CausaLM: Causal Model Explanation Through Counterfactual Language Models

Amir Feder
Faculty of Industrial Engineering and
Management
Technion - Israel Institute of Technology
`feder@campus.technion.ac.il`

Nadav Oved
Faculty of Industrial Engineering and
Management
Technion - Israel Institute of Technology
`nadavo@campus.technion.ac.il`

Uri Shalit
Faculty of Industrial Engineering and
Management
Technion - Israel Institute of Technology
`urishalit@technion.ac.il`

Roi Reichart
Faculty of Industrial Engineering and
Management
Technion - Israel Institute of Technology
`roiri@technion.ac.il`

Understanding predictions made by deep neural networks is notoriously difficult, but also crucial to their dissemination. As all machine learning–based methods, they are as good as their training data, and can also capture unwanted biases. While there are tools that can help understand whether such biases exist, they do not distinguish between correlation and causation, and might be ill-suited for text-based models and for reasoning about high-level language concepts. A key problem of estimating the causal effect of a concept of interest on a given model is that this estimation requires the generation of counterfactual examples, which is challenging with existing generation technology. To bridge that gap, we propose CausaLM, a framework for producing causal model explanations using counterfactual language representation models. Our approach is based on fine-tuning of deep contextualized embedding models with auxiliary adversarial tasks derived from the causal graph of the problem. Concretely, we show that by carefully choosing

auxiliary adversarial pre-training tasks, language representation models such as BERT can effectively learn a counterfactual representation for a given concept of interest, and be used to estimate its true causal effect on model performance. A byproduct of our method is a language representation model that is unaffected by the tested concept, which can be useful in mitigating unwanted bias ingrained in the data.[1]

1. Introduction

The rise of deep learning models (deep neural networks [DNNs]) has produced better prediction models for a plethora of fields, particularly for those that rely on unstructured data, such as computer vision and natural language processing (NLP) (Peters et al. 2018; Devlin et al. 2019). In recent years, variants of these models have disseminated into many industrial applications, varying from image recognition to machine translation (Szegedy et al. 2016; Wu et al. 2016; Aharoni, Johnson, and Firat 2019). In NLP, they were also shown to produce better language models, and are being widely used both for language representation and for classification in nearly every subfield (Tshitoyan et al. 2019; Gao, Galley, and Li 2018; Lee et al. 2020; Feder et al. 2020).

Although DNNs are very successful, this success has come at the expense of model explainability and interpretability. Understanding predictions made by these models is difficult, as their layered structure coupled with non-linear activations do not allow us to reason about the effect of each input feature on the model's output. In the case of text-based models, this problem is amplified. Basic textual features are usually composed of n-grams of adjacent words, but these features alone are limited in their ability to encode meaningful information conveyed in the text. While abstract linguistic concepts, such as topic or sentiment, do express meaningful information, they are usually not explicitly encoded in the model's input.[2] Such concepts might push the model toward making specific predictions, without being directly modeled and therefore interpreted.

Effective concept-based explanations are crucial for the dissemination of DNN-based NLP prediction models in many domains, particularly in scientific applications to fields such as healthcare and the social sciences that often rely on model interpretability for deployment. Failing to account for the actual effect of concepts on text classifiers can potentially lead to biased, unfair, misinterpreted and incorrect predictions. As models are dependent on the data they are trained on, a bias existing in the data could potentially result in a model that under-performs when this bias no longer holds in the test set.

Recently, there have been many attempts to build tools that allow for DNN explanations and interpretations (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017), which have developed into a sub-field often referred to as Blackbox-NLP (Linzen et al. 2019). These tools can be roughly divided into **local explanations**, where the effect of a feature on the classifier's prediction for a specific example is tested, and **global explanations**, which measure the general effect of a given feature on a classifier. A prominent research direction in DNN explainability involves utilizing network artifacts such as attention mechanisms, which are argued to provide a powerful representation tool (Vaswani et al. 2017) to explain how certain decisions are made (but see Jain and Wallace [2019] and Wiegreffe and Pinter [2019] for a discussion of the actual explanation

---

1 Our code and data are available at: `https://amirfeder.github.io/CausaLM/`.
2 By concept, we refer to a higher level, often aggregated unit, compared to lower level, atomic input features such as words. Some examples of linguistic concepts are sentiment, linguistic register, formality, or topics discussed in the text. For a more detailed discussion of concepts, see Kim et al. (2018) and Goyal et al. (2019a).

power of this approach). Alternatively, there have been attempts to estimate simpler, more easily interpretable models around test examples or their hidden representations (Ribeiro, Singh, and Guestrin 2016; Kim et al. 2018).

