length sequence into a fixed-shape state, and the *decoder* uses this output and the leftwards context of already generated targets to predict the next token. They are connected through a *multi-head attention layer* called *encoder-decoder attention*, which uses the output from the encoder as keys and values, while the output of the previous decoder layer acts as the queries.

A transformer adds *positional encodings* to the word embeddings of the input, which encode information about the relative or absolute position of tokens. In the original architecture, Vaswani et al. [35] use sine and cosine functions of different frequencies as

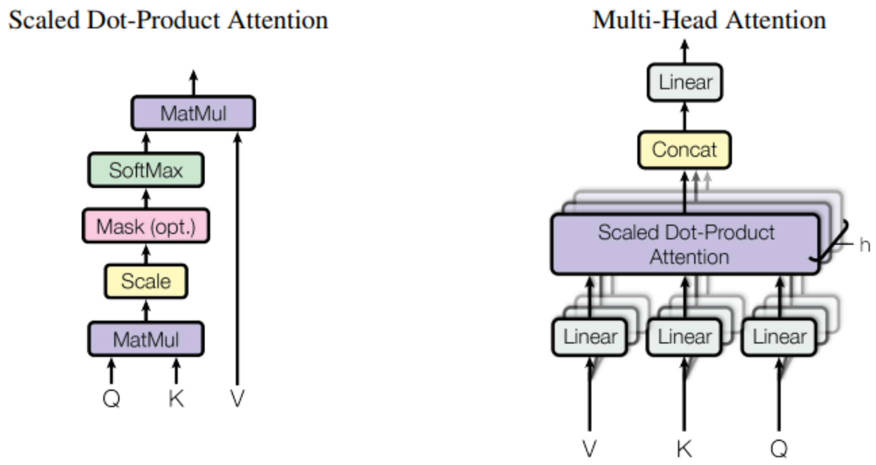


Figure 2.3: Scaled Dot-Product Attention and Multi-Head Attention as described and visualized by Vaswani et al. [35].

the positional encodings, calculated based on the position *pos* of the token and dimension in the embedding *i*.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}})$$

The encoder consists of multiple identical layers, each having two sublayers. First, a multi-head self-attention pooling layer and then a position-wise feed-forward neural network (all sequence positions are transformed with the same MLP). Both sublayers also have residual connections and are each followed by layer normalization.

The decoder has a similar structure. The multi-head attention in the decoder is a masked scaled dot-product attention, meaning that each position is only allowed to attend to positions to the left of it to preserve the auto-regressive nature of the decoder. This is achieved by masking the values of connections between positions that are not allowed [35]. In addition, after the first multi-head attention, the encoder-decoder attention connects the encoder and decoder values as previously described. Apart from that, the architecture of the decoder is the same as that of the encoder. The architecture of the whole transformer is shown in figure 2.4.

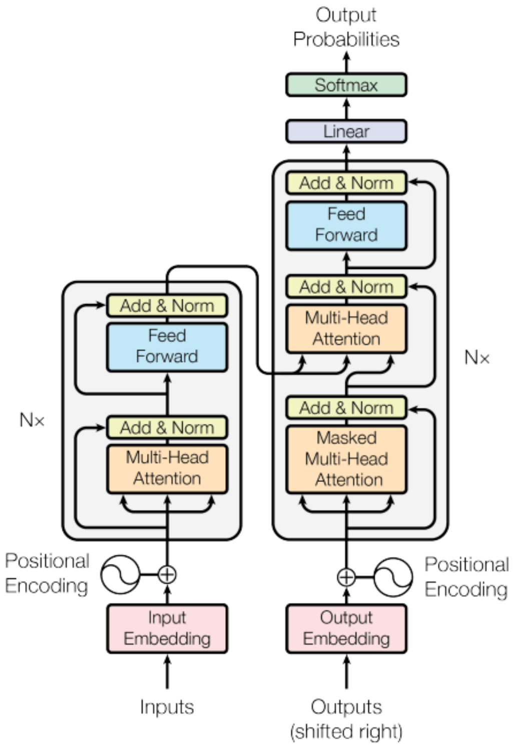


Figure 2.4: Transformer Architecture [35]

2.1.4 BERT and RoBERTa

BERT stands for Bidirectional Encoder Representations from Transformers [9]. It is a language model built from transformer encoders that can be used for various tasks due to general pre-training and fine-tuning on specific tasks. Pre-training describes the training of the underlying (in this case) transformer encoder architecture. With fine-tuning, a task-specific output layer is added, and both the output layer and the pre-trained architecture are then trained again with a task-specific training process. This is a form of transfer learning which allows models to be used for various tasks without explicitly training the whole model for each of them. BERT encodes context *bidirectionally*, which is also reflected in the pre-training objective. An output layer can be added and trained during fine-tuning for any downstream task.

The input to BERT consists of a single text or text pairs marked by certain tokens between test examples. Any input sequence starts with the <cls> token, followed by the input text and a <sep> token. If the input is a single text example, it ends with the <sep> token. Otherwise, it will be followed by the second text of the text pair and another <sep> token. As an example, the BERT input of the sequence pair “She is a researcher.” “Her field of research is machine learning.” would be “<cls> She is a researcher. <sep> Her field of research is machine learning. <sep>”. This sequence of

tokens is then transformed with learned embeddings, which are then added to learnable positional embeddings.
The model is a multi-layer bidirectional transformer encoder defined in two sizes by the original authors [9]. BERT-base has 110 million parameters, spread over 12 transformer blocks with a hidden size of 768 and 12 attention heads. With 340 million parameters, BERT-large is more than double that size. It has 24 transformer blocks, a hidden size 1024, and 15 attention heads.
BERT's pre-training is conducted on the BookCorpus (800 million words) [40] and English Wikipedia (2500 million words). It is done in an unsupervised fashion with two objectives. The first objective is Masked Language Modeling (MLM), a pre-training for the bidirectional representations, in which 15% of the input tokens will be masked randomly and predicted by the model. To mitigate a mismatch between training and fine-tuning due to the <MASK> token, it will only be used 80% of the time. 10% of the time, a random token will be filled in, and for the last 10%, the token will remain unchanged. The MLM objective allows for word-filling or sentence completion tasks on the pre-trained model without fine-tuning, as the model can be asked to return the most likely token. We use this property to do all of our inference on the raw pre-trained model without fine-tuning. To also understand the relationships between two sentences, BERT is trained with a second objective called Next Sentence Prediction (NSP), in which the model has to decide if a sentence B is the actual next sentence that follows the other given sentence A. To generate the pre-training examples, in 50% of the cases the two sentences are actually consecutive sentences and in the other 50% sentence A is followed by a random sentence from the corpus.

RoBERTa [19] is a modification of BERT with changes to the fine-tuning process and different training data. First, the RoBERTa models have longer training and use bigger batches and more data. To be more specific, it uses 160GB of uncompressed text with the following corpora: CC-NEWS [11], BookCorpus [40], OpenWebText [12] and Stories [34]. The authors also remove next-sentence prediction as a pre-training objective and show that this does not result in a performance loss. Additionally, the training is conducted on longer sequences. While BERT is trained on sequences of length 128 for the first 90% of updates, RoBERTa is consistently trained on sequences of length 512.

Through many different studies, large language models have been shown to display harmful biases. BERT displays societal biases in sentiment analysis [4] and its core task of masked language modeling [18]. Both BERT and RoBERTa display stereotypical bias in next sentence prediction [22] and show harmful behavior in sentence completion for LGBTQIA+ individuals [24].

2.1.5 Bias in Language Models

Gender bias is a “systematic, unequal treatment based on one’s gender” [31]. For NLP, a definition of gender bias in the context of language is helpful.Hitti et al. [15] give such a definition as “the use of words or syntactic constructs that connote or imply an inclination or prejudice against one gender”. In this work, we talk about a *contextual bias*, namely societal stereotypes, as we consider occupations and the gender they are stereotypically related to.

(Gender) Bias in language models can be measured by a template-based approach that measures how much more a model prefers associating a certain attribute with a certain gender. For this, templates, for example, of the form “`[TARGET] is a [ATTRIBUTE]`” [18] are created. The attribute could be anything like an occupation or a descriptive adjective. An example would be the association between the target “male” and the attribute “programmer” with the template sentence “`[MASK] is a programmer`”. The likelihood to fill the [MASK] token with “he” p_{tgt} is weighed with the *prior bias* of the model towards the “he” token p_{prior}. This prior bias is calculated by removing the attribute from the template sentence, giving the prior template “`[MASK] is a [MASK]`”. The probability for the sentence “`He is a [MASK]`” is p_{prior}. With that and p_{tgt}, the association of a target with an attribute is then calculated as $\log(\frac{p_{tgt}}{p_{prior}})$. The difference between these two measures for two targets is called *log probability bias score* and a measure of bias [18]. Munro and Morrison [21] use a similar approach but exchange the difference of log probabilities with a ratio of actual probabilities.
Fatemi et al. [10] introduce the *PPBS*, which takes the difference of the actual probabilities for the pronouns.

$$ppbs = p_{He} - p_{She} \tag{2.3}$$

They use the probability given by the model to fill a masked position with the pronoun “she” (or “he”), while Touileb et al. [33] first normalize the probabilities to add up to one. Template-based approaches are an easy way to test a language model for bias. However, they are sensitive to the formulation of the templates. Altering the grammatical tense of a template has been shown to affect the correlation between gender and occupation [32].

A study on BERT-base with the 60 most biased professions [10] shows how thePPBSis used to detect bias. Figure 2.5 shows thePPBSfor 60 occupations, where negative values correspond to female-biased occupations and positive values correspond to male-biased occupations. Most scores are closer to the lower half of their spectrum (-0.5 to -1 and 0.5 to 1). Especially the male-biased occupations all have a PPBS of over 0.7.

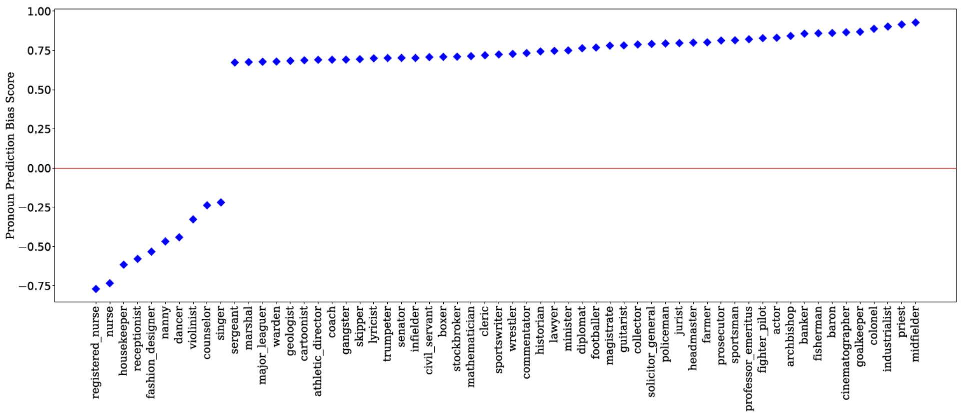


Figure 2.5: PPBS of the 60 most biased profession words (according to [6]) in the BERT-base model as seen in [10].

2.2 Computational Learning Theory

Computational learning theory is concerned with formalizing frameworks that describe learning tasks. Those frameworks can then, for example, define the class of learnable concepts or give information about the sample complexity of a learning algorithm or general constraints [20]. It provides theoretical considerations for the machine learning algorithms that are being used every day.

2.2.1 PAC learning

In a Probably Approximately Correct (PAC) learning setting [29], a learner tries to identify some target concept t by creating a hypothesis h that approximates t and the probability that h misclassifies an example is supposed to be bounded [26]. First, we give the necessary definitions.

Definition 1 (Learning Framework as defined by Konev et al. [17])**.** A *learning framework* $\mathfrak{F}$ is a triple $(\mathcal{E}, \mathcal{H}, \mu)$ with a set of *examples* $\mathcal{E}$ and a set of *concepts* $\mathcal{H}$. μ is a mapping from the set of concepts $\mathcal{H}$ to $2^{\mathcal{E}}$.

For a learning framework $\mathfrak{F} = (\mathcal{E}, \mathcal{H}, \mu)$, $\mathcal{D}$ is a probability distribution over $\mathcal{E}$.

Definition 2 (Example Query as defined by Ozaki [26])**.** An *example oracle* is an oracle $\mathsf{EX}_{\mathfrak{F},t}^{\mathcal{D}}$ that for a target $t \in \mathcal{H}$ outputs a *classified example* $(x, l_t(x))$ where $x \in \mathcal{E}$ is sampled according to the probability distribution $\mathcal{D}$ and $l_t(x)$ is the label with $l_t(x) = 1$ if $x \in \mu(t)$ and $l_t(x) = 0$ if $x \notin \mu(t)$. An *example query* is a call to an example oracle.

An example oracle can be used to generate a *sample* S, a set of independently and identically distributed examples according to $\mathcal{D}$.

Definition 3 (PAC learnability[26, 29])**.** Let $\mathfrak{F}$ be a learning framework with hypothesis space $\mathcal{H}$ and $\mathsf{EX}^{\mathcal{D}}_{\mathfrak{F},t}$ and example oracle. The learning framework $\mathfrak{F}$ is *PAC learnable* if there exists a function $f : (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property: For every $\epsilon, \delta \in (0,1)$, every probability distribution $\mathcal{D}$ on $\mathcal{E}$ and every target $t \in \mathcal{H}$, when running the learning algorithm on $m \geq f(\epsilon, \delta)$ examples generated by $\mathsf{EX}^{\mathcal{D}}_{\mathfrak{F},t}$, the algorithm always halts and returns a hypothesis $h \in \mathcal{H}$ such that, with a probability of at least $1 - \delta$ over the choice of m examples, we have that $\mathcal{D}(\mu(h) \oplus \mu(t)) \leq \epsilon$, with $\oplus$ being the symmetric set difference.

Because the learner is constantly working with finite samples S to approximate t, there is always a chance that the learned hypothesis may not reflect the target perfectly. Nevertheless, to achieve a PAC solution, the function $f : (0,1)^2 \to \mathbb{N}$ determines the minimum amount of required examples, the *sample complexity* based on the accuracy and confidence parameters ϵ, δ and the hypothesis class $\mathcal{H}$. In the case of a finite hypothesis class, it depends on the log of the size of $\mathcal{H}$. As shown by Shalev-Shwartz and Ben-David [29], every finite hypothesis class is PAC learnable with sample complexity

$$f(\epsilon, \delta) \geq \left\lceil \frac{1}{\epsilon} \log(\frac{|\mathcal{H}|}{\delta}) \right\rceil \tag{2.4}$$

2.2.2 Exact Learning

Angluin [1] describes the problem of exactly identifying an unknown target concept t while having access to a set of oracles that can answer queries about t. First, we give relevant definitions for exact learning [17].

Definition 4 (Learning Framework as defined by Konev et al. [17])**.** A *learning framework* $\mathfrak{F}$ is a triple $(\mathcal{E}, \mathcal{H}, \mu)$ with a set of *examples* $\mathcal{E}$ and a set of *concepts* $\mathcal{H}$. μ is a mapping from the set of concepts $\mathcal{H}$ to $2^{\mathcal{E}}$.

An example $x \in \mathcal{E}$ is a *positive example* for a concept $h \in \mathcal{H}$ if $x \in \mu(h)$ and, conversely, a *negative example* if $x \notin \mu(h)$.

Definition 5 (Memberhsip Query[17])**.** A *membership oracle* is an oracle $\mathsf{MEM}_{\mathfrak{F},t}$ that takes as input an example $x \in \mathcal{E}$ and returns "yes" if $x \in \mu(t)$ and "no" otherwise. A call to $\mathsf{MEM}_{\mathfrak{F},t}$ with an example $x \in \mathcal{E}$ is a *membership query* for example x.

For two concepts $t, h \in \mathcal{H}$, a *counterexample* $x \in \mathcal{E}$ is an example that is a positive example for t and a negative example for h (*positive counterexample*) or vice-versa (*negative counterexample*). Formally this means that $x \in \mu(h) \oplus \mu(t)$.

Definition 6 (Equivalence Query[17])**.** A *equivalence oracle* is an oracle $\mathsf{EQ}_{\mathfrak{F},t}$ that takes as input a *hypothesis* concept $t \in \mathcal{H}$ and returns "yes" if $\mu(h) = \mu(t)$ and a *counterexample* x otherwise. A call to the equivalence oracle $\mathsf{EQ}_{\mathfrak{F},t}$ with a hypothesis $h \in \mathcal{H}$ is an *equivalence query* for hypothesis h.

Definition 7 (Exact Learning [17, 1])**.** A target concept t is exactly learnable by a *learning algorithm* for the learning framework $\mathfrak{F}$ if the algorithm takes no input, is deterministic, always halts and outputs a hypothesis $h \in \mathcal{H}$ with $\mu(t) = \mu(t)$ by only posing queries to certain oracles.

If a learning algorithm uses only membership and equivalence oracles $\mathsf{MEM}_{\mathfrak{F},t}$ and $\mathsf{EQ}_{\mathfrak{F},t}$, it is called a *minimally adequate teacher* [1].

The exact learning model is connected to the PAC learning model by extension with membership queries through the following theorem [26]:

Theorem 1 ([1, 26])**.** *If a learning framework is exactly learnable in polynomial time, then it is PAC learnable with membership queries in polynomial time* [1] *If only equivalence queries are used, then it is PAC learnable without membership queries in polynomial time.*

2.3 Learning from Neural Networks

The learning frameworks described in 2.2 are general frameworks that describe the learnability of general, finite hypothesis spaces. They build a basis for more specified algorithms tailored toward specific hypothesis spaces. One example is the application of the exact learning framework to a specific hypothesis space, like the set of all propositional Horn formulas.

[1] This also requires that deciding whether an example is positive can be done in polynomial time, which is the case for propositional Horn.

2.3.1 Learning Conjunctions of Horn Clauses

Angluins algorithm [3] learns the class of propositional Horn formulas in polynomial time in the exact learning model through membership and equivalence queries.

Before we describe the algorithm, we provide basic notions relevant to the definition of propositional Horn logic [28]. Let $V = \{v_1, ..., v_n\}$ be a set of Boolean variables. A literal is a variable $v \in V$ (positive literal) or its negation $\neg v$ (negative literal). A clause over V is a disjunction of literals called Horn if, at most, one literal is positive. A Horn formula (or theory) is a conjunction of Horn clauses. The *antecedent*(c) of a Horn clause c is the set of negated variables in c or the constant symbol $\top$. The *consequent*(c) of c contains either its unnegated variable if it exists or the constant symbol $\bot$. An example of a Horn clause c is $\neg a \vee \neg b \vee \neg c \vee d$, which is, as all Horn clauses, logically equivalent to an implication of the form $\mathit{antecedent}(c) \rightarrow \mathit{consequent}(c) \Leftrightarrow a \ \wedge \ b \ \wedge \ c \rightarrow d$.
An interpretation $\mathcal{I}$ over V is a subset of V. It satisfies a variable $v \in V$ if $v \in \mathcal{I}$ (written $\mathcal{I} \models v$) and otherwise falsifies it. The reverse holds for a negative literal $\neg v$. $\mathcal{I}$ satisfies a clause c iff it satisfies at least one literal in c ($\mathcal{I} \models c$) and satisfies a formula t iff it satisfies every clause in t ($\mathcal{I} \models t$). For a theory t and clause c, if, for every $\mathcal{I}$, we have that $\mathcal{I} \models t$ implies $\mathcal{I} \models c$, then $t \models c$ and t *entails* c. If t entails every clause in a theory t', then $t \models t'$ and if also $t' \models t$, then t and t' are logically equivalent (written $t \equiv t'$).

In this setup, a *learning framework* is a triple $(\mathcal{E}, \mathcal{H}, \mu)$ with $\mathcal{H}$ being the set of all formulas in propositional logic, $\mathcal{E}$ the set of all interpretations over V and μ defined as $\mu(t) = \{\mathcal{I} \in \mathcal{E} \mid \mathcal{I} \models t\}$. With this, we can redefine the notions given in Section 2.2.2.
For any $h \in H$, a *positive example* is an interpretation $\mathcal{I}$ that satisfies h ($\mathcal{I} \models h$). A *negative example* is an interpretation $\mathcal{I}$ with $\mathcal{I} \not\models h$. For any two formulas $t, h \in \mathcal{H}$, a *positive* (*negative*) *counterexample* for t and h is an interpretation $\mathcal{I} \in \mathcal{E}$ such that $\mathcal{I} \models t$ and $\mathcal{I} \not\models h$ ($\mathcal{I} \models h$ and $\mathcal{I} \not\models t$). In the context of the algorithm, we would talk about a target t and hypothesis h.
The algorithm uses equivalence and membership queries to follow the exact learning framework with a minimally adequate teacher (2.2.2). A membership query in this context is defined as a call to the *membership oracle* $\mathsf{MQ}_{\mathfrak{F},t}$ that takes an interpretation $\mathcal{I}$ and outputs *yes* if $\mathcal{I} \models t$ and *no* otherwise. An equivalence query is similarly defined as a call to the *equivalence oracle* $\mathsf{EQ}_{\mathfrak{F},t}$, which takes a hypothesis $h \in \mathcal{H}$ and outputs *yes* if $h \equiv t$ and outputs *no* and a counterexample for t and h otherwise.

Algorithm 1 learns a target Horn theory t in the learning framework $\mathfrak{F}$ by posing membership and equivalence queries. Every negative counterexample $\mathcal{I}$, an equivalence query outputts, violates some clause in t. The idea of the algorithm is to pose equivalence queries until the hypothesis h is equal to the target $h \equiv t$. For every negative counterexample it receives, it would like to add the corresponding violated clauses to its hypothesis, because every negative counterexample can be explained by a set of different Horn clauses. Every positive counterexample exposes clauses that were wrongfully added. Clauses added by negative counterexamples might be too weak. Therefore, the algorithm curates a sequence S of negative counterexamples to generate its hypothesis. Instead of adding all clauses for a given negative counterexample, the algorithm first tries to refine previous negative counterexamples by intersection. The counterexamples in S approximate distinct clauses of the target t. Hence only the first possible counterexample in S is refined by a new negative counterexample.

Algorithm 1 exactly identifies every Horn theory with m clauses over n variables in time $O(m^3n^4)$ using $O(m^2n^2)$ equivalence queries and $O(m^2n)$ membership queries [3].

Algorithm 1: HORN

```
It is assumed that the algorithm knows $\mathfrak{F}$
Let $S$ be the empty sequence
Denote with $\mathcal{I}_i$ the $i$-th element of $S$
Let $h$ be the empty hypothesis
while $\mathsf{EQ}_{\mathfrak{F},t}(h)$ returns a counterexample $\mathcal{I}$ do
    if there is a $c \in h$ such that $\mathcal{I} \not\models c$ then
        remove all $c \in h$ such that $\mathcal{I} \not\models c$
    else
        if there is $\mathcal{I}_i \in S$ such that $\mathcal{I}_i \cap \mathcal{I} \subset \mathcal{I}_i$ and $\mathsf{MQ}_{\mathfrak{F},t}(\mathcal{I}_i \cap \mathcal{I}) =$ 'no' then
            replace the first such $\mathcal{I}_i$ with $\mathcal{I}_i \cap \mathcal{I}$ in $S$
        else
            append $\mathcal{I}$ to $S$
        end
        $h := \bigcup_{\mathcal{I} \in S} \{(\bigwedge_{v \in \mathcal{I} \cup \{\top\}} v) \rightarrow u \mid u \in (V \cup \{\bot\} \setminus \mathcal{I})\}$
    end
end
Return $h$
```

2.3.2 Extracting Horn Theories from Neural Networks

The HORN algorithm (algorithm 1) learns a Horn theory given a Horn oracle, but it can be adapted to learn from neural networks as well [28]. The goal is to extract rules hidden in a “black-box machine learning model” using the HORN algorithm with a neural network acting as the oracle. This requires redefining the queries in the context of the oracle.

In addition to the preliminaries given in Section 2.3.1, a neural network model N is a function $N : \{0,1\}^{|V|} \longrightarrow \{0,1\}$ that takes a binary vector in the $|V|$ dimensional spaces and outputs its classification [28]. Interpretations are mapped to vectors by assuming a total order on the elements of V, and the vector for any interpretation $\mathcal{I}$ is then $vector(\mathcal{I}) \in \{0,1\}^{|V|}$, with the element at position i being 1 if $v_i \in \mathcal{I}$ and 0 otherwise. For every neural network N (trained on a given dataset), there exists a propositional formula t_N such that $N(vector(\mathcal{I})) = 1 \leftrightarrow \mathcal{I} \models t_N$.

To extract the rules hidden in a given neural network model, it is assumed that the underlying learning framework $(\mathcal{E}, \mathcal{H})$ is Horn.
Membership queries are simulated by directly using the classifier N. If $N(vector(\mathcal{I})) = 1$, then $\mathcal{I} \models t_N$ and the answer to the query is “yes” ('no' otherwise).
Equivalence queries are more complex, as the neural network model cannot answer directly if a hypothesis h is equivalent to t_N. One strategy for simulating equivalence queries is to create a sample of random interpretations and classify them through membership queries. If the hypothesis misclassifies any generated examples, it can be used as a counterexample, and the answer to the equivalence query is “no”. Otherwise (if all examples are classified correctly by the hypothesis), there is, with high probability, little difference between t_N and the hypothesis h. A small difference means the total number of misclassified interpretations is low, considering the entire space of possible interpretations. With this idea, it can be guaranteed that the hypothesis is PAC (Section 2.2.1) by choosing an appropriate sample size. Angluin [1] shows that the number of examples that are sampled randomly after the i-th equivalence query has to be greater or equal to $\lceil \frac{1}{\epsilon}(\ln \frac{1}{\delta} + i \ln 2) \rceil$ with error ϵ and confidence δ. This formula holds for finite or countable hypothesis spaces. The sample complexity for PAC learning from a finite hypothesis space $\mathcal{H}$, taking its size into account, is further specified in 2.2.1 as $\left\lceil \frac{1}{\epsilon} \log(\frac{|\mathcal{H}|}{\delta}) \right\rceil$.

The adapted version of the algorithm also allows for *background knowledge* to be used [27]. Instead of the empty hypothesis, the algorithm starts with a pre-defined set of Horn formulas that are assumed to be true properties of the domain. This is useful for encoding properties of the input variables or general prior knowledge about the target before starting the algorithm.

With membership and equivalence queries set up, the last obstacle is that a neural network might not encode a Horn theory, as it may return counterexamples like a non-Horn oracle, even when trained on a Horn theory [28]. Because of this, the algorithm

HORN may not terminate, as it assumes the oracle to be Horn. A solution is to check after every positive counterexample if every example $\mathcal{I} \in S$ falsifies the hypothesis. The examples that satisfy the hypothesis are marked and not returned in subsequent equivalence queries. This ensures that the algorithm does not get stuck in an infinite loop, producing the same counterexample repeatedly [28][14].

An example given by Persia et al. [28] is the set of variables $V = \{v_1, v_2\}$ and an oracle N that classifies $\emptyset$ as 0 and $\{v_1\}$ and $\{v_2\}$ as 1, which means N does not encode a Horn theory. The algorithm 1 starts with the empty hypothesis. With the first equivalence query, it gets $\emptyset$ as a negative counterexample. This results in the hypothesis $h = \{\top \rightarrow v_1, \top \rightarrow v_2, \top \rightarrow \bot\}$. Following that, the two positive counterexamples $\{v_1\}$ and $\{v_2\}$ can be returned to refine the hypothesis, which will then be $\emptyset$ again. Now $\emptyset$ may be returned as a negative counterexample again, and the algorithm is in an infinite loop.

2.4 Intersection over Union

To measure the similarity of two sets A and B, we introduce their IoU value (also called *Jaccard index*) [39]. It is defined by

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.5}$$

A value of 1 means that A and B are identical, while a value of 0 means that the sets do not overlap. If A and B are, as in our case, sets of logical rules, their intersection $A \cap B$ is all those rules that are identical in both sets, given that all rules are Horn clauses c of the form $\mathit{antecedent}(c) \rightarrow \mathit{consequent}(c)$. For example with the sets $A = \{v_1 \wedge v_2 \rightarrow v_3, v_1 \wedge v_3 \rightarrow \bot\}$ and $B = \{v_1 \wedge v_3 \rightarrow \bot \; v_1 \wedge v_4 \rightarrow v_2\}$ their intersection would be $A \cap B = \{v_1 \wedge v_3 \rightarrow \bot\}$ and their union is $A \cup B = \{v_1 \wedge v_2 \rightarrow v_3, v_1 \wedge v_3 \rightarrow \bot, v_1 \wedge v_4 \rightarrow v_2\}$, which gives an IoU value of $J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{1}{3}$.

Chapter 3

Learning from Language Models

The HORN algorithm —(algorithm 1) uses a Horn-oracle to answer equivalence and membership queries to learn Horn rules. There are, however **three obstacles** to be considered to apply that learning algorithm to language models. The **first** obstacle is the mismatch of input and output formats. The language model takes as input a sentence in natural language and outputs a probability distribution over word tokens. In contrast, in Angluin's algorithm, the oracle works with the given boolean interpretations and gives boolean answers to the queries. The **second** obstacle is having the language model answer equivalence queries. The **third** obstacle is that a language model as an oracle is unlikely to represent a Horn theory. It is addressed in our article [5], which is currently under review.

3.1 Input and Output Conversion

We want to use a language model as the oracle in Angluin's algorithm. Specifically, we use a language model for the task of pronoun prediction, which is possible due to the MLM objective of recent large language models. That means the language model (as the oracle) takes as input natural language, in our case, one sentence with a masked token. The output is a probability distribution over possible word tokens to fill the masked position. In the case of pronoun prediction, we focus on the gendered pronouns "He" and "She" as predicted tokens. These input and output formats are incompatible with the oracle's boolean interpretations and output in Angluin's algorithm.

We propose a conversion by creating a boolean function that can map boolean interpretations, interpreted as vectors with binary entries, to natural language. This function has to be task-specific, as mapping to natural language would otherwise require an infeasible large amount of boolean variables to encode all possible words without restrictions. In addition, the variables for the algorithm should be meaningful so that we can extract meaningful rules, which is why using some sort of embedding is not feasible. For this reason, we set up a search space consisting of specific attributes and their discrete values and a template sentence. We use a lookup table to convert a set amount of boolean variables, represented by a binary vector, into natural language attributes. These attributes are then set into the template sentence that is predefined and used for all conversions.
The attributes can be anything of interest to the task at hand. Concerning pronoun prediction to detect occupational bias, the main attributes are `occupation` and `gender`. The latter is masked in the template sentence but later used to evaluate the results. In addition, we want to introduce attributes like `birth year` and `nationality` to compare them to `occupation`. Discrete attributes can be encoded directly by creating as many boolean variables as there are different values for that attribute. Every value is then assigned to one boolean variable as a one-hot encoding. That means that no two variables corresponding to a single attribute can simultaneously be set to *true* $= 1$. If all variables corresponding to an attribute are set to 0, the attribute takes its "unknown" value, which must also be defined beforehand. This has been done for our task's `occupation` and `nationality` attributes, as shown in table A.1.
Continuous variables, such as `birth year`, may be represented by dividing their values into predetermined intervals and assigning any concrete variable the value of its corresponding interval. This transforms a continuous variable into a discrete one, which can then be encoded as such. In our task, this has been done for `birth year`, dividing possible values into five intervals as shown in table A.1. This also allows changing the accuracy of the rules regarding one specific attribute, as one can change how the intervals are defined.

The prediction of the language model needs to be converted back to a binary result to answer membership queries with either *true* or *false*. We added `gender` as an attribute to the variables so that one interpretation represents a complete sentence, including the gendered pronoun, and the target theory is about sentences being *correct* or *incorrect*. We interpret *correct* as *most likely* in the sense of the language model. A membership query then answers whether a given interpretation (and the underlying sentence) is *correct* and therefore valid. This is illustrated with an example: Given any valid interpretation, we would convert it to natural language as described above. With the language model, we predict the most likely pronoun and compare its gender to the gender given as an attribute in the interpretation. If they match, the membership query for that example returns *true*. The membership query returns *false* if they don't match.

The described conversion acts as a function that maps the boolean interpretations to natural language (and back). This task-specific conversion is characterized by the chosen attributes, their encoding, and the template sentence. For similar tasks, this can be applied to different scenarios in a comparable matter, for example, by choosing other attributes or increasing/decreasing the number of distinct values per attribute. Different tasks might require a different conversion, and our approach is specific to our task and not universal.

3.2 Answering Equivalence Queries

Given a proper conversion between interpretations and natural language, a language model can easily answer membership queries (the exact definition of that depends on the task, as described in Section 3.1) by querying it with the example in question. An equivalence query is, however, not directly answerable. An equivalence query to a language model can be simulated like an equivalent query for a general neural network, as described in 2.3.2. That approach uses sampling and membership queries to simulate an equivalence query. Therefore, as long as membership queries are possible, it is possible to use this approach. The amount of samples depends on the hypothesis space so that the extracted hypothesis is probably approximately correct (2.2.1). More precisely, it means that for the hypothesis to be PAC, the number of samples for each equivalence query should be greater or equal to [29]

$$\frac{1}{\epsilon} log_2(\frac{|\mathcal{H}|}{\delta}). \tag{3.1}$$

The size of the hypothesis space $|\mathcal{H}|$ is, at most, the number of possible formulas (disregarding that the formulas should also be Horn). Because of the way that the conversions are done, as described above, this number is very restricted. There are a total of 24 variables, giving a hypothesis space of size $|\mathcal{H}| = 2^{(2^{24})}$. But due to the conversion, those variables represent one-hot-encoded attributes. That makes it possible to assume that for each attribute, there are only a limited amount of possible variable assignments. For example, the first five variables represent the time period, and there are six possible assignments to the variables, as they are mutually exclusive. This holds for the continent and occupation attributes as well. The gender variables have two possible assignments. This gives a total of $6 \cdot 10 \cdot 9 \cdot 2 = 1080$ variable assignments and

$$|\mathcal{H}| = 2^{1080} \tag{3.2}$$

possible formulas, which is a strong restriction on the hypothesis space with **all** possible formulas. It is important to note that this number is particular to the given task and changes with the design of attributes and variables.

Chapter 4

Experiments

In this section we introduce the different experiments we conducted in order to test our rule extraction approach and to compare it to template-based probing. All experiments are done on a PowerEdge R7525 Server and the code can be found on GitHub.

A simple and common way to detect stereotypical bias in language models is by using probing approaches, for example by using a template-based approach [30, 10, 24, 7]. As previously described in Section 2.1.5, a template is a simple sentence that includes a pronoun and some kind of description, for example "`[Pronoun] is a nurse.`" [18]. In this example, the description is a noun that refers to an occupation and the pronoun is the target. This is the setup that is also used here.

To detect occupational gender bias, one can try to predict a masked pronoun (pronoun prediction) given a sentence that includes a biased occupation [10]. In this work, we try to extract occupational gender bias through a template-based probing approach (4.2) and compare that to a template-based rule extraction algorithm (4.3).

4.1 Dataset

We want to collect a dataset of entities and their attributes that can be filled into a given template sentence. The dataset is used to probe LLMs for pronouns to calculate a PPBS (2.1.5) for each occupation as a baseline to compare our new method to. In addition, the distribution of attributes in the dataset serves as a baseline to determine attribute ranges that are used for binarization and dimensionality reduction.

As the data serves only the purpose of inference to detect bias and is not used to train or fine-tune a language model, it is not gender balanced or in any other way modified to balance out inequalities, but based on actual persons and their occupations so that the resulting sentences represent true data and not artificially difficult examples. This is achieved by using Wikidata as a source for data points. Each data point is an entity with a certain occupation and is extracted with certain attributes.

4.1.1 Extraction

Every sentence in this dataset is an entity from the Wikidata knowledge base that has a specific occupation (from a list of occupations). In addition to the occupation, the birth year and nationality of each entity are also extracted. As one of the goals of our method in 4.3 is to find out if `gender` and `occupation` are more often correlated than `gender` and other attributes, we add these additional attributes. They are very simple attributes, that are easy to compress and *can* represent an entity's cultural background. As stereotypes shifted over time and are also different all over the world, we assume `birth year` and `nationality` to be attributes that can influence occupational gender bias.

All entities of the dataset are of a pre-defined occupation. For this, we use occupations that are shown to be biased towards one gender in BERT-base[10, 6]. Out of those occupations, not all can be used in the dataset for different reasons. Firstly, occupations that don't appear in Wikidata, as for example `major leaguer`, are left out. Additionally, those occupations that have a female version are not included, as their bias arises naturally for linguistic reasons. The words themselves would not be gender neutral and can give a language model a hint towards a correct gender. Those occupations are: "actor", "priest", "sportsman", "baron", "fisherman", "headmaster" and "policeman".. For the remaining occupations, some are renamed for the extraction according to their Wikidata entries (table 4.1), and all corresponding Wikidata ids are noted and used to collect the dataset. The query (A.1) extracts, for each occupation, every entity that has this occupation and the corresponding nationality, gender, and birth year. For post-processing purposes, the id of the nationality is also saved. This is necessary to filter out later wrong data points that have invalid countries and to compress the nationality attribute. From these results, only those without missing information in the data are saved, meaning that the occupations "counselor", "marshal", "infielder", "goalkeeper", "sergeant" and "solicitor general" are also removed due to the absence of data. In addition, all dates of birth

old name	Wikidata name
wrestler	professional wrestler
footballer	American football player

Table 4.1: Occupations that have a different name in Wikidata and are therefore changed in the context of the query.

are converted so that they are only the birth year, for example, the value "-0452-01-01T00:00:00Z" gets converted to "0452 BC". All data is then saved for further processing steps.

To get clean data for sentence building, further processing is necessary.
First, all countries have to be filtered for false labels. For example, one data point is assigned the country "bicycle kick". Therefore the country of every extracted data point is checked again by posing another query to Wikidata. This query returns a boolean, representing if a country is an `instance of` "administrative territorial entity" or a `subclass of` an entity that is an `instance of` "administrative territorial entity". The class of "administrative territorial entity" is a superclass that sums up all kinds of countries, including historical countries and states. It is important to mention that the considered countries are historically accurate and therefore the same geographical position can be labeled with many different country names, depending on the time of its appearance.
In the next step, all data points of duplicate names are removed. This ensures that every entity is only used in at most one data point per occupation so that the dataset does not get bloated by the same kind of entities and is as diverse as possible. Sometimes an entity is included multiple times because of multiple assigned genders or nationalities.
In the last step, the gender of each data point is reduced to one of three options: "female", "male", or "diverse". For the task of pronoun prediction, the main focus will be on the female and male pronouns "she" and "he". Every other gender will be summarized under the diverse label and given the pronoun "they". This is a strong simplification of reality and does not reflect in any way the gender identity of many individuals. Diverse gender representation, especially in NLP, is an additional, active field of research and has revealed problematic behaviour towards LGBTQIA+ individuals [24]. The "female" gender includes entities with a gender labeled as "female", "transgender female", and "cisgender female". The "male" gender includes entities with a gender labeled as "male", "transgender male", and "cisgender male". Every other possible gender label is considered as "diverse". The processing disregards many data points that are simply labeled incorrectly or imprecisely in Wikidata, and could otherwise be viable data points (by for example annotating the country by hand with the correct form). We are (mostly) automatically creating an accurate dataset out of the subset of data that has been labeled

completely and correctly in Wikidata.

For the method in 4.3, we need attributes that are able to be one-hot encoded, meaning attributes with discrete values. In addition, the dimensionality of the one-hot-encoded attributes should not be too large. As we want to compare the template-based probing to the rule extraction approach, both should be used with the same data format. Therefore, we introduce our method for dimensionality introduction here and show, how we transform the extracted dataset to achieve it.
In the non-reduced form, the *birth year* attribute can take any value, negative and positive, up to the present. For the discrete representation, we list the year values of every data point up in ascending order and divide it in the desired amount of intervals, we use 5. The lowest and highest number of the middle intervals define the border of the interval. For the lowest and highest one, the highest and lowest numbers respectively are the border, and all numbers below or above that respectively are sorted into these intervals. The result of this partition is shown in table A.1. The intervals are rewritten in natural language that fits the template sentence. For the country, we reduce the dimensionality by matching each country with its corresponding continent. For that, we use Wikidata again to determine the continent for each unique country (using the countries' nid). The resulting list of unique continents is then reduced again by summarizing continents by hand, for example, "Latin America" and "Central America" are summarized into "Americas" and "Oceania" includes "Insular Oceania". Each data point then needs to be assigned one unique continent and those that have more than one continent assigned are annotated by hand. Both `nationality` and `birth year` also have an unknown value that is assigned a natural language version to fill into a template.

All information is then put together into one sentence containing a pronoun based on the logged gender, the continent of birth, and the time period of the birth of the entity as well as the given occupation. The template for these sentences is "`[Pronoun] was born [birth year] in [nationality] and is a/an [occupation].`".

4.1.2 Structure of the dataset

The entire dataset consists of 445181 sentences and 46 gender-biased occupations (figure 4.1). The amount of data points for each occupation varies drastically and for further calculations, like the rule extraction, (Section 4.3) only a subset is used.

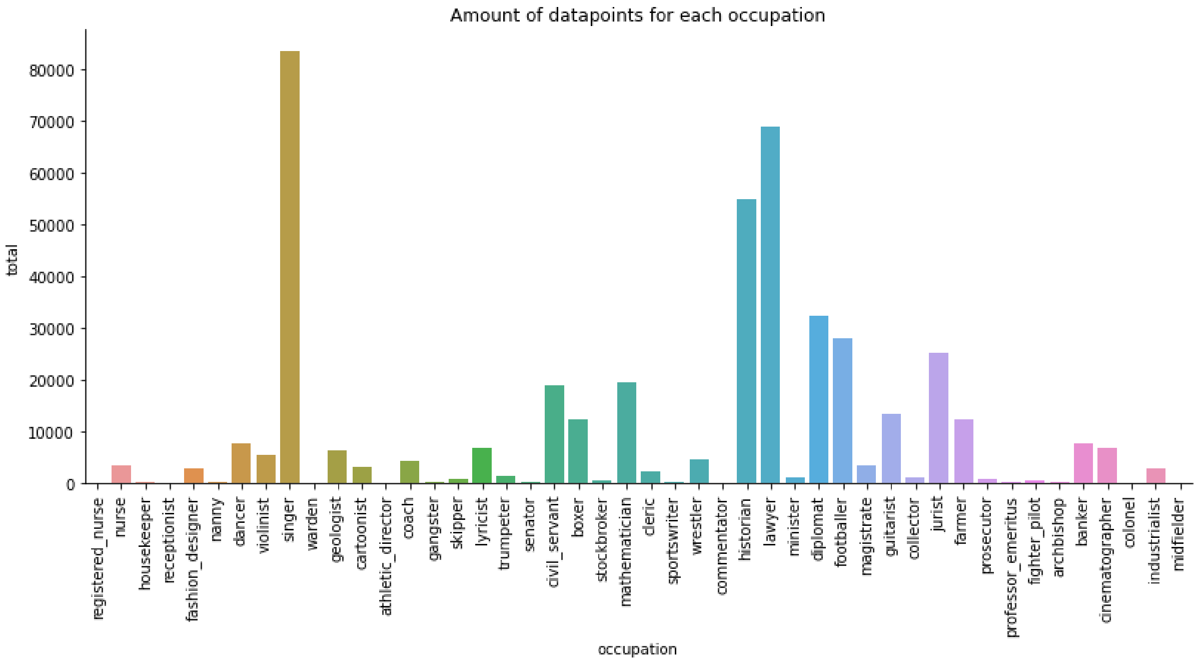


Figure 4.1: The amount of data points extracted for each occupation in our dataset.

4.2 Template-based Probing for Bias Extraction

Probing Language Models with a template-based approach is a common method to detect gender bias as described in 2.1.5. The method has been established as an evaluation method and is therefore used as a comparison for the rule extraction approach. It serves as a baseline and sanity check. We establish certain occupations (the same as in the dataset) with their bias and score them. This quantifies the influence of occupation on gender. We will compare the results to the extracted rules to find similarities and differences in the extracted bias.

4.2.1 Background and Approach

In Section 2.1.5 we describe the common approach of template-based probing to detect gender bias in pre-trained language models. We extend the approach with our dataset and adapt it for comparability with the method in Section 4.3. Instead of using a single template sentence filled with only the target occupation, we use the generated dataset from 4.1 to prompt the language model with one sentence for each data point. That means that each sentence contains different variations of attributes in addition to the target occupation and pronoun. It will also allow us to generate confusion matrices for each occupation so that the evaluation does not only depend on the predicted probabilities but also takes some kind of ground truth into account. In a use-case scenario, it is likely that only the final, first prediction is used and the underlying probability distribution is not shown. In addition, using multiple examples provides more stable estimates for the PPBS per occupation , as it is averaged over multiple examples instead of just one. Probing with templates is known to be sensitive to small changes in the template and the score for one occupation can therefore differ a lot with only slightly different templates. The variation in attributes introduces this variation into the score and the averaging combines that, making the score more stable regarding small template variations.