Unfortunately, existing model explanation tools often rely on local perturbations of the input and compute shallow correlations, which can result in misleading, and sometimes wrong, interpretations. This problem arises, for example, in cases where two concepts that can potentially explain the predictions of the model are strongly correlated with each other. An explanation model that only considers correlations cannot indicate if the first concept, the second concept, or both concepts are in fact the cause of the prediction.

In order to illustrate the importance of causal and concept-based explanations, consider the example presented in Figure 1, which will be our running example throughout the paper. Suppose we have a binary classifier, trained to predict the sentiment conveyed in news articles. Say we hypothesize that the choice of adjectives is driving the classification decision, something that has been discussed previously in computational linguistics (Pang, Lee, and Vaithyanathan 2002). However, if the text is written about a controversial figure, it could be that the presence of its name, or the topics that it induces, are what is driving the classification decision, and not the use of adjectives. The text in the figure is an example of such a case, where both adjectives and the mentioning of politicians seem to affect one another, and could be driving the classifier's prediction. Estimating the effect of Donald Trump's presence in the text on the predictions of the model is also hard, as this presence clearly affects the choice of adjectives, the other political figures mentioned in the text, and probably many additional textual choices.

Notice that an explanation model that only considers correlations might show that the mention of a political figure is strongly correlated with the prediction, leading to worries about the classifier having political bias. However, such a model cannot indicate whether the political figure is in fact the cause of the prediction, or whether it is actually the type of adjectives used that is the true cause of the classifier output, suggesting that the classifier is not politically biased. This highlights the importance of causal concept-based explanations.

A natural causal explanation methodology would be to generate counterfactual examples and compare the model prediction for each example with its prediction for the counterfactual. That is, one needs a controlled setting where it is possible to compute

---

President **Trump** did his best imitation of **Ronald Reagan** at the State of the Union address, falling just short of declaring it Morning in America, the **iconic** imagery and message of a campaign ad that **Reagan** rode to re-election in 1984. **Trump** talked of Americans as pioneers and explorers; he lavished praise on members of the military, several of whom he recognized from the podium; he **optimistically** declared that the best is yet to come. It was a **masterful** performance – but behind the **sunny** smile was the same old **Trump**: **petty**, **angry**, **vindictive** and **deceptive**. He refused to shake the hand of House Speaker **Nancy Pelosi**, a snub she returned in kind by ostentatiously ripping up her copy of the President's speech at the conclusion of the address, in full view of the cameras.

Figure 1
An example of a political commentary piece published at `https://edition.cnn.com`. Highlighted in **blue** and **red** are names of political figures from the US Democratic and Republican parties, respectively. Adjectives are highlighted in **green**.

the difference between an actual observed text, and what the text would have been had a specific concept (e.g., a political figure) not existed in it. Indeed, there have been some attempts to construct counterfactuals for generating local explanations. Specifically, Goyal et al. (2019b) proposed changing the pixels of an image to those of another image classified differently by the classifier, in order to compute the effect of those pixels. However, as this method takes advantage of the spatial structure of images, it is hard to replicate their process with texts. Vig et al. (2020) offered to use mediation analysis to study which parts of the DNN are pushing toward specific decisions by querying the language model. Although their work further highlights the usefulness of counterfactual examples for answering causal questions in model interpretation, they create counterfactual examples manually, by changing specific tokens in the original example. Unfortunately, this does not support automatic estimation of the causal effect that high-level concepts have on model performance.

Going back to our example (Figure 1), training a generative model to condition on a concept, such as the choice of adjectives, and produce counterfactual examples that only differ by this concept, is still intractable in most cases involving natural language (see Section 3.3 for a more detailed discussion). While there are instances where this seems to be improving (Semeniuta, Severyn, and Barth 2017; Fedus, Goodfellow, and Dai 2018), generating a version of the example where a different political figure is being discussed while keeping other concepts unaffected is very hard (Radford et al. 2018, 2019). Alternatively, our key technical observation is that instead of generating a counterfactual text we can more easily generate a counterfactual textual representation, based on adversarial training.