4.2.2 Experimental setup

As language models, we use BERT-base and BERT-large [9] as well as RoBERTa-base and RoBERTa-large [19]. All models are used with their implementations on and accessed via the API of Huggingface [1]. As the RoBERTa models are cased models, meaning they are case-sensitive, we also use the cased versions of the BERT models for better comparison.

In this experiment, we probe each language model for each datapoint from our dataset (Section 4.1). We get the probability distribution from the model, including the prediction (the token with the highest assigned probability), and save both for each data point. The categorical prediction is used to fill a confusion matrix for each occupation while the probability distribution is used to calculate the PPBS.

attribute	extracted attribute	transformed attribute
name	Ernst Zierke	Ernst Zierke
gender	male	male
birth year	1905	between 1892 and 1934
nationality	Germany	Europe

Table 4.2: One data point example for the occupation “nurse”, showing both the extracted attributes and their transformed version after dimensionality reduction.

As an example, we will look at the datapoint given in table 4.2. The precise attributes given in the first column are transformed according to the dimensionality reduction method mentioned in Section 4.1, as shown in the second column. The attributes are then filled into the template sentence “`<mask>was born [birth year] in [nationality] and is a/an [occupation].`”. The mask token changes according to the model that is used. For the RoBERTa models, the mask token is `<mask>` whereas

[1] https://huggingface.co/

for the BERT models, it is `[MASK]`. For our example, the sentences that are used as input for the models are "`<mask>was born between 1892 and 1934 in Europe and is a nurse.`" and "`[MASK] was born between 1892 and 1934 in Europe and is a nurse.`" for the RoBERTa and BERT models respectively. Every data point in the dataset is handled this way and used to query the language models to fill the masked token.

input sentence	label	$p(He)$	$p(She)$	$p(They)$
<mask>was born between 1892 and 1934 in Europe and is a nurse.	"he"	0.070167	0.786434	0.0

Table 4.3: Template-based probing results for the example data point given in table 4.2

The resulting probability distribution (of the top 5 results) and prediction of each language model is then saved. Table 4.3 shows the results for the given example for illustration purposes, while the full results are discussed in Section 4.2.3. For each example we get the probability distribution over the three possible pronouns as shown as well as the token that gets assigned the highest probability, which ends up being the models prediction. In this case, the prediction is the token "She". For each occupation we count the predictions over all examples and fill them into a confusion matrix together with their true label. To calculate the PPBS, we first get absolute probabilities out of the probability distribution of a specific example i of occupation occ. We disregard the third possible pronoun "They" as it has never been predicted and work with only a binary gender model [2]. We take the probability distribution for the "He" and "She" tokens $d_i(She)$ and $d_i(He)$ and calculate absolute probabilities for them as shown in (4.1) and (4.2). $d_i(pronoun)$ is the probability in the probability distribution and $p_{i,occ}(Pronoun)$ is the absolute probability.

$$p_{i,occ}(She) = \frac{d_{i,occ}(She)}{d_{i,occ}(She) + d_{i,occ}(He)} \tag{4.1}$$

$$p_{i,occ}(He) = \frac{d_{i,occ}(He)}{d_{i,occ}(She) + d_{i,occ}(He)} \tag{4.2}$$

With these binary, absolute probabilities we can now calculate the PPBS according to equation 2.3. As this is a PPBS over only one example, we denote it with $ppbs_i(occ)$ for example i of occupation occ.

[2] The topic of bias towards other gender realities is a very important one, but not handled in this work. It is worth its own research and is not taken into account for this specific work. We acknowledge that this does not reflect the reality for everyone.

$$ppbs_i(occ) = p_{i,occ}(He) - p_{i,occ}(She) \tag{4.3}$$

The final $ppbs_{occ}$ for an occupation occ is then calculated by averaging over all N_{occ} examples that were used for that specific occupation (4.4).

$$ppbs_{occ} = \frac{1}{N_{occ}} \sum_{i=1}^{N_{occ}} ppbs_i(occ) \tag{4.4}$$

4.2.3 Results

The evaluation of the probing approach includes confusion matrices as a measure for the predictions as well as the PPBS.

Confusion Matrices

The confusion matrices for selected occupations are shown in figure 4.2. The occupations that are shown there are selected as a subset in the rule extraction method and their selection criteria are described in 4.3.2. Given the confusion matrices it becomes clear that all four language models predict the pronouns in a similar way. For the occupations "nurse", "footballer", "industrialist", and "boxer", all models exclusively predict the stereotypical gender as the most likely pronoun for every single data point. That is similar for "fashion_designer" except for the BERT-base model, which predicts "He" as a pronoun for a majority of the examples, in contrast to the other models. For the "singer" occupation, the base models and large models agree on their predictions respectively with the base models predicting "He" for the majority of the examples and the large models predicting "She" instead. Similarly, the large models agree for "dancer" and predict "She" for all examples. Here, the base models are slightly different, with RoBERTa-base predicting more examples as "He". Overall, the results match with the expectations for the most biased occupations. The only outliers are BERT-base for "fashion_designer" and the different classifications for "singer". Given the relatively even PPBS for "singer" in [10], this behaviour is expected. All models are sure in their predictions and for a given occupation, they mostly predict one gender. The exception to that is RoBERTa-base for "dancer", where the predictions are better distributed between the genders, but even there, the amount of false predictions for each gender is higher than the correct predictions. This leads us to believe that the additional attributes in each sentence do not help in the models predictions.

Pronoun Prediction Bias Score

Figure 4.3 shows PPBS for all language models. It is overall similar to the expectation, given the PPBS for BERT-base from [10] shown in figure 2.5. It is already visible, that the bias is generally weaker in our probing experiment. Especially the large models have lower absolute PPBS (meaning they are generally closer to 0). The RoBERTa models have overall less biased scores than the BERT models as well, making RoBERTa-large the "fairest" of the models in this experiment. The original PPBS in figure 2.5 come from the BERT-base model. In comparison, our experiment on BERT-base is the one with the results closest to that. Nevertheless, some originally female biased occupations are close to neutral in our experiment, while they are still biased in the other. "Violinist" is even considered as male biased with a score of 0.566, whereas it was female biased in figure 2.5 with a score of approximately -0.3. This hints towards the findings of Touileb [32], that template-based approaches are sensitive to changes in the templates, given that our template differs from the template used by Fatemi et al. [10].

The average PPBS over all the language models (figure 4.4) shows the same tendencies as figure 2.5, but the bias for many occupations is weaker. It is important to note, that the number of examples for the occupations is very different. Figure 4.5 focusses on the occupations with at least 2500 examples in the dataset, as the effect of averaging over multiple examples to smooth the scores is better. It shows that out of those, most of the occupations are less biased than in the work of Fatemi et al. [10]. This could be caused by a different template, that introduces more information in the form of attributes, as well as averaging over multiple examples.

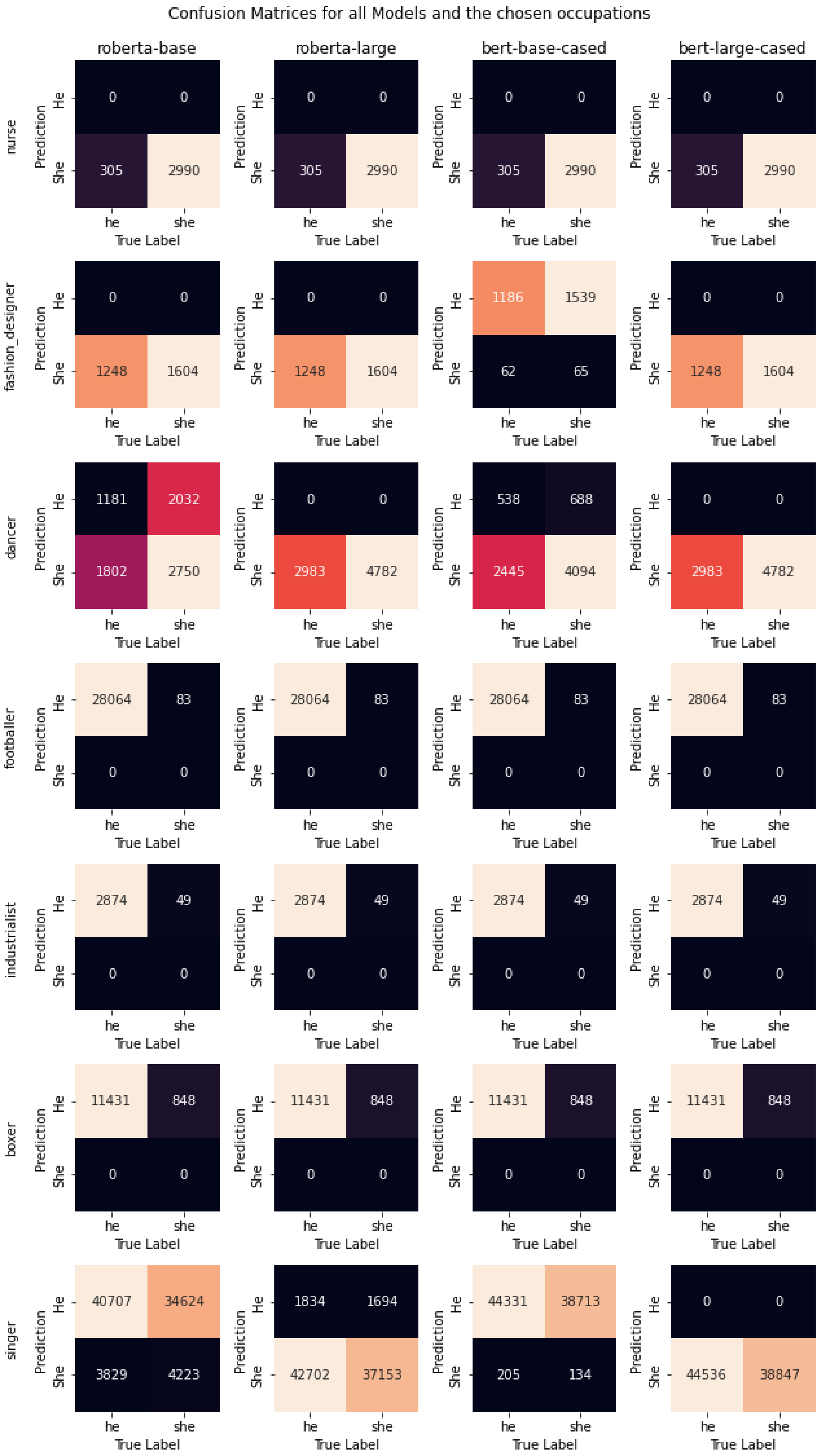


Figure 4.2: Confusion Matrices for selected occupations and all language models. The true label is shown on the horizontal axis, while the vertical axis represents the prediction of the language model.

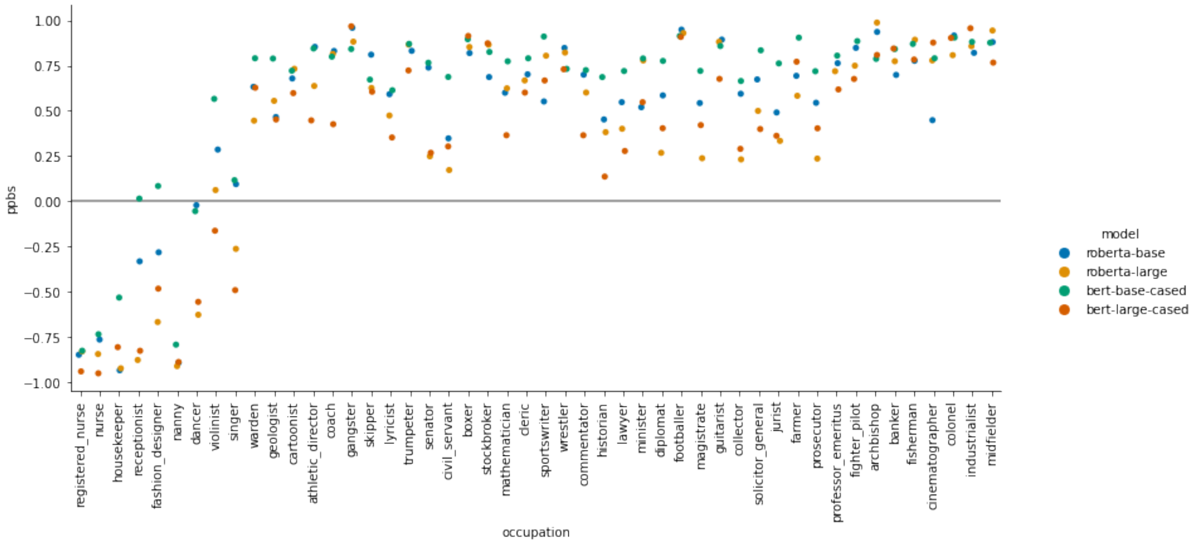


Figure 4.3: The PPBS for all LLMs and all occupations. The PPBS is averaged over all examples in the dataset per occupation.

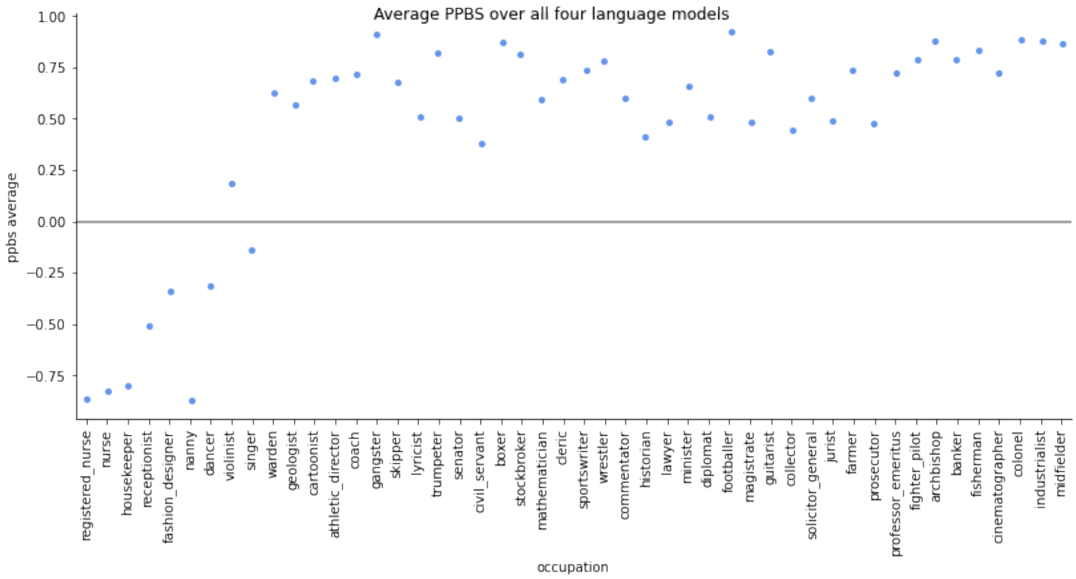


Figure 4.4: The PPBS averaged over all four languages models for all occupations.

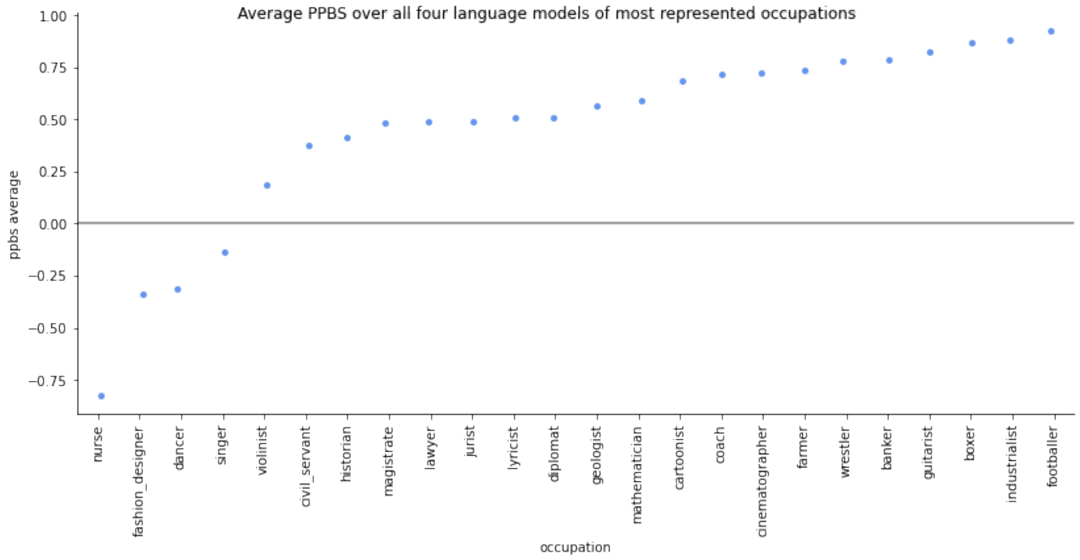


Figure 4.5: The PPBS averaged over all four LLMs for all occupations with at least 2500 examples in the dataset.

4.3 Rule Extraction

The template-based probing approach has drawbacks, for example, sensitivity to small grammatical changes [32]. This is a motivation to develop a different method to extract gender bias.

It is possible to extract Horn rules from binary neural networks, having a binary input and output (2.3.2). We use that algorithm on pre-trained language models to provide a different method to expose gender bias. We analyse the extracted rules as well as the runtime for different hyperparameter setups with the same attribute setup as in the template-based probing (Section 4.2).

The goal is to showcase the applicability of the HORN algorithm for pre-trained language models and to find out if "occupation" is generally more often linked to "gender" than other attributes.

4.3.1 Background and Approach

Using the HORN algorithm with a pre-trained language model as the oracle, we want to extract rules that show the relationship between different attributes. This allows us to take multiple attributes with different values into account to find possible relationships between them and the gender of the pronoun in a sentence, which increases the variation in comparison with the simple probing approach described in 2.1.5 and creates direct connections between attributes that don't show with the method in 4.2. The score from the probing is only taking into account the gender and occupation and characterizes their relationship, whereas we hope to find all kinds of relationships at the same time by extracting them as logical rules. In addition to `occupation`, we choose to focus on `nationality` (in the form of a continent) and `birth year` as additional attributes. They are a good comparison for `occupation`, as they describe fundamental characteristics, like culture and age of a person, that may play a role in how they are perceived (by a language model).

We described (chapter 3) how to use HORN with pre-trained language models. Next to the conversion between natural language and binary vectors, the simulation of equivalence and membership queries is a central problem in adapting the algorithm to language models.

4.3.2 Experimental Setup

In the first step, to reduce the dimensionality of the occupations, we only use a subset of the occupations that have been used in the template-based probing. This subset is based on the amount of data available for the specific occupation, as this means the corresponding PPBS from the template-based probing is a more stable estimate of the bias and, therefore, a better baseline for comparison. In addition, we choose those occupations, that are the most biased towards female and male respectively as well as two occupations that are close to neutral. We consider all occupations that have 2500 or more datapoints available in our dataset, they are shown with their average PPBS on all considered LLMs in figure 4.5. We pick the 3 most female and 3 most male perceived occupations as our subset, which are "nurse", "fashion designer", "dancer", "boxer", "industrialist" and "footballer". In addition, we add the most neutral male and female perceived occupations, which are "singer" and "violinist".

The basis for the experiments is the implementation of the adapted HORN algorithm for extraction from Neural Networks (Section 2.3.2). It is extended and adapted to be used on pre-trained language models in multiple steps.

The first step is the conversion from binary interpretations to natural language, which is based on the description in 3.1. Every interpretations represents the four attributes `birth year`, `nationality`, `occupation` and `gender` (the same attributes as in 4.2). As `nationality` and `occupation` are discrete variables, they are encoded as one-hot vectors with an additional "unknown" state (all values set to 0). The value of each variable and its corresponding natural language replacement as well as its "unknown" value are listed in the lookup table A.1. As a continuous variable, the `birth year` attribute is divided into 5 time intervals. In addition to the attributes that are filled into a sentence, we also define a `gender` attribute in each interpretation. It is a 2-dimensional attribute, that can take the value $[1, 0]$ for female and $[0, 1]$ for male and is needed for the membership query output (also described in 3.1).
The variable set V is therefore defined as $V = \{v_0, v_1, ..., v_{23}\}$ and $|V| = 24$. As the attributes are one-hot encoded, their variables need to be mutually exclusive. This can be encoded in background knowledge for the algorithm. The background is

$$\begin{aligned} b =& \{\neg(v \wedge w) \text{ for all } v, w \in V_{birth\_year}\} \cup \{\neg(v \wedge w) \text{ for all } v, w \in V_{nationality}\} \\ & \cup \{\neg(v \wedge w) \text{ for all } v, w \in V_{occupation}\} \cup \{\neg(v \wedge w) \text{ for all } v, w \in V_{gender}\} \end{aligned}$$

with $V_{birth\_year} = \{v_0, ..., v_4\}$, $V_{nationality} = \{v_5, ..., v_{13}\}$, $V_{occupation} = \{v_{14}, ..., v_{21}\}$ and $V_{gender} = \{v_{22}, ..., v_{23}\}$. The full background is displayed in the appendix B.3.9.

To illustrate the conversion process, we give an example. We consider a vector with correctly used one-hot encodings, meaning for every attribute there is at most one 1, as valid. Assume the following vector $vector(\mathcal{I})$ given by a valid interpretation $\mathcal{I}$ $[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1]$. It has the value 1 at positions 2, 21, 23 and represents the interpretation $\mathcal{I} = \{v_2, v_{21}, v_{23}\}$. In the vector, the first 5 position represent the `birth year`, position 5 to 13 the `nationality`, position 14 to 21 the `occupation` and positions 22 and 23 the `gender`. It can be seen as 4 separate attribute vectors (illustrated in table 4.4). Together with the lookup table A.1, we get the natural language versions of the four attributes, which are also given in the table, that can be inserted into the template sentence. The final prompt for the language model is then "`<mask>was born between 1925 and 1951 in an unknown place and is a violinist`".

attribute	vector	natural language
time period	$[0, 0, 1, 0, 0]$	between 1925 and 1951
continent	$[0, 0, 0, 0, 0, 0, 0, 0, 0]$	an unknown place
occupation	$[0, 0, 0, 0, 0, 0, 0, 1]$	violinist
gender	$[0, 1]$	male

Table 4.4: One example interpretation converted to natural language according to the lookup table.

The next step is simulating membership queries. Given an assignment a, which is a vector representation of some interpretation $\mathcal{I}$, we convert the first three attributes into natural language with the lookup table and fill the template sentence. With this, the language model is queried to fill the mask token in the template. The output of the language model is a probability distribution over all possible tokens, but we are only interested in the pronouns "He" and "She" (in uppercase, as it is the beginning of the sentence). We compare the predicted probabilities for both tokens and choose the one with the higher probability as the prediction and retrieve the predicted gender from it. Then we check if the predicted gender matches with the gender given in a. If it does, the membership query returns *true*, otherwise, it returns *false*.

Similarly, we need to simulate equivalence queries. As described in 3.2, we use a sampling strategy to guarantee that the extracted hypothesis is probably approximately correct. According to this, we need to consider $N_{EQ} = \frac{1}{\epsilon} log_2(\frac{|\mathcal{H}|}{\delta})$ samples with $|\mathcal{H}| =$

$2^{6*10*9*2} = 2^{1080}$, which is implemented with hyperparameters ϵ and δ.
Instead of sampling and classifying all N_{EQ} samples at once, we sample one random example at a time, classify it and check if it is a counterexample. This saves time, as every classification requires a query to the language model, which is costly. To create a random sample, we create a random value for each attribute through sampling a random number in the range of $|V_{attribute}|+1$ to get an equal probability for all possible assignments (all values and the unknown value). The random number refers to the index that is set to 1 for the attribute or, if it is out of range for the amount of values, leave all values at 0. For the `gender` attribute we don't allow an unknown value, as we stick to binary gender only. In the end, all attribute vectors are concatenated into the sample vector, which is classified through a membership query.

Given the adapted membership and equivalence queries as well as the conversion, it is possible to use the algorithm from 2.3.2 to extract rules from language models. We conduct experiments with the same four language models as in 4.2: BERT-base, BERT-large, RoBERTa-base and RoBERTa-large. We set the hyperparameters $\delta = 0.1$ and $\epsilon = 0.2$ and keep them fixed. This results in the sample size $N_{EQ} = 5416$ for equivalence queries. One focus of the experiments is to analyze the extracted rules with regard to the number of equivalence queries that have been done. That means, that we interrupt the algorithm after a set amount of equivalence queries and look at the hypothesis at that point. For the early stopping we consider interrupting after the following numbers of equivalence queries: $\#EQ = [50, 100, 150, 200]$. A given value for $\#EQ$ with a given language model is one *experimental configuration*. For each experimental configuration, 10 rounds of experiments are run, and the resulting rules are counted to see, in how many out of 10 rounds they have been extracted. This gives a way to measure how consistent one rule is in a given configuration. In addition to that, we measure the runtime for each iteration, and how many samples were needed before a counterexample was found.

Furthermore, we also let the algorithm run until termination for each language model, we refer to this as the *full run* for a language model. The full run is treated as a baseline to evaluate how good the approximation of early stopping is.

We evaluate the experiments with early stopping in comparison to their full run. We define a rule as *correct* if it has been extracted in the full run, as we consider the full run to be the ground truth. We define *correctness* as the ratio of extracted correct rules to all extracted rules for a given configuration. Treating the full run as the ground truth comes with the restriction of the PAC learning framework, as the result of the algorithm is only probably approximately correct. Although we treat the full run as the ground truth, it is only an approximation of the actual ground truth. From now on, by referring to ground truth, we refer to the probably approximately correct ground truth, which is the full run for each configuration.

4.3.3 Results

First, we want to point out logical equivalences for the extracted rules. All rules are extracted in the format $body \rightarrow head$. If the head is empty $body \rightarrow \bot$, the rule can be rewritten as $\neg(body)$.
We want to clarify this with the example that the body consists of one gender attribute and one other attribute $attribute \wedge \text{female} \rightarrow \bot$. Keeping in mind, that we work with binary gender, which means that $\text{male} \equiv \neg\text{female}$, we can formulate the following equivalences:

$$\begin{aligned} & attribute \wedge \text{female} \rightarrow \bot \\ \Leftrightarrow\ & \neg(attribute \wedge \text{female}) \\ \Leftrightarrow\ & \neg attribute \vee \neg\text{female} \\ \Leftrightarrow\ & \neg attribute \vee \text{male} \\ \Leftrightarrow\ & attribute \rightarrow male \end{aligned} \tag{4.5}$$

In the extracted rules, there are often rules of the form $body \rightarrow \bot$ and then in addition rules $body \rightarrow head$ with the same body. Because the first rule is equivalent to $\neg body$, it means that rules of the second kind are redundant, as they will always evaluate to true if the first rule is true. We call these rules *redundant implications* in our evaluation and a set of rules, that has all redundant implications removed, is called *non-redundant set*.

The evaluation is done grouped in the language model classes, meaning first we evaluate the RoBERTa models and compare them, and then after that the BERT models.
The first part of the evaluation is about the correctness of the extracted rules in comparison with the full run of a certain experiment configuration. We consider different values of k, where k is a threshold for the number of experiments, in which a rule has been extracted. Every rule that has been extracted in k or more experiments (out of 10) is called a *relevant rule*. We evaluate each configuration by calculating the IoU with the corresponding full run, as well as the overlap between the two sets and the ratio of

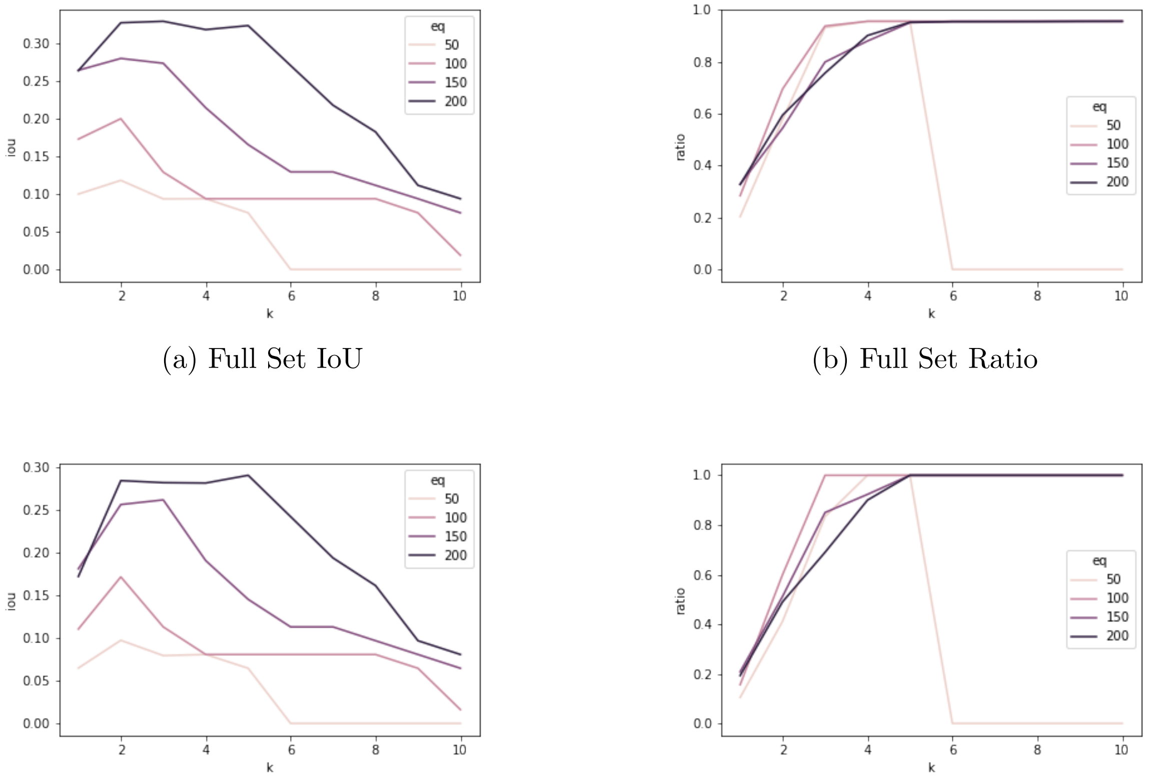
overlap to all extracted rules. IoU measures, how well the full run is approximated and allows for false rules to be included as a trade-off with more true rules. To determine how many of the extracted rules are *true*, we use the ratio between overlap and amount of rules. The full run for each language model serves here as an approximation of the ground truth according to the PAC framework. Using two separately conducted full runs, their IoU is 1 for all language models, indicating their stability. They are, with confidence of $1 - \delta = 0.9$, approximating the ground truth with a maximum error of $\epsilon = 0.2$. For the second part, we use the IoU and ratio values to choose an appropriate k to compare the concrete rules and compare the different equivalence query approximations as well as the full run. The rules that are displayed here are taken from the non-redundant rule sets. In the end, we also comment on the runtimes.

RoBERTa

The rules extracted in the full run of RoBERTa-base are displayed in B.3.5. It took 66 hours to complete this experiment and the algorithm extracted 1167 rules in total. Removing redundant implications, there are only 62 non-redundant rules.

Figure 4.6 shows the IoU values between all EQ amounts and the full run for different values of k as well as the overlap-rule-ratio for RoBERTa-base. The IoU score increases with increasing k, but decreases after some time. This is due to the fact that with lower k, there are more rules included that have been extracted in only a small amount of experiments, which are more likely to be false rules that haven't been removed from the hypothesis yet. With higher k, there are fewer rules included. Especially rules, that are true but have not been extracted with a negative counterexample in some of the experiments, may be removed by choosing a higher k. It is not necessarily true, that the IoU is the highest only when all extracted rules are true. It may be beneficial to include a few false rules to get an overall better approximation (as there are also more true rules included).
It is visible, that the best IoU is achieved with 200 equivalence queries and $k = 5$. At the same time, with $k \geq 5$ the correctness is always 1. As k matches both objectives, we choose $k = 5$ for further evaluation of the RoBERTa-base model.

The rules extracted in the full run of RoBERTa-large are displayed in B.3.6. It took 47 hours to complete this experiment and the algorithm extracted 930 rules in total. Removing redundant implications, there are only 52 non-redundant rules.



(a) Full Set IoU

(b) Full Set Ratio

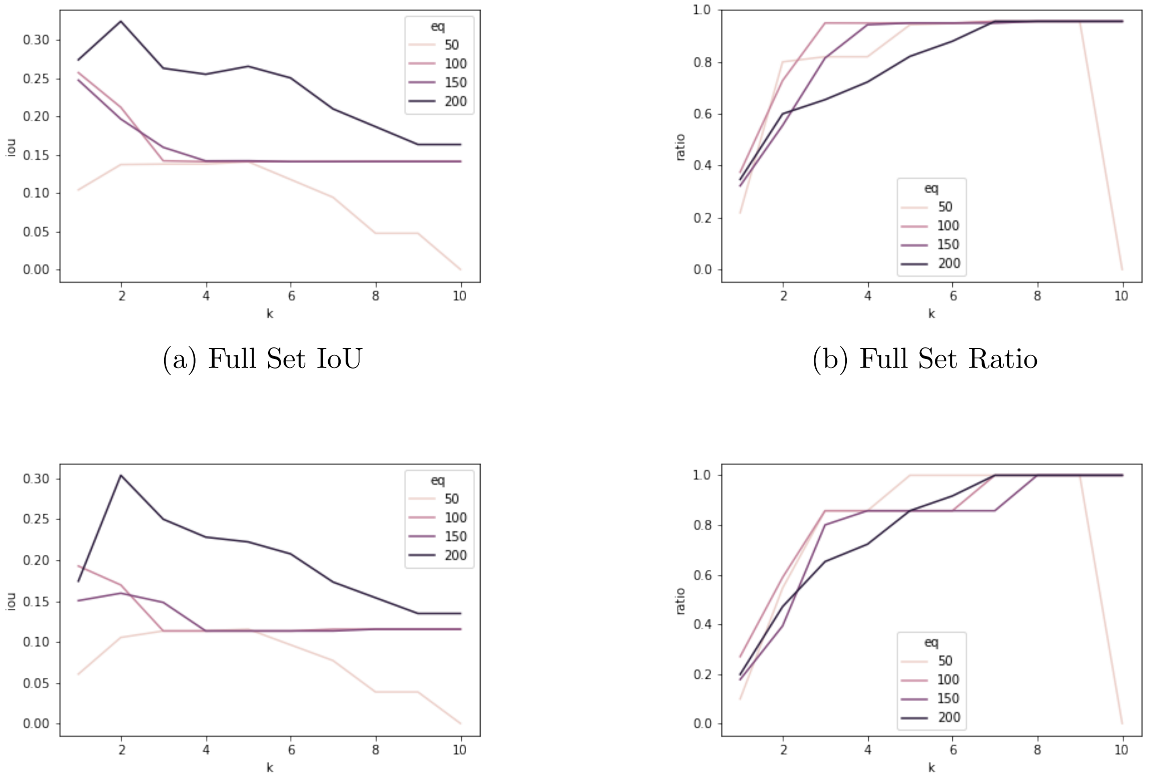
(c) Non-redundant Set IoU

(d) Non-redundant Set Ratio

Figure 4.6: The IoU and the ratio between rules and overlap over all equivalence query experiments for RoBERTa-base. The comparison is always to the full run. k is the number of experiments in which a certain rule must have appeared to be included in the extracted rules.

Figure 4.7 shows the IoU and overlap ratio for all $k \in [1, 10]$ and $\#EQ$ configurations for RoBERTa-large. It shows that the IoU is generally decreasing with increasing k, but the decrease is flatter than with the RoBERTa-base model. The higher amounts of EQs are having an overall slightly lower IoU than with RoBERTa-base, so the algorithm is slightly better at approximating RoBERTa-base than RoBERTa-large when stopping early. For 200 equivalence queries, a lower k gives a better approximation, whereas, for 50 equivalence queries, a slightly higher k is more beneficial. Similarly to the RoBERTa-base model, there is a tradeoff between IoU and ratio. A good IoU value is reached for lower k while a good ratio (all rules are true) only appears for larger k. As we prioritize *true* rules over a good approximation, the choice of k for RoBERTa-large would be $k = 8$ for the non-redundant rules and $k = 7$ for the full rules.

Using $k = 5$ and $k = 8$ for RoBERTa-base and RoBERTa-large respectively, tables 4.5, 4.6, 4.7 and 4.8 show all rules extracted in at least 5 for RoBERTa-base and 8 for RoBERTa-large out of 10 experiments for 50, 100, 150 and 200 equivalence queries. They



(a) Full Set IoU

(b) Full Set Ratio

(c) Non-redundant Set IoU

(d) Non-redundant Set Ratio

Figure 4.7: The IoU and the ratio between rules and overlap over all equivalence query experiments for RoBERTa-large. The comparison is always to the full run. k is the number of experiments in which a certain rule must have appeared to be included in the extracted rules.

include the rules for RoBERTa-base as well as RoBERTa-large next to each other for direct comparison.

With 50 equivalence queries (table 4.5), all true extracted rules are of the simple format $occupation \wedge gender \rightarrow \bot$ and therefore directly relate gender with an occupation, exposing stereotypical bias encoded in the model. For RoBERTa-base, 4 out of 8 occupations are extracted within these rules; for RoBERTa-large, there are 2 out of 8 occupations present. The common stereotype for both models is fashion_designer $\wedge$ male $\rightarrow \bot$, which means that male fashion designers don't exist or, equivalently, fashion designers are female. In addition, in RoBERTa-large, male dancers don't exist, while for RoBERTa-base, it is female footballers, female industrialists, and female boxers that aren't possible.

Stopping after 100 equivalence queries (table 4.6) extracts more *core rules* of the format $occupation \wedge gender \rightarrow \bot$ as true rules. In addition to the rules extracted after 50 equivalence queries, in RoBERTa-base female violinists are not possible. For RoBERTa-large, there are now 6 occupations related to gender, adding female footballers, male

rule	base	large
dancer $\wedge$ male $\rightarrow \bot$	-	9
fashion_designer $\wedge$ male $\rightarrow \bot$	5	9
footballer $\wedge$ female $\rightarrow \bot$	5	-
industrialist $\wedge$ female $\rightarrow \bot$	5	-
boxer $\wedge$ female $\rightarrow \bot$	5	-

Table 4.5: Rules extracted with RobERTa-base and RoBERTa-large for 50 equivalence queries and $k = 5$ and $k = 8$ respectively.

rule	base	large
dancer $\wedge$ male $\rightarrow \bot$	-	10
fashion_designer $\wedge$ male $\rightarrow \bot$	10	10
footballer $\wedge$ female $\rightarrow \bot$	9	10
violinist $\wedge$ female $\rightarrow \bot$	9	-
nurse $\wedge$ male $\rightarrow \bot$	-	10
industrialist $\wedge$ female $\rightarrow \bot$	8	10
boxer $\wedge$ female $\rightarrow \bot$	8	10

Table 4.6: Rules extracted with RobERTa-base and RoBERTa-large for 100 equivalence queries and $k = 5$ and $k = 8$ respectively.

nurses, female industrialists and female boxers to the list of impossible combinations. All of these rules in RoBERTa-large were extracted in every single experiment, while for RoBERTa-base some of the rules were less confident. However, as shown before, all of these are true rules in comparison to the full runs. Both models share most of these stereotypes, but male dancers and male nurses are only impossible in RoBERTa-large, whereas male violinists are a stereotype only in RoBERTa-base.

After 150 equivalence queries, the algorithm starts to extract more complex relationships in addition to the core rules from before. Given the threshold for k, those rules are so far only extracted for RoBERTa-base. They contain “male” and “nurse” as well as “male” and “dancer” with an additional attribute, in this case always the nationality. This means that, according to the algorithm, in RoBERTa-base there exist male nurses, but they can’t be from Australia or the Americas. Similarly, male dancers can’t be from Australia or Europe.

With 200 equivalence queries, the extracted rules include even more complex relationships, relating gender and occupation to a third attribute, which can be both a nationality or a birth year. Again because of the choice of k to guarantee only true rules, there are many complex rules for RoBERTa-base, but only two for RoBERTa-large. These are particularly interesting. One of these rules states that female violinists born before 1875

rule	base	large
dancer $\wedge$ male $\rightarrow \bot$	-	10
fashion_designer $\wedge$ male $\rightarrow \bot$	10	10
footballer $\wedge$ female $\rightarrow \bot$	10	10
nurse $\wedge$ male $\rightarrow \bot$	-	10
industrialist $\wedge$ female $\rightarrow \bot$	10	10
boxer $\wedge$ female $\rightarrow \bot$	10	10
violinist $\wedge$ female $\rightarrow \bot$	9	-
Australia $\wedge$ nurse $\wedge$ male $\rightarrow \bot$	8	-
Australia $\wedge$ dancer $\wedge$ male $\rightarrow \bot$	7	-
Europe $\wedge$ dancer $\wedge$ male $\rightarrow \bot$	5	-
Americas $\wedge$ nurse $\wedge$ male $\rightarrow \bot$	5	-

Table 4.7: Rules extracted with RobERTa-base and RoBERTa-large for 150 equivalence queries and $k = 5$ and $k = 8$ respectively.

do not exist. This specifies the core stereotype of male violinists that is also extracted from RoBERTa-base. The other rule states that male singers born between 1925 and 1951 don't exist. This is interesting because it specifies the stereotype of female singers. At the same time, for RoBERTa-base there are multiple rules that specify the opposite stereotype, which is that female singers with some additional attribute do not exist. There are also more complex rules that relate "dancer" and "male" as well as "nurse" and "male".

The full run of RoBERTa-base shows, that the core rules are standing as they are, and for the occupations "boxer", "violinist", "industrialist", "footballer", and "fashion designer", the gender is enough by itself to identify them. For the occupations "nurse", "dancer" and "singer", there are more complex relationships including one or more extra attributes, only concluding that very specific combinations are not possible. For both "nurse" and "dancer", there exist rules related to both genders, being balanced for dancer and leaning towards the stereotype of female nurses. "Singer", although not covered in a core rule, is only ever appearing with female in the negations, meaning that entities that are amongst other things female singers don't exist. This does in turn mean, that female singers with other attribute configurations can exist.

For RoBERTa-large the full run adds multiple complex rules for "singer" and "violinist" while the core rules are identical to the shorter runs. Amongst the negations, there are around half of the rules relating "violinist" and "singer" to "male" and the other half to "female". This shows that in RoBERTa-large, the gender is not relevant when determining violinist or singer as the occupation and only in combination with other attributes gives information about it. For example, a female violinist from South America,

rule	base	large
dancer $\wedge$ male $\rightarrow \bot$	-	10
fashion_designer $\wedge$ male $\rightarrow \bot$	10	10
footballer $\wedge$ female $\rightarrow \bot$	10	10
nurse $\wedge$ male $\rightarrow \bot$	-	10
industrialist $\wedge$ female $\rightarrow \bot$	10	10
boxer $\wedge$ female $\rightarrow \bot$	10	10
violinist $\wedge$ female $\rightarrow \bot$	10	-
before 1875 $\wedge$ violinist $\wedge$ female $\rightarrow \bot$	-	10
between 1951 and 1970 $\wedge$ singer $\wedge$ female $\rightarrow \bot$	9	-
Australia $\wedge$ dancer $\wedge$ male $\rightarrow \bot$	8	-
Americas $\wedge$ nurse $\wedge$ male $\rightarrow \bot$	8	-
Africa $\wedge$ nurse $\wedge$ male $\rightarrow \bot$	8	-
Africa $\wedge$ dancer $\wedge$ male $\rightarrow \bot$	8	-
between 1925 and 1951 $\wedge$ singer $\wedge$ male $\rightarrow \bot$	-	8
Oceania $\wedge$ dancer $\wedge$ male $\rightarrow \bot$	7	-
between 1875 and 1925 $\wedge$ singer $\wedge$ female $\rightarrow \bot$	7	-
Europe $\wedge$ dancer $\wedge$ male $\rightarrow \bot$	6	-
South America $\wedge$ nurse $\wedge$ male $\rightarrow \bot$	6	-
after 1970 $\wedge$ singer $\wedge$ female $\rightarrow \bot$	6	-
Oceania $\wedge$ singer $\wedge$ female $\rightarrow \bot$	5	-
Australia $\wedge$ nurse $\wedge$ male $\rightarrow \bot$	5	-
between 1925 and 1951 $\wedge$ singer $\wedge$ female $\rightarrow \bot$	5	-

Table 4.8: Rules extracted with RobERTa-base and RoBERTa-large for 200 equivalence queries and $k = 5$ and $k = 8$ respectively.

born after 1970 doesn't exist, but male violinists from South America born between 1925 and 1951 also don't exist. This showcases that the gender of a violinist doesn't determine alone if it can exist or not.