It is important to note that it is not even necessarily clear what the concepts are that should be considered as the "generating concepts" of the text.[3] In our example we only consider adjectives and the political figure, but there are other concepts that generate the text, such as the topics being discussed, the sentiment being conveyed, and others. The number of concepts that would be needed and their coverage of the generated text are also issues that we touch on below. The choice of such *control concepts* depends on our model of the world, as in the *causal graph* example presented in Figure 2 (Section 3). In our experiments we control for such concepts, as our model of the world dictates both *treated concepts* and *control concepts*.[4]

In order to implement the above principles, in this article we propose a model explanation methodology that manipulates the representation of the text rather than the text itself. By creating a text encoder that is not affected by a specific concept of interest, we can compute the *counterfactual representation*. Our explanation method, which we name **Causal Model Explanation through Counterfactual Language Models** (**CausaLM**), receives the classifier's training data and a concept of interest as input, and outputs the causal effect of the concept on the classifier in the test set. It does that by pre-training an additional instance of the language representation model used by the classifier, with an adversarial component designed to "forget" the concept of choice, while keeping the other "important" (control) concepts represented. Following

---

3 Our example also sheds more light on the nature of a concept. For example, if we train a figure classifier on the text after deleting the name of the political figure, it will probably still be able to classify the text correctly according to the figure it discusses. Hence, a concept is a more abstract entity, referring to an entire "semantic space/neighbourhood."

4 While failing to estimate the causal effect of a concept on a sentiment classifier is harmful, it pales in comparison to the potential harm of wrongly interpreting clinical prediction models. In Appendix A we give an example from the medical domain, where the importance of causal explanations has already been established (Zech et al. 2018).
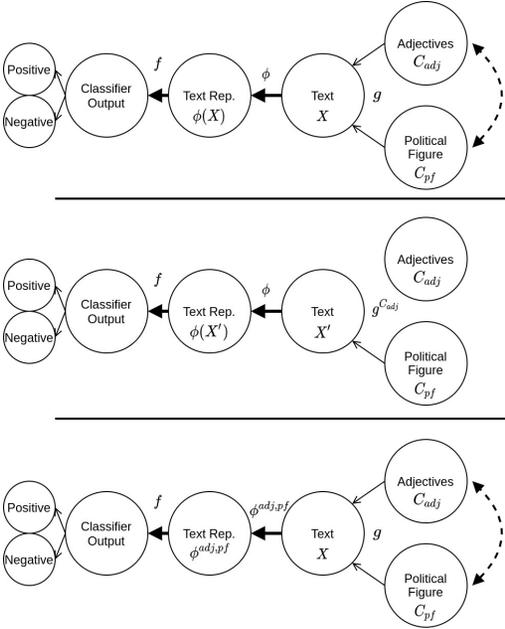
Figure 2
Three causal graphs relating the concepts of *Adjectives* and *Political Figure*, texts, their
representations, and classifier output. The top graph describes the original data-generating
process g. The middle graph describes the case of directly manipulating the text. In this case,
using the generative process $g^{C_{adj}}$ allows us to generate a text X' that is the same as X but does
not contain *Adjectives*. The bottom graph describes our approach, where we manipulate the
representation mechanism and not the actual text. The dashed edge indicates a possible hidden
confounder of the two concepts.

the additional training step, the representation produced by this counterfactual model
can be used to measure the concept's effect on the classifier's prediction for each test
example, by comparing the classifier performance with the two representations.

We start by diving into the link between causality and interpretability (Section 2).
We then discuss how to estimate causal effects from observational data using language
representations (Section 3): Defining the causal estimator (Section 3.1 and 3.2), dis-
cussing the challenges of producing counterfactual examples (Section 3.3), and, with
those options laid out, moving to describe how we can approximate counterfactual ex-
amples through manipulation of the language representation (Section 3.3). Importantly,
our concept-based causal effect estimator does not require counterfactual examples—it
works solely with observational data.

To test our method, we introduce in Section 4 four novel data sets, three of which
include counterfactual examples for a given concept. Building on those data sets, we
present in Section 5 four cases where a BERT-based representation model can be mod-
ified to ignore concepts, such as *Adjectives*, *Topics*, *Gender*, and *Race*, in various settings
involving sentiment and mood state classification (Section 5). To prevent a loss of
information on correlated concepts, we further modify the representation to remember
such concepts while forgetting the concept whose causal effect is estimated. Although
in most of our experiments we test our methods in controlled settings, where the true
causal concept effect can be measured, our approach can be used in the real world,

where such ground truth does not exist. Indeed, in our analysis we provide researchers with tools to estimate the quality of the causal estimator without access to gold standard causal information.