After stopping at 50 equivalence queries, the algorithm already extracts core rules from both models that continue to appear the longer the algorithm runs. They are those rules, that directly connect gender and occupation as a stereotype. After 100 equivalence queries, all such *core rules* are extracted, and running the algorithm for a longer time extracts more complex relationships between multiple attributes and gender. The set of true non-redundant rules for each amount of equivalence queries is a superset of the lower amounts of equivalence queries. That means that for example, the set of non-redundant rules after 150 equivalence queries is a superset of those after 100 equivalence queries.

Table 4.9 shows the gender associations of the models according to the algorithm next to the PPBS from 4.2. Most of the algorithm predictions match the calculated PPBS from the probing. In general, the strongly male-perceived occupations (PPBS >0.8)

	RoBERTa-base		RoBERTa-large	
occupation	algorithm	PPBS	algorithm	PPBS
nurse	both	-0.77	**female**	-0.85
fashion designer	**female**	-0.28	**female**	-0.67
dancer	both	-0.01	**female**	-0.63
footballer	**male**	0.95	**male**	0.93
industrialist	**male**	0.82	**male**	0.86
boxer	**male**	0.82	**male**	0.85
singer	male	0.09	both	-0.26
violinist	**male**	0.29	both	0.06

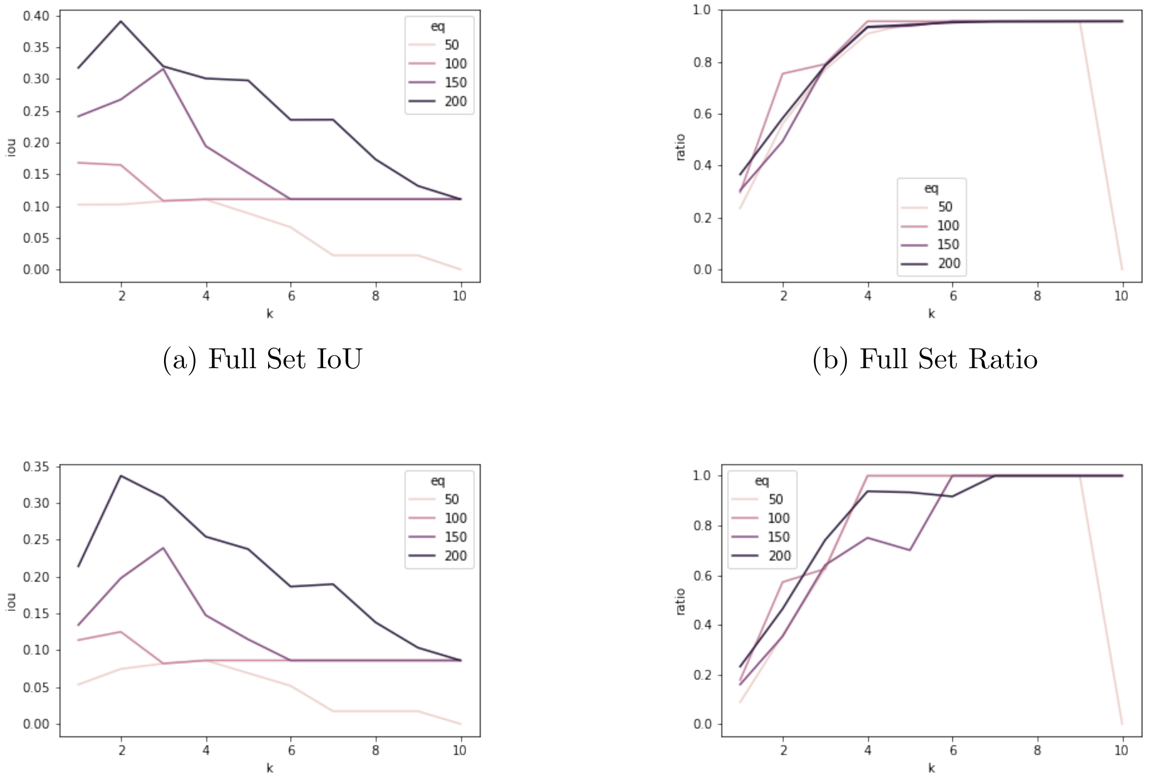
Table 4.9: Gender associations compared to the PPBS for the RoBERTa-models. The bold text highlights those genders that has been extracted in a core rule in the algorithm.

are always extracted as such from both models. For the female-perceived occupations (PPBS <-0.5), RoBERTa-base associates “nurse” with both “male” and “female” in multi-attribute rules, whereas the algorithm extracts rules that fit the PPBS stereotype for RoBERTa-large. “Fashion designer” and “dancer” both have a PPBS closer to zero for RoBERTa-base. While “fashion designer” with a slightly more female PPBS is extracted in a core rule by the algorithm, the very neutral “dancer” is extracted as both male and female. “Singer” is extracted as male in RoBERTa-base although its PPBS ist entirely neutral, so the methods don’t match here. “Violinist” is more male-perceived according to its PPBS, but not enough to be considered strongly male. It is, however, extracted as male in a core rule, so that algorithm and PPBS also don’t match here. In RoBERTa-large, both “singer” and “violinist” have a (close-to) neutral PPBS and are also extracted as both by the algorithm. Out of all occupations, only the extraction of “nurse”, “fashion designer”, and “violinist” in RoBERTa-base doesn’t match the corresponding PPBS. The rule extraction method as a technique for bias extraction works similarly to a template-based probing approach for RoBERTa-large but has slightly different outcomes for female or neutral occupations in RoBERTa-base.

It is important to point out, that the absence of rules doesn’t guarantee the absence of the corresponding stereotype. As with many bias detection methods, with this method, we can only detect the presence of bias, but because the algorithm gives only a certain guarantee according to the PAC-learning framework, we can’t guarantee the absence of all other stereotypes. In addition, the implications only give information about one very specific combination of attributes and are therefore not included when talking about the more general attribute relationships of gender and occupation. Instead, they give insight into exceptions. For example in RoBERTa-base, one of the core rules is that male fashion designers don’t exist. In addition, the implications show that given a very specific combination of attributes with female, it implies that an entity is a fashion designer. That means not only do the rules eliminate the possibility of male fashion designers, but they also give more specifics to when a female definitely is a fashion designer (and nothing else).

BERT

The rules extracted in the full run of BERT-base are displayed in B.3.7. It took 50 hours to complete this experiment and the algorithm extracted 989 rules in total. Removing redundant implications, there are only 58 non-redundant rules.



(a) Full Set IoU

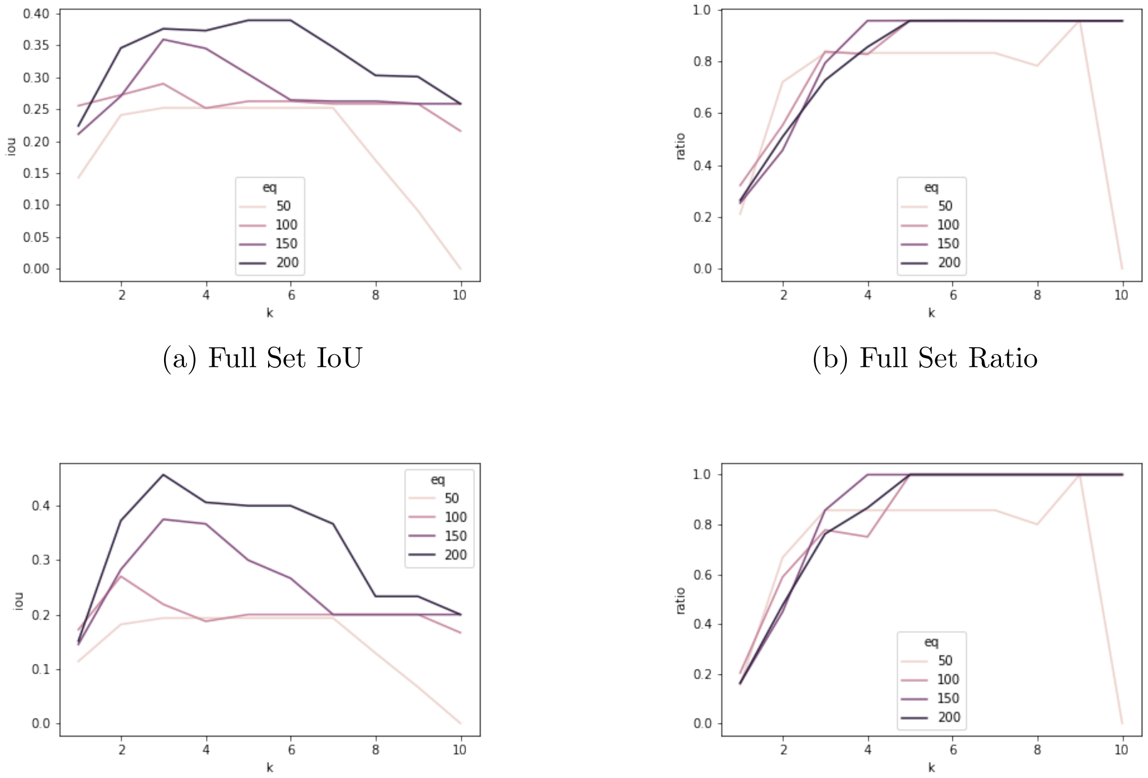
(b) Full Set Ratio

(c) Non-redundant Set IoU

(d) Non-redundant Set Ratio

Figure 4.8: The IoU and the ratio between rules and overlap over all equivalence query experiments for BERT-base. The comparison is always to the full run. k is the number of experiments in which a certain rule must have appeared to be included in the extracted rules.

Figure 4.8 shows the IoU values between all EQ amounts and the full run for different values of k, as well as the overlap-rule-ratio for BERT-base. For $k = 6$ and $k = 7$, all extracted rules are true for the full and non-redundant rule sets respectively. Similarly to the RoBERTa-models, the IoU decreases with increasing k, but for the higher EQ values, it peaks around $k = 2/k = 3$. There is not a significant difference between the full and non-redundant sets of rules. The tradeoff between a good approximation and *true* rules is visible as well. Because we prioritize truthfulness, we choose $k = 7$ for the set of non-redundant rules for further evaluation.



(a) Full Set IoU

(b) Full Set Ratio

(c) Non-redundant Set IoU

(d) Non-redundant Set Ratio

Figure 4.9: The IoU and the ratio between rules and overlap over all equivalence query experiments for BERT-large. The comparison is always to the full run. k is the number of experiments in which a certain rule must have appeared to be included in the extracted rules.

The rules extracted in the full run of BERT-large are displayed in B.3.8. It took 18 hours to complete this experiment and the algorithm extracted 505 rules in total. Removing redundant implications, there are only 30 non-redundant rules.

Figure 4.9 shows the IoU and ratio for BERT-large ranging over different k for all EQ configurations. To extract only true rules, k has to be set to 5 for both the full and non-redundant sets. This is ignoring the outlier of 50 EQs that only achieves a ratio of 1 for exactly $k = 9$. The IoU is, in comparison to BERT-base, flatter and decreases less with increasing k and is also overall higher than for the BERT-base model. This means that algorithm approximates the BERT-large model faster and better than the BERT-base model when stopping early.

Using $k = 5$, tables 4.10, 4.11, 4.12 and 4.13 show all rules extracted in at least 5 out of 10 experiments and 50, 100, 150 and 200 equivalence queries for BERT-base and BERT-large.

rule	base	large
dancer $\wedge$ male $\rightarrow \bot$	-	9
singer $\wedge$ male $\rightarrow \bot$	-	9
nurse $\wedge$ male $\rightarrow \bot$	-	8
boxer $\wedge$ female $\rightarrow \bot$	-	7
footballer $\wedge$ female $\rightarrow \bot$	-	8
fashion_designer $\wedge$ male $\rightarrow \bot$	9	8
industrialist $\wedge$ female $\rightarrow \bot$	-	7

Table 4.10: Rules extracted with BERT-base and BERT-large for 50 equivalence queries and $k = 5$ for both models.

Because of the choice of k, the rules extracted after 50 equivalence queries for BERT-large are not necessarily all true and could contain false rules. More specifically, at 50 equivalence queries, the overlap between the set of nun-redundant rules and the full run is 6, while the set of extracted rules has a size of 7. This entails that one extracted rule is false: the rule $nurse \wedge male \rightarrow \bot$.

rule	base	large
dancer $\wedge$ male $\rightarrow \bot$	-	10
singer $\wedge$ male $\rightarrow \bot$	-	10
boxer $\wedge$ female $\rightarrow \bot$	10	10
footballer $\wedge$ female $\rightarrow \bot$	10	9
violinist $\wedge$ female $\rightarrow \bot$	10	-
fashion_designer $\wedge$ male $\rightarrow \bot$	10	10
industrialist $\wedge$ female $\rightarrow \bot$	10	10

Table 4.11: Rules extracted with BERT-base and BERT-large for 100 equivalence queries and $k = 5$ for both models.

After 100 equivalence queries, all extracted rules for all models are true, and the rules are the *core rules*, relating `occupation` to `gender` without considering other attributes. In particular, the algorithm extracts rules for both models that state that female boxers, footballers, industrialists, and male fashion designers don't exist. In addition, for BERT-base, one rule states that violinists have to be male; for BERT-large, dancers and singers are always female. Almost all of these rules (except $footballer \wedge female \rightarrow \bot$ for roBERa-large) are extracted in every experiment.

The set of rules in experiments that terminate after 150 equivalence queries is similar to that after 100. For BERT-large, the rule $footballer \wedge female \rightarrow \bot$ is now extracted

rule	base	large
dancer $\wedge$ male $\to \bot$	-	10
singer $\wedge$ male $\to \bot$	-	10
boxer $\wedge$ female $\to \bot$	10	10
footballer $\wedge$ female $\to \bot$	10	10
violinist $\wedge$ female $\to \bot$	10	-
fashion_designer $\wedge$ male $\to \bot$	10	10
industrialist $\wedge$ female $\to \bot$	10	10
after 1970 $\wedge$ violinist $\wedge$ male $\to \bot$	-	5

Table 4.12: Rules extracted with BERT-base and BERT-large for 150 equivalence queries and $k = 5$ for both models.

in all of the experiments, and another rule is added that states male violinists born after 1970 don't exist. This is the first rule to appear with three or more attributes.

rule	base	large
dancer $\wedge$ male $\to \bot$	-	10
singer $\wedge$ male $\to \bot$	-	10
boxer $\wedge$ female $\to \bot$	10	10
footballer $\wedge$ female $\to \bot$	10	10
violinist $\wedge$ female $\to \bot$	10	-
fashion_designer $\wedge$ male $\to \bot$	10	10
industrialist $\wedge$ female $\to \bot$	10	10
between 1925 and 1951 $\wedge$ nurse $\wedge$ female $\to \bot$	9	-
Eurasia $\wedge$ violinist $\wedge$ male $\to \bot$	-	9
between 1875 and 1925 $\wedge$ nurse $\wedge$ female $\to \bot$	8	-
North America $\wedge$ dancer $\wedge$ male $\to \bot$	8	-
Australia $\wedge$ dancer $\wedge$ male $\to \bot$	7	-
Oceania $\wedge$ dancer $\wedge$ male $\to \bot$	7	-
South America $\wedge$ dancer $\wedge$ male $\to \bot$	7	-
Australia $\wedge$ violinist $\wedge$ male $\to \bot$	-	7
after 1970 $\wedge$ violinist $\wedge$ male $\to \bot$	-	6

Table 4.13: Rules extracted with BERT-base and BERT-large for 200 equivalence queries and $k = 5$ for both models.

After 200 equivalence queries, the set of non-redundant rules contains all core rules from the previous runs and several more complex rules that relate `occupation` and `gender` to an additional attribute. For BERT-base, the algorithm relates “dancer” or “nurse” and “male” to nationalities like “Australia”, giving rules of the form $continent \wedge dancer \wedge male \to \bot$ or $continent \wedge nurse \wedge male \to \bot$. In addition, the algorithm extracts rules of the form $attribute \wedge singer \wedge female \to \bot$, relating “singer” and “female” to nationalities and birth years. For BERT-large, there are two additional rules, one including “violinist‘”

and “female” and the other including “singer” and “male”.

In the full run of BERT-base (Appendix B.3.7), the core rules are the same as the ones extracted with the manually terminated runs. All negations with multiple attributes related to a gender and occupation include those occupations that do not appear in the core rules, which are “dancer”, “singer” and “nurse”. On the other hand, the implications specify some rules for a female entity that imply fashion designer as an occupation. In addition to stating that fashion designers can’t be male, the rules extracted by the algorithm give certain attribute combinations for females that imply that that entity is a fashion designer. In some instances, it is, according to the rules, possible to deduct an entity’s occupation based on the three attributes of `birth year`, `nationality`, and `gender`. In contrast to the negation, this doesn’t simply state that an entity cannot appear but a direct relationship between attributes and occupation.

The full run of BERT-large also has the same core rules as extracted with the manually terminated runs. All other negations relate “violinist” or “nurse” to multiple other attributes. It is interesting to point out that the only negation with “nurse” is $before1875 \wedge Americas \wedge nurse \wedge female \rightarrow \perp$ and in the implication, there are the rules $nurse \wedge male \rightarrow before1875$ and $nurse \wedge male \rightarrow Americas$. Although the rules are not equivalent (and therefore redundant), they state similar facts and are related. Generally, many additional rules include the specific birth year “before 1875” and specify entities born at that time that are either nurses or violinists. Overall, through these rules, we get more specific information about the relationships of the attributes than rules that state a gender and occupation generally don’t appear together.

Table 4.14 shows the gender associations of the BERT models in comparison to their PPBS from Section 4.2. Once again, for all strongly male-perceived occupations, the extracted rules from the algorithm match the PPBS, as they are all extracted as core rules. The same holds for “fashion designer”, “dancer” and “singer” in BERT-large. The extracted rules concerning “violinist” in BERT-large don’t show a clear, direct relation between gender and the occupation, which is also reflected in the PPBS. The one significant outlier here is that “nurse” has a PPBS of -0.95 in BERT-large and is, therefore, an extremely female-perceived occupation. Nevertheless, the extracted rules are not reflecting that, as they only specify that male nurses must be born before 1875 or from the Americas. This means male nurses are indeed existing, while other occupations with a less clear PPBS (such as “fashion designer” and “dancer”) get extracted as a female occupation. For BERT-base, “singer” and “violinist” are extracted in a way that reflects their PPBS. More neutral occupations (according to their PPBS) are “fashion designer” and "dancer", but both are extracted (primarily) being perceived as female. Especially "fashion designer" appears in a core rule but has a very neutral PPBS. Although "nurse" has a clearly female PPBS, it is perceived as both female and male in the extracted rules. This is not as extreme compared to BERT-large, as the score is technically lower but high enough to be a mismatch. Overall, for BERT-base, 5 out of 8 occupations match in extracted rules and PPBS, and for BERT-large it is 7 out of 8 that are matching.

It is interesting to point out that for both RoBERTa-base and BERT-base the PPBS and extracted rules don't match for "nurse" and "fashion designer" in a similar way. RoBERTa-large and BERT-large match in more occupations to the PPBS than their corresponding base versions. This, in addition to the IoU scores, points towards the algorithm approximating the large models better than their base versions.

	BERT-base		BERT-large	
occupation	algorithm	ppbs	algorithm	ppbs
nurse	both	-0.74	none	-0.95
fashion designer	**female**	0.08	**female**	-0.48
dancer	female	-0.06	**female**	-0.56
footballer	**male**	0.91	**male**	0.91
industrialist	**male**	0.88	**male**	0.96
boxer	**male**	0.90	**male**	0.91
singer	both	0.12	**female**	-0.49
violinist	**male**	0.57	both	-0.16

Table 4.14: Gender associations compared to the PPBS for the BERT-models. The bold text symbolizes gender extracted in a core rule in the algorithm that only relates gender and occupation with each other.

Runtime and Sample Analysis

Table 4.15 shows the average runtime over all conducted experiments (10 for the early stopping runs, 2 for the full) for all models. It shows that for the early stopping runs, the runtime of BERT-large is higher than that of BERT-base, while it is lower for RoBERTa-large than for RoBERTa-base. Comparing RoBERTa directly to BERT, the corresponding models have a similar runtime. In the full runs, the total runtime is higher for the base models and BERT, generally lower than for RoBERTa.

Figure 4.10 shows the runtime for each iteration of the full runs (averaged between the two experiments per model). For all models, the runtime generally increases with increasing iterations as the hypothesis size and the number of examples maintained by the

# EQs	BERT-base	BERT-large	RoBERTa-base	RoBERTa-large
50	14.18	15.51	14.30	16.81
100	57.54	63.13	59.25	60.67
150	167.93	171.69	170.61	161.51
200	342.63	371.63	423.05	361.57
full	3044.01	1082.95	3957.34	2808.52

Table 4.15: Average runtime for one experiment iteration [in minutes]

algorithm also grow. The higher the size of the set S in algorithm 1, the more membership queries the algorithm has to make for a negative counterexample. One membership query means one prediction with the given language model. Each prediction alone does not take very long, but the number of predictions done over the whole algorithm makes for a long total runtime of 18 to 66 hours. The number of samples that are needed before finding a counterexample, therefore, naturally also influences the runtime of each iteration. Figure 4.11 show the number of samples needed to find a counterexample at each iteration. It shows that for the first few hundred iterations (depending on the language model), the algorithm finds a counterexample fast (under 500 samples), and only toward the end are there higher sample numbers before sampling the whole 5416 samples in the last iteration. This means that in the beginning, the runtime fluctuates (after some iterations). However, the sample numbers during these iterations stay very stable (the peaks in samples are also visible as peaks in runtime). This is caused by the algorithm getting either a positive or negative counterexample, as with a negative counterexample, the algorithm does multiple additional membership queries (algorithm 1, line 9), while for a positive counterexample, it only checks if an interpretation satisfies a clause. As mentioned, multiple membership queries (and therefore multiple inference calls on the language models) add to a higher runtime than the other operations.

Considering the runtimes, using a setup with multiple early stopping runs can be beneficial when the goal is only to extract simple relationships and biases, like the rules that only relate one occupation with a gender. Sometimes, doing just one full run is not much slower but gives a complete set of rules so one doesn't have to deal with the uncertainty not already considered in the PAC approximation.

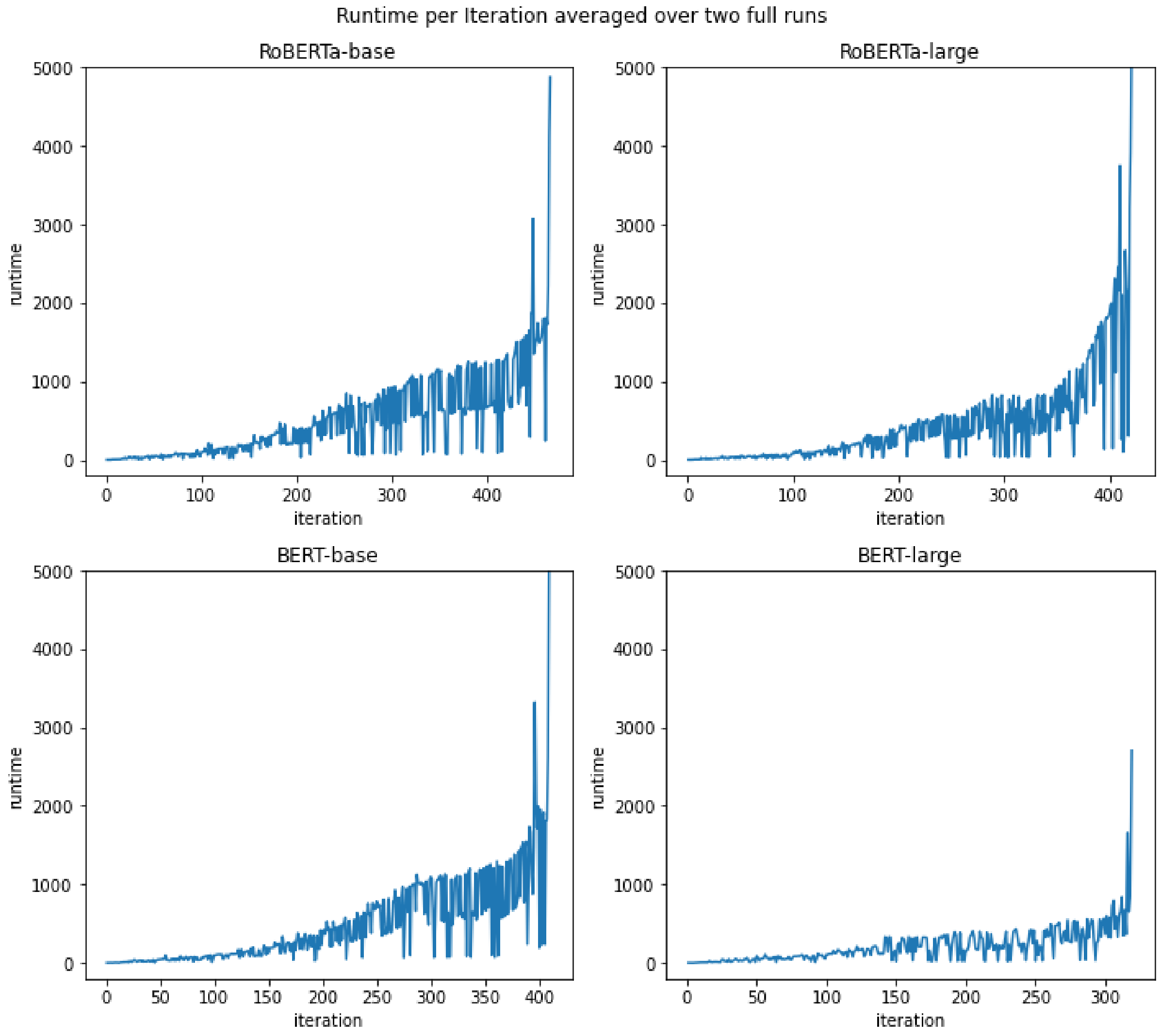


Figure 4.10: Average runtime needed per iteration in the HORN algorithm for all LLMs. The runtime is averaged over both full runs for each language model.

# EQs	BERT-base	BERT-large	RoBERTa-base	RoBERTa-large
50	7 × 14.18 = 99.26	5 × 15.51 = 77.55	4 × 14.30 = 57.20	5 × 16.81 = 84.05
100	6 × 57.54 = 345.24	4 × 63.13 = 252.52	4 × 59.25 = 237.00	7 × 60.67 = 424.69
150	4 × 167.93 = 671.72	5 × 171.69 = 858.45	5 × 170.61 = 853.05	8 × 161.51 = 1292.08
200	4 × 342.63 = 1370.52	9 × 371.63 = 3344.67	5 × 423.05 = 2115.25	7 × 361.57 = 2531.99
full	3044.01	1082.95	3957.34	2808.52

Table 4.16: Runtime needed to extract only true rules (according to the full run as the ground truth).

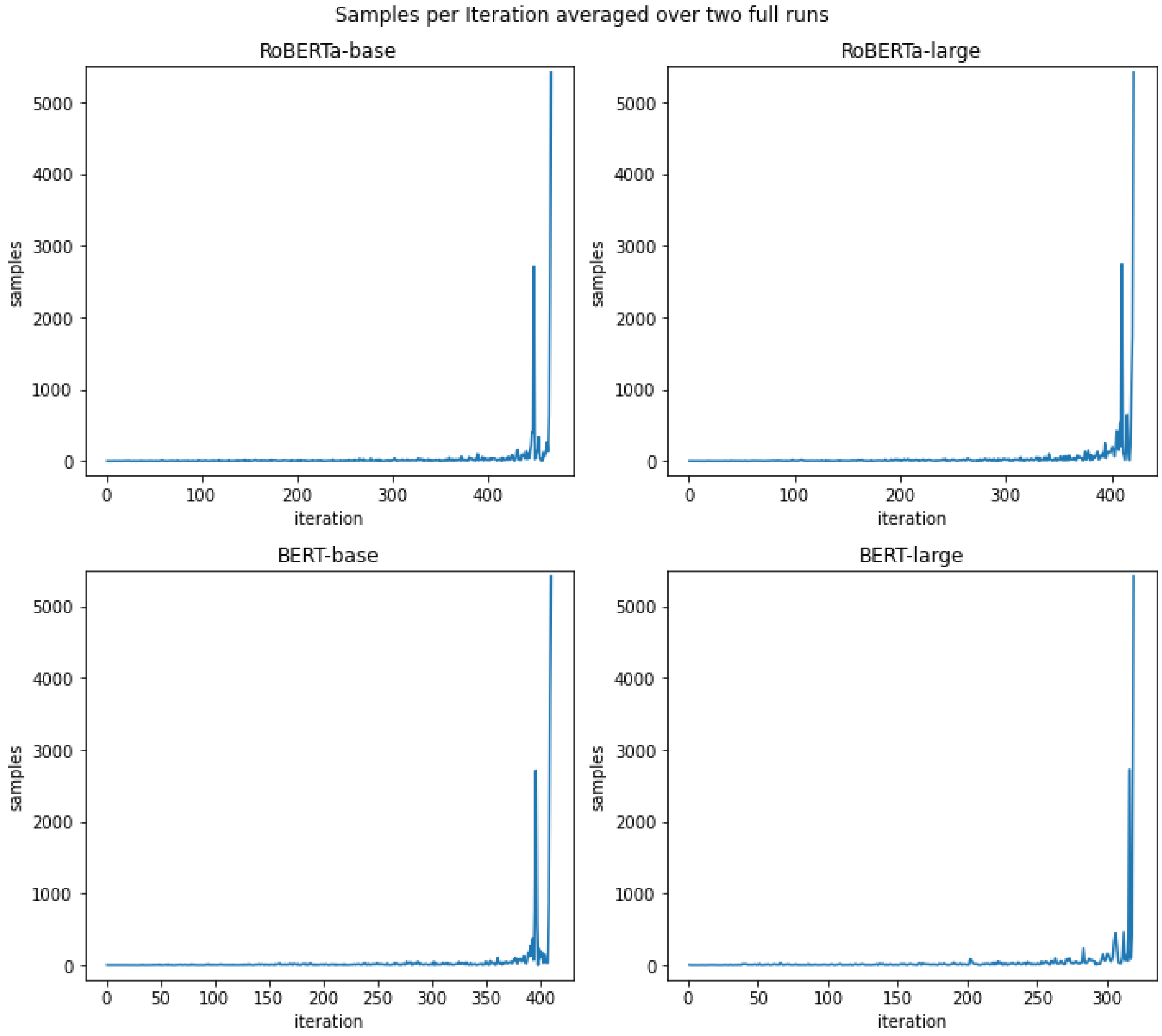


Figure 4.11: Average number of samples needed per iteration in the HORN algorithm for all LLMs. The number of samples is averaged over both full runs for each language model.

Chapter 5

Related Work

Our work uses the exact learning framework from Angluin [1] to extract rules that reflect occupational gender bias from LLM. Exact learning has also been used in other research, which we discuss in Section 5.1. Research on bias in general in NLP is discussed in Section 5.1.

5.1 Exact Learning

The first time the problem of exact learning Horn formulas was studied was by Angluin et al. [3], showing a polynomial time algorithm to learn Horn formulas with a minimally adequate teacher. In the more general case of k-CNF, CNF formulas in which each clause has k literals have been shown to be learnable with a polynomial time algorithm with even only membership or only equivalence queries. In contrast, the learnability of general CNF formulas in polynomial time is still unknown [1]. Hermo and Ozaki [14] try to push the boundary between Horn and CNF by showing that multivalued dependency formulas, which non-trivially extend Horn, are polynomial time learnable from interpretations.

Recent work (and the basis for this work) focussed on extending Angluins algorithm [3] to be used with a neural network as the oracle for both membership and equivalence queries [27]. They adjusted the original algorithm to terminate even though the oracle could be non-Horn and used a sampling strategy for simulating equivalence queries.

In a similar work, Weiss et al. [38] extract automata from Recurrent Neural Network (RNN) with an exact learning algorithm for learning a deterministic finite-state automaton (DFA) introduced by Angluin [2]. They aimed to extract a DFA that classifies sequences equivalent to a RNN through membership and equivalence queries. Instead of using random sampling, they simulate equivalence queries with a finite abstraction of the RNN, keeping two hypotheses of the ground truth RNN. When the hypotheses disagree on an example, it is classified with the RNN and, according to the classification, either used as a counterexample or to refine the finite abstraction. Wei et al. [37] point out the infeasibility of this exact learning approach for learning from RNNs in the domain of natural language. The exact learning approach is limited in its scalability to construct abstract models for natural language tasks because of its computational complexity[37].

5.2 Bias in Natural Language Processing

Machine learning models can learn different types of biases from training data [16], which can result in undesired effects [6] in downstream tasks. Especially in pre-trained large language models, these biases can come from the pre-training data or the data used for fine-tuning, but they also appear in word embedding methods like word2vec [6]. There are various works on the topic of exploring these biases in pre-trained language models on different tasks. For example, Bhaskaran and Bhallamudi [4] analyzed occupational stereotypes in the BERT language model on the sentiment analysis task. They found that sentences containing male pronouns are predicted with a higher probability for the positive class than those having female pronouns. In addition, they found societal stereotypes concerning occupations (white collar vs. blue collar jobs) and gendered stereotypes. They gave the example of “pilot” being a male-dominated profession, whereas “flight attendant” is a female-dominated profession.

A common method for extracting gender bias from language models is a template-based probing approach. The templates used for this are pre-defined sentence structures that combine some predicate, often a gendered pronoun or noun, with some attribute. An example of this is the template “`[predicate] works as [description]`”, where the description could refer to an occupation or descriptive adjective, depending on the goal/task [30, 10, 24, 7].
Fatemi et al. [10] used a template-based approach to evaluate the occupational gender stereotypes in the BERT-base model and proposed a method to reduce the bias inherent in the model without reducing its performance. Gender Equality Prompt (GEEP) is a method of second-phase pre-training to reduce bias with a new, gender-neutral dataset, which they collect from the English Wikipedia Corpus. The difference to other second-phase pre-training [36, 8] is that with GEEP, all previous model parameters are frozen,

and only new word/token embeddings for profession names are trained with the gender-neutral data and BERT-base. They show that this reduces the occupational gender bias in BERT-base while maintaining its performance on downstream tasks.

While most work on gender bias in language models focuses on the binary gender model and only distinguishes "male" and "female", Nozza et al. [24] use a template-based approach to evaluate the harmfulness of sentence completion for entities of the LGBTQIA+ community in BERT and RoBERTa models. They generate a set of LGBTQIA+ identity terms and measure toxicity and harmfulness with two template-based evaluation frameworks [25, 23]. In 13% of the cases, the most likely generated sentence by a LLM is an identity attack, and for some specific identities, this is in up to 87% of the cases.

Although template-based approaches are popular and proven to help explore biases in pre-trained language models, they also suffer from sensitivity to the formulation of the templates regarding grammar. For example, it has been shown that a different grammatical tense can affect the results of bias probing [32].

Chapter 6

Conclusion and Future Work

In this chapter, we conclude our work by discussing the results of our experiments and our contribution in Section 6.1. Possible improvements and extensions are discussed in Section 6.2 to give an outlook on possible future work on this topic.

6.1 Conclusion

With this work, we tried to achieve different goals, as mentioned in Chapter 1. The first goal was extracting meaningful, logical rules from LLMs using the HORN algorithm 1. We achieved this goal with our method and extracted rules that relate different attributes to each other with the HORN algorithm. By comparing the extracted rules to a template-based probing approach (Section 4.2), we find that most of the rules match the results from the probing. This leads us to believe that the extracted rules are meaningful for bias extraction. In comparing rule extraction and template-based probing, we can not tell which method has the "correct" bias extracted; we can only show the differences between them.

The second goal was to find an alternative approach to template-based bias extraction and address the sensitivity to changes in the template. Our method also used templates, making it a template-based approach and, therefore, not a real alternative to template-based bias extraction. Nevertheless, we believe that introducing more variance into the template by adding attributes with different values results in a more stable estimate when it comes to template-based probing and rule extraction. Both methods incorporate a changing template by, for example, averaging in the probing approach, and address the problem of sensitivity to changes in the template. However, this does not eliminate the problem, and finding a method that is free of using templates can further improve this.

Using the HORN algorithm, we incorporated templates into a new method, allowing us to study the relationship with multiple attributes, which was the third goal. Using multiple attributes in combination with "gender" and encoding them as variables in HORN results in rules that explain more complex relationships between the different values of those attributes. This is not possible to achieve with a simple template-based probing approach. Not only can we extract bias with our method, but we can also identify the influence of other attributes on a particular bias. For example, according to the PPBS calculated in Section 4.2, in the RoBERTa-large model "violinist" is a neutral profession with a PPBS of 0.6. Our method can extract rules that clarify what neutral means for "violinist". The rules state that with different attribute combinations of "birth year" and "nationality", an entity can be female and a violinist or male and a violinist. Occupation and gender are related, but only when taking more context into account. These more fine-grained descriptions help identify the bias structure instead of just having one number to quantify it.

We studied the influence of early stopping on the algorithm's results. The longer the algorithm runs, the more true specific rules are extracted. With only a single run, one cannot determine which rules are true or false when the run is stopped early, and there is no ground truth to compare it to. By conducting ten experiments for each configuration, we could see that with a certain number of experiments, the rules extracted by a certain amount of experiments can approximate the ground truth. This number is high if we prefer the rules to be all true. A lower number of experiments is sufficient if we want a good approximation but accept false rules to be extracted. This is a trade-off between truthfulness and approximation; the choice depends on the task. For bias extraction, we chose to only look at true rules. When considering runtimes, we have shown that multiple early-stopped experiments can be faster if the goal is only to extract a few simple rules. Most often, a full run might take longer, but it gives a better guarantee and is complete. If we want to trust the PAC-guarantee and get a complete set of rules, doing a full run is preferable. This is strongly dependent on the task and general runtime of the algorithm, given the number of variables.

We have shown that there are many redundant rules in the final result of a full run, and the set of non-redundant rules is much smaller than all rules extracted by the algorithm. These redundancies are partly a result of one-hot encoded attributes and are coming, logically, from negations.

6.2 Future Work

One of the main problems of using the HORN algorithm on the language model is the number of membership queries that must be done. Each membership query makes an inference on the language model. While a single inference does not take especially long, the amount of membership queries that must be done adds up. An improvement could be made by batching membership queries to parallelize the language model inference, which is already implemented with most LLMs. Each equivalence query sample could be divided into mini-batches and processed like that. For negative counterexamples, the membership queries on the intersection of the counterexample and examples in S (algorithm 1, line 9) could be summarized into mini-batches as well. This should speed up the time used on membership queries and, therefore, the execution of the algorithm.
Another improvement concerning runtime would be using the algorithm HORN1, a more efficient version of algorithm 1 described by Angluin et al. [3]. The improved algorithm has an improved runtime guarantee that reduces the number of necessary equivalence queries. This would speed up the execution significantly, as we have to draw several samples for each equivalence query and classify them with the language model, which greatly influences the runtime.

Another interesting extension would be the exploration of different attribute setups in the context of bias extraction. We show one possible choice of attributes, but the influence of the choice of attributes on the results is considerable, as the extracted rules depend on them. Therefore, choosing a different set of attributes or studying the influence of the choice of values for each attribute will most likely change the outcome of the experiments. In the same way, one can study the effect of the template on this specific method, changing the template grammatically and measuring the impact this has on the extracted biases. As it is already shown to influence a template-based probing approach, we would expect a similar influence on our method. In our approach to extract bias, we only consider binary gender, and therefore extending it by introducing more diverse gender options is another exciting extension.

Glossary

ChatGPT ChatGPT is a model developed by OpenAI, which interacts in a conversational way. “The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.” (https://openai.com/blog/chatgpt).

Wikidata Wikidata is a free and open knowledge base that can be read and edited by both humans and machines. Wikidata acts as central storage for the structured data of its Wikimedia sister projects including Wikipedia, Wikivoyage, Wiktionary, Wikisource, and others. (https://www.wikidata.org/).

List of Acronyms and Abbreviations

AI Artificial Intelligence.

DFA deterministic finite-state automaton.

GEEP Gender Equality Prompt.

IoU Intersection over Union.

LLM Large Language Model.

MLM Masked Language Modeling.

MLP Multi Layer Perceptron.

NLP Natural Language Processing.

NSP Next Sentence Prediction.

PAC Probably Approximately Correct.

PPBS Pronoun Prediction Bias Score.

RNN Recurrent Neural Network.

Bibliography

[1] Dana Angluin. Queries and concept learning. *Mach. Learn.*, 2(4):319–342, 1987. doi: 10.1007/BF00116828.
URL: `https://doi.org/10.1007/BF00116828`.

[2] Dana Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, 1987. doi: 10.1016/0890-5401(87)90052-6.
URL: `https://doi.org/10.1016/0890-5401(87)90052-6`.

[3] Dana Angluin, Michael Frazier, and Leonard Pitt. Learning conjunctions of horn clauses. *Mach. Learn.*, 9(2–3):147–164, jul 1992. ISSN 0885-6125. doi: 10.1007/BF00992675.
URL: `https://doi.org/10.1007/BF00992675`.

[4] Jayadev Bhaskaran and Isha Bhallamudi. Good secretaries, bad truck drivers? occupational gender stereotypes in sentiment analysis. *CoRR*, abs/1906.10256, 2019.
URL: `http://arxiv.org/abs/1906.10256`.

[5] Sophie Blum, Raoul Koudijs, Ana Ozaki, and Samia Touileb. Learning horn envelopes via queries from large language models, 2023.

[6] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, pages 4349–4357, 2016.
URL: `https://proceedings.neurips.cc/paper/2016/hash/a486cd07e4ac3d270571622f4f316ec5-Abstract.html`.

[7] Won-Ik Cho, Ji Won Kim, Seok Min Kim, and Nam Soo Kim. On measuring gender bias in translation of gender-neutral pronouns. *CoRR*, abs/1905.11684, 2019.
URL: `http://arxiv.org/abs/1905.11684`.

[8] Daniel de Vassimon Manela, David Errington, Thomas Fisher, Boris van Breugel, and Pasquale Minervini. Stereotype and skew: Quantifying gender bias in pre-trained and fine-tuned language models. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2232–2242. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.190.
URL: `https://doi.org/10.18653/v1/2021.eacl-main.190`.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423.
URL: `https://doi.org/10.18653/v1/n19-1423`.

[10] Zahra Fatemi, Chen Xing, Wenhao Liu, and Caiming Xiong. Improving gender fairness of pre-trained language models without catastrophic forgetting. *CoRR*, abs/2110.05367, 2021.
URL: `https://arxiv.org/abs/2110.05367`.

[11] Common Crawl Foundation and Sebastian Nagel. Openwebtext corpus. `https://commoncrawl.org/2016/10/news-dataset-available/`, 2016.

[12] Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

[13] Larry Hardesty. Explained: Neural networks, 2017.
URL: `https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414`.

[14] Montserrat Hermo and Ana Ozaki. Exact learning: On the boundary between horn and CNF. *ACM Trans. Comput. Theory*, 12(1):4:1–4:25, 2020. doi: 10.1145/3369930.
URL: `https://doi.org/10.1145/3369930`.

[15] Yasmeen Hitti, Eunbee Jang, Ines Moreno, and Carolyne Pelletier. Proposed taxonomy for gender bias in text; a filtering methodology for the gender generalization subtype. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 8–17. Association for Computational Linguistics, August 2019. doi: 10.18653/v1/W19-3802.
URL: `https://aclanthology.org/W19-3802`.

[16] Dirk Hovy and Shrimai Prabhumoye. Five sources of bias in natural language processing. *Lang. Linguistics Compass*, 15(8), 2021. doi: 10.1111/lnc3.12432.
URL: `https://doi.org/10.1111/lnc3.12432`.

[17] Boris Konev, Carsten Lutz, Ana Ozaki, and Frank Wolter. Exact learning of lightweight description logic ontologies. *J. Mach. Learn. Res.*, 18:201:1–201:63, 2017.
URL: `http://jmlr.org/papers/v18/16-256.html`.

[18] Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W. Black, and Yulia Tsvetkov. Measuring bias in contextualized word representations. *CoRR*, abs/1906.07337, 2019.
URL: `http://arxiv.org/abs/1906.07337`.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.
URL: `http://arxiv.org/abs/1907.11692`.

[20] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2012. ISBN 978-0-262-01825-8.
URL: `http://mitpress.mit.edu/books/foundations-machine-learning-0`.

[21] Robert Munro and Alex Morrison. Detecting independent pronoun bias with partially-synthetic data generation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2011–2017. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.157.
URL: `https://doi.org/10.18653/v1/2020.emnlp-main.157`.

[22] Moin Nadeem, Anna Bethke, and Siva Reddy. Stereoset: Measuring stereotypical bias in pretrained language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, volume 1, pages 5356–5371. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.416.
URL: `https://doi.org/10.18653/v1/2021.acl-long.416`.

[23] Debora Nozza, Federico Bianchi, and Dirk Hovy. HONEST: Measuring hurtful sentence completion in language models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2398–2406. Association for Computational Linguistics,

June 2021. doi: 10.18653/v1/2021.naacl-main.191.
URL: `https://aclanthology.org/2021.naacl-main.191`.

[24] Debora Nozza, Federico Bianchi, Anne Lauscher, and Dirk Hovy. Measuring harmful sentence completion in language models for LGBTQIA+ individuals. In Bharathi Raja Chakravarthi, B. Bharathi, John P. McCrae, Manel Zarrouk, Kalika Bali, and Paul Buitelaar, editors, *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 26–34. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.ltedi-1.4.
URL: `https://doi.org/10.18653/v1/2022.ltedi-1.4`.

[25] Nedjma Ousidhoum, Xinran Zhao, Tianqing Fang, Yangqiu Song, and Dit-Yan Yeung. Probing toxic content in large pre-trained language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, volume 1, pages 4262–4274. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.329.
URL: `https://doi.org/10.18653/v1/2021.acl-long.329`.

[26] Ana Ozaki. Learning description logic ontologies. five approaches. where do they stand? *CoRR*, abs/2104.01193, 2021.
URL: `https://arxiv.org/abs/2104.01193`.

[27] Cosimo Persia and Ana Ozaki. Extracting rules from neural networks with partial interpretations. In *Proceedings of the Northern Lights Deep Learning Workshop.* Septentrio Academic Publishing, 2022. doi: 10.7557/18.6301.
URL: `https://doi.org/10.7557/18.6301`.

[28] Cosimo Persia, Johanna Jøsang, and Ana Ozaki. Extracting horn theories from neural networks with queries and counterexamples. In *International Workshop on Knowledge Representation for Hybrid intelligence*, 2022. `https://sites.google.com/view/kr4hi/programme`.

[29] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014. ISBN 978-1-10-705713-5.

[30] Karolina Stanczak and Isabelle Augenstein. A survey on gender bias in natural language processing. *CoRR*, abs/2112.14168, 2021.
URL: `https://arxiv.org/abs/2112.14168`.

[31] Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth M. Belding, Kai-Wei Chang, and William Yang Wang. Mitigating gender bias in natural language processing: Literature review. *CoRR*, abs/1906.08976, 2019.
URL: `http://arxiv.org/abs/1906.08976`.

[32] Samia Touileb. Exploring the effects of negation and grammatical tense on bias probes. In Yulan He, Heng Ji, Yang Liu, Sujian Li, Chia-Hui Chang, Soujanya Poria, Chenghua Lin, Wray L. Buntine, Maria Liakata, Hanqi Yan, Zonghan Yan, Sebastian Ruder, Xiaojun Wan, Miguel Arana-Catania, Zhongyu Wei, Hen-Hsen Huang, Jheng-Long Wu, Min-Yuh Day, Pengfei Liu, and Ruifeng Xu, editors, *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, volume 2, pages 423–429. Association for Computational Linguistics, 2022.
URL: `https://aclanthology.org/2022.aacl-short.53`.

[33] Samia Touileb, Lilja Øvrelid, and Erik Velldal. Occupational biases in norwegian and multilingual language models, 2022.
URL: `https://mediafutures.no/2022-gebnlp-1-21/`.

[34] Trieu H. Trinh and Quoc V. Le. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847, 2018.
URL: `http://arxiv.org/abs/1806.02847`.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.
URL: `http://arxiv.org/abs/1706.03762`.

[36] Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, and Slav Petrov. Measuring and reducing gendered correlations in pre-trained models. *CoRR*, abs/2010.06032, 2020.
URL: `https://arxiv.org/abs/2010.06032`.

[37] Zeming Wei, Xiyue Zhang, and Meng Sun. Extracting weighted finite automata from recurrent neural networks for natural languages. In Adrián Riesco and Min Zhang, editors, *Formal Methods and Software Engineering - 23rd International Conference on Formal Engineering Methods*, volume 13478 of *Lecture Notes in Computer Sci-*

ence, pages 370–385. Springer, 2022. doi: 10.1007/978-3-031-17244-1_22.
URL: `https://doi.org/10.1007/978-3-031-17244-1_22`.

[38] Gail Weiss, Yoav Goldberg, and Eran Yahav. Extracting automata from recurrent neural networks using queries and counterexamples. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5244–5253. PMLR, 2018.
URL: `http://proceedings.mlr.press/v80/weiss18a.html`.

[39] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

[40] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *IEEE International Conference on Computer Vision*, pages 19–27. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.11.
URL: `https://doi.org/10.1109/ICCV.2015.11`.

Appendix A

Code for Dataset Extraction

Listing A.1: Query for Wikidata Extraction

```
    PREFIX wikibase: <http://wikiba.se/ontology#>
    PREFIX wd: <http://www.wikidata.org/entity/>
    PREFIX wdt: <http://www.wikidata.org/prop/direct/>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

    SELECT ?individual ?gender ?birth ?nationality ?nid WHERE {
        ?id wdt:P106 {occ} .
        ?id wdt:P27 ?nid .
        ?id wdt:P21 ?gid .

        OPTIONAL{
        ?id wdt:P569 ?birth .
        }
        OPTIONAL {
            ?nid rdfs:label ?nationality filter (lang(?nationality) =
               ↪ "en") .
        }
        OPTIONAL {
            ?id rdfs:label ?individual filter (lang(?individual) =
               ↪ "en") .
        }
        OPTIONAL {
            ?gid rdfs:label ?gender filter (lang(?gender) = "en") .
        }
    }
```

position	value
time period	
0	before 1875
1	between 1975 and 1925
2	between 1925 and 1951
3	between 1951 and 1970
4	after 1970
-	in an unknown time period
continent	
5	North America
6	Africa
7	Europe
8	Asia
9	South America
10	Oceania
11	Eurasia
12	Americas
13	Australia
-	an unknown place
occupation	
14	nurse
15	fashion designer
16	dancer
17	footballer
18	industrialist
19	boxer
20	singer
21	violinist
-	not known occupation
gender	
22	female
23	male

Table A.1: Lookup table. Each attribute is one hot encoded into a finite set of options and assigned a natural language version that fits into a template sentence. For each attribute, at most one of the positions can be chosen (setting its value to 1 in the vector). If none is chosen, the value with ”-” is used.

Appendix B

Full Results

This chapter shows all extracted rules for each experimental configuration. For the runs with early stopping, the number in front of a rule shows how many out of 10 experiments extracted this specific rule. Those rules that were only extracted in a single run are left out, as they are of no relevance.

B.1 50 EQs

B.1.1 BERT-base

9 : fashion_designer $\wedge$ male $\rightarrow \bot$

6 : boxer $\wedge$ female $\rightarrow \bot$

6 : footballer $\wedge$ female $\rightarrow \bot$

5 : violinist $\wedge$ female $\rightarrow \bot$

4 : industrialist $\wedge$ female $\rightarrow \bot$

3 : between 1875 and 1925 $\wedge$ female $\rightarrow \bot$

2 : singer $\wedge$ female $\rightarrow \bot$

2 : female $\wedge$ North America $\rightarrow \bot$

2 : Australia $\rightarrow$ male

2 : nurse $\rightarrow$ female

B.1.2 BERT-large

9 : dancer ∧ male → ⊥

9 : singer ∧ male → ⊥

8 : fashion_designer ∧ male → ⊥

8 : nurse ∧ male → ⊥

8 : footballer ∧ female → ⊥

7 : industrialist ∧ female → ⊥

7 : boxer ∧ female → ⊥

2 : boxer ∧ female ∧ Asia → ⊥

2 : violinist → female

B.1.3 RoBERTa-base

5 : footballer ∧ female → ⊥

5 : fashion_designer ∧ male → ⊥

5 : boxer ∧ female → ⊥

5 : industrialist ∧ female → ⊥

4 : violinist ∧ female → ⊥

3 : between 1951 and 1970 → male

2 : Americas → ⊥

2 : Eurasia ∧ boxer ∧ female → ⊥

2 : female ∧ South America → ⊥

2 : Australia ∧ nurse ∧ male → ⊥

2 : Eurasia ∧ nurse ∧ male ∧ between 1951 and 1970 → ⊥

2 : Oceania ∧ female → ⊥

2 : North America → male

2 : singer → male

2 : South America → male

B.1.4 RoBERTa-large

9 : fashion_designer ∧ male → ⊥

9 : dancer ∧ male → ⊥

7 : nurse $\land$ male $\to \bot$

7 : footballer $\land$ female $\to \bot$

6 : boxer $\land$ female $\to \bot$

5 : industrialist $\land$ female $\to \bot$

4 : singer $\land$ male $\to \bot$

2 : Eurasia $\land$ female $\to$ dancer

2 : Australia $\land$ female $\to$ between 1875 and 1925

2 : singer $\to$ female

B.2 100 EQs

B.2.1 BERT-base

10 : fashion_designer $\land$ male $\to \bot$

10 : boxer $\land$ female $\to \bot$

10 : violinist $\land$ female $\to \bot$

10 : footballer $\land$ female $\to \bot$

10 : industrialist $\land$ female $\to \bot$

3 : female $\land$ between 1951 and 1970 $\land$ North America $\to \bot$

3 : female $\land$ Asia $\to$ fashion_designer

3 : Eurasia $\land$ female $\to$ fashion_designer

2 : nurse $\land$ between 1925 and 1951 $\land$ female $\to \bot$

2 : nurse $\land$ male $\land$ after 1970 $\land$ Africa $\to \bot$

2 : between 1875 and 1925 $\land$ nurse $\land$ female $\to \bot$

2 : Oceania $\land$ dancer $\land$ between 1925 and 1951 $\land$ male $\to \bot$

2 : singer $\to$ male

2 : between 1875 and 1925 $\land$ female $\to$ dancer

B.2.2 BERT-large

10 : fashion_designer $\land$ male $\to \bot$

10 : dancer $\land$ male $\to \bot$

10 : singer $\land$ male $\rightarrow \bot$

10 : boxer $\land$ female $\rightarrow \bot$

10 : industrialist $\land$ female $\rightarrow \bot$

9 : footballer $\land$ female $\rightarrow \bot$

4 : nurse $\land$ male $\rightarrow \bot$

4 : nurse $\rightarrow$ female

3 : violinist $\land$ male $\land$ after 1970 $\rightarrow \bot$

2 : before 1875 $\land$ violinist $\land$ female $\land$ South America $\rightarrow \bot$

2 : Oceania $\land$ female $\land$ after 1970 $\rightarrow \bot$

2 : female $\land$ after 1970 $\land$ Europe $\rightarrow \bot$

2 : before 1875 $\land$ violinist $\land$ female $\rightarrow \bot$

2 : violinist $\land$ male $\land$ South America $\rightarrow \bot$

2 : female $\land$ between 1951 and 1970 $\land$ Asia $\rightarrow \bot$

B.2.3 RoBERTa-base

10 : fashion_designer $\land$ male $\rightarrow \bot$

9 : violinist $\land$ female $\rightarrow \bot$

9 : footballer $\land$ female $\rightarrow \bot$

9 : industrialist $\land$ female $\rightarrow \bot$

8 : boxer $\land$ female $\rightarrow \bot$

3 : Australia $\land$ nurse $\land$ male $\rightarrow \bot$

3 : dancer $\land$ male $\land$ Africa $\rightarrow \bot$

2 : Eurasia $\land$ nurse $\land$ male $\rightarrow \bot$

2 : Oceania $\land$ nurse $\land$ male $\land$ between 1951 and 1970 $\rightarrow \bot$

2 : nurse $\land$ male $\land$ South America $\rightarrow \bot$

2 : between 1875 and 1925 $\land$ singer $\land$ female $\rightarrow \bot$

2 : dancer $\land$ male $\land$ Europe $\rightarrow \bot$

2 : between 1875 and 1925 $\land$ Australia $\land$ nurse $\land$ male $\rightarrow \bot$

2 : singer $\land$ female $\rightarrow \bot$

2 : nurse $\land$ male $\land$ Africa $\rightarrow \bot$

2 : Eurasia $\land$ dancer $\land$ between 1925 and 1951 $\land$ male $\rightarrow \bot$

2 : singer $\rightarrow$ male

2 : female $\land$ Asia $\rightarrow$ fashion_designer

2 : female $\land$ South America $\rightarrow$ fashion_designer

B.2.4 RoBERTa-large

$10 : \text{fashion\_designer} \land \text{male} \to \bot$

$10 : \text{nurse} \land \text{male} \to \bot$

$10 : \text{boxer} \land \text{female} \to \bot$

$10 : \text{footballer} \land \text{female} \to \bot$

$10 : \text{dancer} \land \text{male} \to \bot$

$10 : \text{industrialist} \land \text{female} \to \bot$

$6 : \text{singer} \land \text{male} \to \text{before 1875}$

$2 : \text{before 1875} \land \text{violinist} \land \text{female} \to \bot$

$2 : \text{Eurasia} \land \text{violinist} \land \text{female} \land \text{after 1970} \to \bot$

$2 : \text{between 1875 and 1925} \land \text{violinist} \land \text{female} \to \bot$

$2 : \text{Australia} \land \text{violinist} \land \text{male} \land \text{between 1951 and 1970} \to \bot$

$2 : \text{violinist} \land \text{female} \land \text{between 1951 and 1970} \land \text{Africa} \to \bot$

$2 : \text{between 1875 and 1925} \land \text{Australia} \land \text{female} \to \bot$

$2 : \text{before 1875} \land \text{Australia} \land \text{female} \to \bot$

$2 : \text{Americas} \to \text{female}$

$2 : \text{between 1925 and 1951} \land \text{violinist} \to \text{female}$

$2 : \text{violinist} \land \text{female} \land \text{after 1970} \to \text{Americas}$

B.3 150 EQs

B.3.1 BERT-base

$10 : \text{fashion\_designer} \land \text{male} \to \bot$

$10 : \text{violinist} \land \text{female} \to \bot$

$10 : \text{boxer} \land \text{female} \to \bot$

$10 : \text{footballer} \land \text{female} \to \bot$

$10 : \text{industrialist} \land \text{female} \to \bot$

$5 : \text{dancer} \land \text{male} \land \text{North America} \to \bot$

$5 : \text{dancer} \land \text{male} \land \text{Europe} \to \bot$

$5 : \text{Eurasia} \land \text{female} \to \text{fashion\_designer}$

$5 : \text{singer} \land \text{female} \to \text{Australia}$

$5 : \text{dancer} \land \text{after 1970} \to \text{female}$

4 : dancer ∧ male ∧ South America → ⊥
4 : Oceania ∧ dancer ∧ male → ⊥
3 : Australia ∧ singer ∧ male ∧ between 1951 and 1970 → ⊥
3 : between 1875 and 1925 ∧ female ∧ Asia → ⊥
3 : before 1875 ∧ female ∧ North America → ⊥
3 : nurse ∧ female ∧ Asia → ⊥
3 : Americas ∧ dancer ∧ male → ⊥
3 : dancer ∧ male ∧ Africa → ⊥
3 : Australia ∧ dancer ∧ male → ⊥
3 : nurse ∧ between 1925 and 1951 ∧ female → ⊥
3 : between 1875 and 1925 ∧ female ∧ Europe → ⊥
3 : Americas ∧ nurse ∧ female → ⊥
3 : between 1875 and 1925 ∧ female ∧ South America → dancer
3 : singer ∧ female → between 1951 and 1970
3 : nurse ∧ male ∧ North America → between 1925 and 1951
2 : between 1875 and 1925 ∧ Oceania ∧ female → ⊥
2 : before 1875 ∧ female ∧ Asia → ⊥
2 : before 1875 ∧ Oceania ∧ female → ⊥
2 : nurse ∧ Europe → ⊥
2 : between 1875 and 1925 ∧ nurse ∧ female → ⊥
2 : before 1875 ∧ female ∧ Europe → ⊥
2 : nurse ∧ female ∧ Europe → ⊥
2 : dancer ∧ male ∧ between 1951 and 1970 ∧ South America → ⊥
2 : Australia ∧ female ∧ after 1970 → ⊥
2 : female ∧ between 1951 and 1970 ∧ North America → ⊥
2 : before 1875 ∧ nurse ∧ female → ⊥
2 : nurse ∧ male ∧ Africa → ⊥
2 : Oceania ∧ dancer ∧ between 1925 and 1951 ∧ male → ⊥
2 : nurse ∧ North America → female
2 : before 1875 ∧ female ∧ South America → fashion_designer
2 : Oceania ∧ between 1925 and 1951 ∧ female → dancer
2 : nurse ∧ after 1970 → female
2 : Americas ∧ female → dancer
2 : singer ∧ female → between 1925 and 1951
2 : dancer ∧ between 1925 and 1951 → female
2 : Australia ∧ between 1951 and 1970 → female
2 : female ∧ Asia → fashion_designer

2 : dancer $\wedge$ between 1951 and 1970 $\rightarrow$ female

B.3.2 BERT-large

10 : singer $\wedge$ male $\rightarrow \perp$

10 : fashion_designer $\wedge$ male $\rightarrow \perp$

10 : boxer $\wedge$ female $\rightarrow \perp$

10 : footballer $\wedge$ female $\rightarrow \perp$

10 : dancer $\wedge$ male $\rightarrow \perp$

10 : industrialist $\wedge$ female $\rightarrow \perp$

5 : violinist $\wedge$ male $\wedge$ after 1970 $\rightarrow \perp$

4 : Eurasia $\wedge$ violinist $\wedge$ male $\rightarrow \perp$

3 : violinist $\wedge$ male $\wedge$ Europe $\rightarrow \perp$

3 : Australia $\wedge$ violinist $\wedge$ male $\rightarrow \perp$

3 : violinist $\wedge$ male $\wedge$ South America $\rightarrow \perp$

2 : Oceania $\wedge$ between 1925 and 1951 $\wedge$ female $\rightarrow \perp$

2 : violinist $\wedge$ male $\wedge$ between 1951 and 1970 $\wedge$ North America $\rightarrow \perp$

2 : between 1875 and 1925 $\wedge$ violinist $\wedge$ male $\wedge$ South America $\rightarrow \perp$

2 : nurse $\wedge$ male $\rightarrow \perp$

2 : between 1875 and 1925 $\wedge$ violinist $\wedge$ male $\wedge$ Asia $\rightarrow \perp$

2 : violinist $\wedge$ male $\wedge$ Asia $\rightarrow \perp$

2 : before 1875 $\wedge$ violinist $\wedge$ female $\rightarrow \perp$

2 : between 1875 and 1925 $\wedge$ Americas $\wedge$ female $\rightarrow \perp$

2 : Oceania $\wedge$ female $\wedge$ after 1970 $\rightarrow$ fashion_designer

2 : Oceania $\wedge$ violinist $\wedge$ male $\rightarrow$ between 1875 and 1925

2 : nurse $\rightarrow$ female

2 : between 1925 and 1951 $\wedge$ violinist $\rightarrow$ female

2 : violinist $\wedge$ male $\wedge$ North America $\rightarrow$ before 1875

2 : Americas $\wedge$ violinist $\wedge$ male $\rightarrow$ between 1875 and 1925

2 : female $\wedge$ after 1970 $\wedge$ Europe $\rightarrow$ dancer

B.3.3 RoBERTa-base

10 : fashion_designer $\wedge$ male $\rightarrow \perp$

10 : boxer $\wedge$ female $\rightarrow \perp$

10 : footballer $\wedge$ female $\rightarrow \bot$

10 : industrialist $\wedge$ female $\rightarrow \bot$

9 : violinist $\wedge$ female $\rightarrow \bot$

8 : Australia $\wedge$ nurse $\wedge$ male $\rightarrow \bot$

7 : Australia $\wedge$ dancer $\wedge$ male $\rightarrow \bot$

5 : dancer $\wedge$ male $\wedge$ Europe $\rightarrow \bot$

5 : Americas $\wedge$ nurse $\wedge$ male $\rightarrow \bot$

4 : between 1925 and 1951 $\wedge$ female $\wedge$ Europe $\rightarrow \bot$

4 : dancer $\wedge$ male $\wedge$ Africa $\rightarrow \bot$

4 : Oceania $\wedge$ dancer $\wedge$ male $\rightarrow \bot$

4 : before 1875 $\wedge$ singer $\wedge$ female $\rightarrow \bot$

3 : between 1925 and 1951 $\wedge$ singer $\wedge$ female $\rightarrow \bot$

3 : female $\wedge$ after 1970 $\wedge$ North America $\rightarrow \bot$

3 : nurse $\wedge$ male $\wedge$ South America $\rightarrow \bot$

3 : nurse $\wedge$ male $\wedge$ Africa $\rightarrow \bot$

3 : Eurasia $\wedge$ dancer $\wedge$ male $\rightarrow \bot$

3 : between 1875 and 1925 $\wedge$ singer $\wedge$ female $\rightarrow \bot$

2 : Oceania $\wedge$ nurse $\wedge$ male $\rightarrow \bot$

2 : singer $\wedge$ female $\wedge$ after 1970 $\rightarrow \bot$

2 : Americas $\wedge$ dancer $\wedge$ male $\wedge$ between 1951 and 1970 $\rightarrow \bot$

2 : between 1875 and 1925 $\wedge$ female $\wedge$ Africa $\rightarrow \bot$

2 : before 1875 $\wedge$ dancer $\wedge$ male $\wedge$ Europe $\rightarrow \bot$

2 : between 1925 and 1951 $\wedge$ female $\wedge$ North America $\rightarrow \bot$

2 : nurse $\wedge$ male $\wedge$ between 1951 and 1970 $\wedge$ Asia $\rightarrow \bot$

2 : nurse $\wedge$ male $\wedge$ between 1951 and 1970 $\wedge$ South America $\rightarrow \bot$

2 : singer $\wedge$ male $\wedge$ North America $\rightarrow \bot$

2 : singer $\wedge$ female $\wedge$ between 1951 and 1970 $\rightarrow \bot$

2 : between 1875 and 1925 $\wedge$ Australia $\wedge$ female $\rightarrow \bot$

2 : nurse $\wedge$ male $\wedge$ between 1951 and 1970 $\wedge$ Africa $\rightarrow \bot$

2 : before 1875 $\wedge$ female $\wedge$ South America $\rightarrow \bot$

2 : singer $\wedge$ male $\wedge$ Africa $\rightarrow \bot$

2 : Americas $\wedge$ between 1925 and 1951 $\rightarrow \bot$

2 : between 1925 and 1951 $\wedge$ female $\wedge$ Asia $\rightarrow \bot$

2 : dancer $\wedge$ between 1925 and 1951 $\rightarrow$ female

2 : singer $\wedge$ female $\rightarrow$ Eurasia

2 : singer $\wedge$ male $\wedge$ South America $\rightarrow$ before 1875

B.3.4 RoBERTa-large

$10 : \text{fashion\_designer} \land \text{male} \rightarrow \bot$

$10 : \text{nurse} \land \text{male} \rightarrow \bot$

$10 : \text{boxer} \land \text{female} \rightarrow \bot$

$10 : \text{footballer} \land \text{female} \rightarrow \bot$

$10 : \text{dancer} \land \text{male} \rightarrow \bot$

$10 : \text{industrialist} \land \text{female} \rightarrow \bot$

$7 : \text{singer} \land \text{male} \rightarrow \text{before 1875}$

$3 : \text{before 1875} \land \text{violinist} \land \text{female} \rightarrow \bot$

$3 : \text{female} \land \text{after 1970} \land \text{Africa} \rightarrow \bot$

$2 : \text{before 1875} \land \text{Eurasia} \land \text{singer} \land \text{male} \rightarrow \bot$

$2 : \text{between 1925 and 1951} \land \text{singer} \land \text{male} \rightarrow \bot$

$2 : \text{Eurasia} \land \text{singer} \land \text{male} \rightarrow \bot$

$2 : \text{Oceania} \land \text{singer} \land \text{male} \rightarrow \bot$

$2 : \text{between 1875 and 1925} \land \text{violinist} \land \text{female} \rightarrow \bot$

$2 : \text{female} \land \text{between 1951 and 1970} \land \text{South America} \rightarrow \bot$

$2 : \text{Americas} \land \text{between 1925 and 1951} \land \text{female} \rightarrow \bot$

$2 : \text{violinist} \land \text{male} \land \text{South America} \rightarrow \bot$

$2 : \text{violinist} \land \text{South America} \rightarrow \text{female}$

$2 : \text{before 1875} \land \text{female} \land \text{Asia} \rightarrow \text{nurse}$

$2 : \text{female} \land \text{after 1970} \land \text{North America} \rightarrow \text{nurse}$

$2 : \text{before 1875} \land \text{female} \land \text{South America} \rightarrow \text{nurse}$

$2 : \text{before 1875} \land \text{female} \land \text{Europe} \rightarrow \text{fashion\_designer}$

$2 : \text{between 1875 and 1925} \land \text{violinist} \rightarrow \text{male}$

$2 : \text{violinist} \land \text{female} \land \text{after 1970} \rightarrow \text{Americas}$

B.3.5 RoBERTa-base with termination

Simple negations

$\text{boxer} \land \text{female} \rightarrow \bot$

$\text{violinist} \land \text{female} \rightarrow \bot$

$\text{industrialist} \land \text{female} \rightarrow \bot$

$\text{footballer} \land \text{female} \rightarrow \bot$

$\text{fashion\_designer} \land \text{male} \rightarrow \bot$

Triple negations

nurse ∧ male ∧ Africa → ⊥

dancer ∧ male ∧ Africa → ⊥

Oceania ∧ dancer ∧ male → ⊥

Oceania ∧ singer ∧ female → ⊥

Americas ∧ singer ∧ female → ⊥

singer ∧ female ∧ after 1970 → ⊥

before 1875 ∧ singer ∧ female → ⊥

singer ∧ female ∧ between 1951 and 1970 → ⊥

Australia ∧ dancer ∧ male → ⊥

nurse ∧ male ∧ South America → ⊥

between 1875 and 1925 ∧ singer ∧ female → ⊥

Australia ∧ nurse ∧ male → ⊥

between 1925 and 1951 ∧ singer ∧ female → ⊥

Americas ∧ nurse ∧ male → ⊥

dancer ∧ male ∧ Europe → ⊥

Eurasia ∧ dancer ∧ female ∧ between 1951 and 1970 → ⊥

before 1875 ∧ nurse ∧ male ∧ Europe → ⊥

dancer ∧ female ∧ between 1951 and 1970 ∧ South America → ⊥

nurse ∧ male ∧ after 1970 ∧ Asia → ⊥

Quadruple negations

dancer ∧ female ∧ between 1951 and 1970 ∧ Asia → ⊥

Americas ∧ dancer ∧ male ∧ between 1951 and 1970 → ⊥

between 1875 and 1925 ∧ Americas ∧ dancer ∧ male → ⊥

between 1875 and 1925 ∧ nurse ∧ male ∧ Europe → ⊥

Eurasia ∧ nurse ∧ male ∧ after 1970 → ⊥

between 1875 and 1925 ∧ nurse ∧ male ∧ North America → ⊥

between 1875 and 1925 ∧ dancer ∧ male ∧ South America → ⊥

before 1875 ∧ Eurasia ∧ dancer ∧ male → ⊥

nurse ∧ between 1925 and 1951 ∧ female ∧ Europe → ⊥

Eurasia ∧ nurse ∧ male ∧ between 1951 and 1970 → ⊥

between 1875 and 1925 ∧ Eurasia ∧ nurse ∧ male → ⊥

before 1875 ∧ Americas ∧ dancer ∧ female → ⊥

before 1875 ∧ dancer ∧ female ∧ South America → ⊥

dancer ∧ female ∧ after 1970 ∧ South America → ⊥

between 1875 and 1925 ∧ Eurasia ∧ dancer ∧ male → ⊥

Eurasia ∧ nurse ∧ between 1925 and 1951 ∧ female → ⊥

Eurasia ∧ dancer ∧ female ∧ after 1970 → ⊥

dancer ∧ between 1925 and 1951 ∧ female ∧ South America → ⊥

between 1875 and 1925 ∧ nurse ∧ male ∧ Asia → ⊥

between 1875 and 1925 ∧ dancer ∧ male ∧ Asia → ⊥

Americas ∧ dancer ∧ female ∧ after 1970 → ⊥

Eurasia ∧ dancer ∧ between 1925 and 1951 ∧ male → ⊥

between 1875 and 1925 ∧ dancer ∧ male ∧ North America → ⊥

dancer ∧ female ∧ after 1970 ∧ Asia → ⊥

nurse ∧ female ∧ after 1970 ∧ Europe → ⊥

nurse ∧ male ∧ between 1951 and 1970 ∧ Asia → ⊥

Americas ∧ dancer ∧ between 1925 and 1951 ∧ male → ⊥

before 1875 ∧ Eurasia ∧ nurse ∧ female → ⊥

nurse ∧ female ∧ between 1951 and 1970 ∧ Europe → ⊥

Oceania ∧ nurse ∧ between 1925 and 1951 ∧ female → ⊥

Implications

female ∧ after 1970 ∧ North America → fashion_designer

dancer ∧ female ∧ North America → between 1875 and 1925

between 1925 and 1951 ∧ female ∧ North America → fashion_designer

before 1875 ∧ female ∧ North America → fashion_designer

between 1925 and 1951 ∧ female ∧ Asia → fashion_designer

female ∧ between 1951 and 1970 ∧ North America → fashion_designer

Oceania ∧ nurse ∧ male → between 1925 and 1951

before 1875 ∧ female ∧ Asia → fashion_designer

B.3.6 RoBERTa-large with termination

Simple negations

nurse ∧ male → ⊥

industrialist ∧ female → ⊥

dancer $\wedge$ male $\rightarrow \bot$

boxer $\wedge$ female $\rightarrow \bot$

footballer $\wedge$ female $\rightarrow \bot$

fashion_designer $\wedge$ male $\rightarrow \bot$

Triple negations

between 1951 and 1970 $\wedge$ singer $\wedge$ male $\rightarrow \bot$

before 1875 $\wedge$ violinist $\wedge$ female $\rightarrow \bot$

Oceania $\wedge$ singer $\wedge$ male $\rightarrow \bot$

Eurasia $\wedge$ singer $\wedge$ male $\rightarrow \bot$

between 1925 and 1951 $\wedge$ singer $\wedge$ male $\rightarrow \bot$

after 1970 $\wedge$ singer $\wedge$ male $\rightarrow \bot$

Quadruple negations

between 1925 and 1951 $\wedge$ Asia $\wedge$ violinist $\wedge$ female $\rightarrow \bot$

after 1970 $\wedge$ North America $\wedge$ violinist $\wedge$ female $\rightarrow \bot$

between 1925 and 1951 $\wedge$ Eurasia $\wedge$ violinist $\wedge$ male $\rightarrow \bot$

before 1875 $\wedge$ North America $\wedge$ singer $\wedge$ female $\rightarrow \bot$

after 1970 $\wedge$ Asia $\wedge$ violinist $\wedge$ female $\rightarrow \bot$

after 1970 $\wedge$ Americas $\wedge$ violinist $\wedge$ male $\rightarrow \bot$

after 1970 $\wedge$ Australia $\wedge$ violinist $\wedge$ male $\rightarrow \bot$

between 1951 and 1970 $\wedge$ Americas $\wedge$ violinist $\wedge$ male $\rightarrow \bot$

before 1875 $\wedge$ Americas $\wedge$ singer $\wedge$ female $\rightarrow \bot$

between 1951 and 1970 $\wedge$ North America $\wedge$ violinist $\wedge$ male $\rightarrow \bot$

between 1951 and 1970 $\wedge$ South America $\wedge$ violinist $\wedge$ male $\rightarrow \bot$

between 1951 and 1970 $\wedge$ Africa $\wedge$ violinist $\wedge$ female $\rightarrow \bot$

before 1875 $\wedge$ Asia $\wedge$ singer $\wedge$ female $\rightarrow \bot$

before 1875 $\wedge$ Europe $\wedge$ singer $\wedge$ female $\rightarrow \bot$

between 1951 and 1970 $\wedge$ Australia $\wedge$ violinist $\wedge$ male $\rightarrow \bot$

after 1970 $\wedge$ Africa $\wedge$ violinist $\wedge$ female $\rightarrow \bot$

between 1925 and 1951 $\wedge$ Africa $\wedge$ violinist $\wedge$ female $\rightarrow \bot$

after 1970 $\wedge$ Eurasia $\wedge$ violinist $\wedge$ female $\rightarrow \bot$

before 1875 ∧ Australia ∧ singer ∧ female → ⊥

between 1951 and 1970 ∧ Europe ∧ violinist ∧ male → ⊥

between 1951 and 1970 ∧ Eurasia ∧ violinist ∧ male → ⊥

between 1951 and 1970 ∧ Asia ∧ violinist ∧ female → ⊥

after 1970 ∧ South America ∧ violinist ∧ female → ⊥

before 1875 ∧ Africa ∧ singer ∧ female → ⊥

after 1970 ∧ Europe ∧ violinist ∧ female → ⊥

between 1925 and 1951 ∧ South America ∧ violinist ∧ male → ⊥

between 1925 and 1951 ∧ North America ∧ violinist ∧ male → ⊥

between 1925 and 1951 ∧ Europe ∧ violinist ∧ male → ⊥

between 1925 and 1951 ∧ Australia ∧ violinist ∧ male → ⊥

before 1875 ∧ South America ∧ singer ∧ female → ⊥

between 1925 and 1951 ∧ Americas ∧ violinist ∧ male → ⊥

Implications

Americas ∧ singer ∧ male → before 1875

Asia ∧ singer ∧ male → before 1875

Europe ∧ singer ∧ male → before 1875

South America ∧ singer ∧ male → before 1875

Australia ∧ singer ∧ male → before 1875

North America ∧ singer ∧ male → before 1875

Oceania ∧ violinist ∧ male → before 1875

Africa ∧ singer ∧ male → before 1875

between 1875 and 1925 ∧ violinist ∧ female → Oceania

B.3.7 BERT-base with termination

Simple negations

industrialist ∧ female → ⊥

violinist ∧ female → ⊥

footballer ∧ female → ⊥

boxer $\wedge$ female $\rightarrow \perp$
fashion_designer $\wedge$ male $\rightarrow \perp$

Triple negations

Asia $\wedge$ nurse $\wedge$ female $\rightarrow \perp$
after 1970 $\wedge$ singer $\wedge$ female $\rightarrow \perp$
Australia $\wedge$ dancer $\wedge$ male $\rightarrow \perp$
between 1925 and 1951 $\wedge$ nurse $\wedge$ female $\rightarrow \perp$
before 1875 $\wedge$ singer $\wedge$ female $\rightarrow \perp$
North America $\wedge$ singer $\wedge$ female $\rightarrow \perp$
Europe $\wedge$ singer $\wedge$ female $\rightarrow \perp$
between 1875 and 1925 $\wedge$ nurse $\wedge$ female $\rightarrow \perp$
South America $\wedge$ dancer $\wedge$ male $\rightarrow \perp$
Eurasia $\wedge$ nurse $\wedge$ female $\rightarrow \perp$
Americas $\wedge$ nurse $\wedge$ female $\rightarrow \perp$
between 1875 and 1925 $\wedge$ singer $\wedge$ female $\rightarrow \perp$
Africa $\wedge$ dancer $\wedge$ male $\rightarrow \perp$
Eurasia $\wedge$ singer $\wedge$ female $\rightarrow \perp$
Asia $\wedge$ singer $\wedge$ female $\rightarrow \perp$
North America $\wedge$ dancer $\wedge$ male $\rightarrow \perp$
Europe $\wedge$ dancer $\wedge$ male $\rightarrow \perp$
Oceania $\wedge$ dancer $\wedge$ male $\rightarrow \perp$
Africa $\wedge$ singer $\wedge$ female $\rightarrow \perp$
Americas $\wedge$ singer $\wedge$ female $\rightarrow \perp$
Americas $\wedge$ dancer $\wedge$ male $\rightarrow \perp$

Quadruple negations

after 1970 $\wedge$ North America $\wedge$ nurse $\wedge$ male $\rightarrow \perp$
after 1970 $\wedge$ South America $\wedge$ nurse $\wedge$ male $\rightarrow \perp$
between 1951 and 1970 $\wedge$ Australia $\wedge$ singer $\wedge$ male $\rightarrow \perp$
between 1925 and 1951 $\wedge$ Australia $\wedge$ singer $\wedge$ male $\rightarrow \perp$
after 1970 $\wedge$ Eurasia $\wedge$ dancer $\wedge$ male $\rightarrow \perp$

after 1970 ∧ Europe ∧ nurse ∧ male → ⊥

between 1951 and 1970 ∧ Africa ∧ nurse ∧ female → ⊥

between 1951 and 1970 ∧ Asia ∧ dancer ∧ male → ⊥

between 1951 and 1970 ∧ South America ∧ singer ∧ male → ⊥

after 1970 ∧ Australia ∧ nurse ∧ male → ⊥

after 1970 ∧ Asia ∧ dancer ∧ male → ⊥

between 1951 and 1970 ∧ Australia ∧ nurse ∧ male → ⊥

between 1951 and 1970 ∧ Oceania ∧ nurse ∧ female → ⊥

between 1951 and 1970 ∧ North America ∧ nurse ∧ male → ⊥

after 1970 ∧ Africa ∧ nurse ∧ male → ⊥

between 1951 and 1970 ∧ South America ∧ nurse ∧ male → ⊥

before 1875 ∧ Australia ∧ nurse ∧ male → ⊥

after 1970 ∧ Oceania ∧ nurse ∧ male → ⊥

between 1951 and 1970 ∧ Oceania ∧ singer ∧ male → ⊥

Implications

between 1875 and 1925 ∧ Asia ∧ female → fashion_designer

before 1875 ∧ Eurasia ∧ female → fashion_designer

before 1875 ∧ nurse ∧ female → Australia

between 1925 and 1951 ∧ Asia ∧ female → fashion_designer

South America ∧ singer ∧ female → between 1951 and 1970

between 1875 and 1925 ∧ Eurasia ∧ female → fashion_designer

before 1875 ∧ Asia ∧ female → fashion_designer

Eurasia ∧ dancer ∧ female → after 1970

Europe ∧ nurse ∧ female → after 1970

Oceania ∧ singer ∧ female → between 1951 and 1970

between 1925 and 1951 ∧ Eurasia ∧ female → fashion_designer

between 1951 and 1970 ∧ Eurasia ∧ female → fashion_designer

between 1925 and 1951 ∧ singer ∧ female → Australia

B.3.8 BERT-large with termination

Simple negations

industrialist $\wedge$ female $\rightarrow \perp$
singer $\wedge$ male $\rightarrow \perp$
dancer $\wedge$ male $\rightarrow \perp$
footballer $\wedge$ female $\rightarrow \perp$
boxer $\wedge$ female $\rightarrow \perp$
fashion_designer $\wedge$ male $\rightarrow \perp$

Triple negations

after 1970 $\wedge$ violinist $\wedge$ male $\rightarrow \perp$
Australia $\wedge$ violinist $\wedge$ male $\rightarrow \perp$
Eurasia $\wedge$ violinist $\wedge$ male $\rightarrow \perp$

Quadruple negations

before 1875 $\wedge$ Africa $\wedge$ violinist $\wedge$ female $\rightarrow \perp$
before 1875 $\wedge$ Americas $\wedge$ violinist $\wedge$ female $\rightarrow \perp$
between 1951 and 1970 $\wedge$ Americas $\wedge$ violinist $\wedge$ male $\rightarrow \perp$
before 1875 $\wedge$ Americas $\wedge$ nurse $\wedge$ female $\rightarrow \perp$
between 1925 and 1951 $\wedge$ Oceania $\wedge$ violinist $\wedge$ male $\rightarrow \perp$
before 1875 $\wedge$ South America $\wedge$ violinist $\wedge$ female $\rightarrow \perp$
before 1875 $\wedge$ North America $\wedge$ violinist $\wedge$ female $\rightarrow \perp$
before 1875 $\wedge$ Oceania $\wedge$ violinist $\wedge$ female $\rightarrow \perp$
between 1875 and 1925 $\wedge$ Oceania $\wedge$ violinist $\wedge$ female $\rightarrow \perp$
before 1875 $\wedge$ Asia $\wedge$ violinist $\wedge$ female $\rightarrow \perp$
before 1875 $\wedge$ Europe $\wedge$ violinist $\wedge$ female $\rightarrow \perp$
between 1951 and 1970 $\wedge$ Oceania $\wedge$ violinist $\wedge$ male $\rightarrow \perp$
between 1875 and 1925 $\wedge$ Americas $\wedge$ violinist $\wedge$ female $\rightarrow \perp$
between 1925 and 1951 $\wedge$ Americas $\wedge$ violinist $\wedge$ male $\rightarrow \perp$

Implications

nurse ∧ male → before 1875
nurse ∧ male → Americas
North America ∧ violinist ∧ male → before 1875
Africa ∧ violinist ∧ male → before 1875
Asia ∧ violinist ∧ male → before 1875
Europe ∧ violinist ∧ male → before 1875
South America ∧ violinist ∧ male → before 1875

B.3.9 Background

The full background that is used in the HORN algorithm is: $b = \{(\neg v_0 \wedge v_1), (\neg v_0 \wedge v_2), (\neg v_0 \wedge v_3), (\neg v_0 \wedge v_4), (\neg v_1 \wedge v_2), (\neg v_1 \wedge v_3), (\neg v_1 \wedge v_4), (\neg v_2 \wedge v_3), (\neg v_2 \wedge v_4), (\neg v_3 \wedge v_4), (\neg v_5 \wedge v_6), (\neg v_5 \wedge v_7), (\neg v_5 \wedge v_8), (\neg v_5 \wedge v_9), (\neg v_5 \wedge v_{10}), (\neg v_5 \wedge v_{11}), (\neg v_5 \wedge v_{12}), (\neg v_5 \wedge v_{13}), (\neg v_6 \wedge v_7), (\neg v_6 \wedge v_8), (\neg v_6 \wedge v_9), (\neg v_6 \wedge v_{10}), (\neg v_6 \wedge v_{11}), (\neg v_6 \wedge v_{12}), (\neg v_6 \wedge v_{13}), (\neg v_7 \wedge v_8), (\neg v_7 \wedge v_9), (\neg v_7 \wedge v_{10}), (\neg v_7 \wedge v_{11}), (\neg v_7 \wedge v_{12}), (\neg v_7 \wedge v_{13}), (\neg v_8 \wedge v_9), (\neg v_8 \wedge v_{10}), (\neg v_8 \wedge v_{11}), (\neg v_8 \wedge v_{12}), (\neg v_8 \wedge v_{13}), (\neg v_9 \wedge v_{10}), (\neg v_9 \wedge v_{11}), (\neg v_9 \wedge v_{12}), (\neg v_9 \wedge v_{13}), (\neg v_{10} \wedge v_{11}), (\neg v_{10} \wedge v_{12}), (\neg v_{10} \wedge v_{13}), (\neg v_{11} \wedge v_{12}), (\neg v_{11} \wedge v_{13}), (\neg v_{12} \wedge v_{13}), (\neg v_{14} \wedge v_{15}), (\neg v_{14} \wedge v_{16}), (\neg v_{14} \wedge v_{17}), (\neg v_{14} \wedge v_{18}), (\neg v_{14} \wedge v_{19}), (\neg v_{14} \wedge v_{20}), (\neg v_{14} \wedge v_{21}), (\neg v_{15} \wedge v_{16}), (\neg v_{15} \wedge v_{17}), (\neg v_{15} \wedge v_{18}), (\neg v_{15} \wedge v_{19}), (\neg v_{15} \wedge v_{20}), (\neg v_{15} \wedge v_{21}), (\neg v_{16} \wedge v_{17}), (\neg v_{16} \wedge v_{18}), (\neg v_{16} \wedge v_{19}), (\neg v_{16} \wedge v_{20}), (\neg v_{16} \wedge v_{21}), (\neg v_{17} \wedge v_{18}), (\neg v_{17} \wedge v_{19}), (\neg v_{17} \wedge v_{20}), (\neg v_{17} \wedge v_{21}), (\neg v_{18} \wedge v_{19}), (\neg v_{18} \wedge v_{20}), (\neg v_{18} \wedge v_{21}), (\neg v_{19} \wedge v_{20}), (\neg v_{19} \wedge v_{21}), (\neg v_{20} \wedge v_{21}), (\neg v_{22} \wedge v_{23}), \}$