Using our newly created data sets, we estimate the causal effect of concepts on a BERT-based classifier utilizing our intervention method and compare it with the ground truth causal effect, computed with manually created counterfactual examples (Section 6). To equip researchers with tools for using our framework in the real world, we provide an analysis of what happens to the language representation following the intervention, and discuss how to choose adversarial training tasks effectively (Section 6.2). As our approach relies only on interventions done prior to the supervised task training stage, it is not dependent on BERT's specific implementation and can be applied whenever a pre-trained language representation model is used. We also show that our counterfactual models can be used to mitigate unwanted bias in cases where its effect on the classifier can negatively affect outcomes. Finally, we discuss the strengths and limitations of our approach, and propose future research directions at the intersection of causal inference and NLP model interpretation (Section 7).

We hope that this research will spur more interest in the usefulness of causal inference for DNN interpretation and for creating more robust models, within the NLP community and beyond.

2. Previous Work

Previous work on the intersection of DNN interpretations and causal inference, specifically in relation to NLP, is rare. While there is a vast and rich literature on each of those topics alone, the gap between interpretability, causality, and NLP is only now starting to close (Vig et al. 2020). To ground our work in those pillars, we survey here previous work in each. Specifically, we discuss how to use causal inference in NLP (Keith, Jensen, and O'Connor 2020), and describe the current state of research on model interpretations and debiasing in NLP. Finally, we discuss our contribution in light of the relevant work.

2.1 Causal Inference and NLP

There is a rich body of work on causality and on causal inference, as it has been at the core of scientific reasoning since the writings of Plato and Aristotle (Woodward 2005). The questions that drive most researchers interested in understanding human behavior are causal in nature, not associational (Pearl 2009a). They require some knowledge or explicit assumptions regarding the data-generating process, such as the world model we describe in the causal graph presented in Figure 2. Generally speaking, causal questions cannot be answered using the data alone, or through the distributions that generate it (Pearl 2009a).

Even though causal inference is widely used in the life and social sciences, it has not had the same impact on machine learning and NLP in particular (Angrist and Pischke 2008; Dorie et al. 2019; Gentzel, Garant, and Jensen 2019). This can mostly be attributed to the fact that using existing frameworks from causal inference in NLP is challenging (Keith, Jensen, and O'Connor 2020). The high-dimensional nature of language does not easily fit into the current methods, specifically as the treatment whose effect is being tested is often binary (D'Amour et al. 2017; Athey et al. 2017). Recently, this seems to be changing, with substantial work being done on the intersection of causal inference and NLP (Tan, Lee, and Pang 2014; Fong and Grimmer 2016; Egami et al. 2018; Wood-Doughty, Shpitser, and Dredze 2018; Veitch, Sridhar, and Blei 2019).

Specifically, researchers have been looking into methods of measuring other confounders via text (Pennebaker, Francis, and Booth 2001; Saha et al. 2019), or using text as confounders (Johansson, Shalit, and Sontag 2016; De Choudhury et al. 2016; Roberts, Stewart, and Nielsen 2018). In this strand of work, a confounder is being retrieved from the text and used to answer a causal question, or the text itself is used as a potential confounder, with its dimensionality reduced. Another promising direction is causally driven representation learning, where the representation of the text is designed specifically for the purposes of causal inference. This is usually done when the treatment affects the text, and the model architecture is manipulated to incorporate the treatment assignment (Roberts et al. 2014; Roberts, Stewart, and Nielsen 2018). Recently, Veitch, Sridhar, and Blei (2019) added to BERT's fine-tuning stage an objective that estimates propensity scores and conditional outcomes for the treatment and control variables, and used a model to estimate the treatment effect. As opposed to our work, they are interested in creating low-dimensional text embeddings that can be used as variables for answering causal questions, not in interpreting what affects an existing model.

While previous work from the causal inference literature used text to answer causal questions, to the best of our knowledge we are the first (except for Vig et al. [2020]) that are using this framework for causal model explanation. Specifically, we build in this research on a specific subset of causal inference literature, counterfactual analysis (Pearl 2009b), asking causal questions aimed at inferring what would have been the predictions of a given neural model had conditions been different. We present this counterfactual analysis as a method for interpreting DNN-based models, to understand what affects their decisions. By intervening on the textual representation, we provide a framework for answering causal questions regarding the effect of low- and high-level concepts on text classifiers without having to generate counterfactual examples.

Vig et al. (2020) also suggest using ideas from causality for DNN explanations, but focus on understanding how information flows through different model components, while we are interested in understanding the effect of textual concepts on classification decisions. They are dependant on manually constructed queries, such as comparing the language model's probability for a male pronoun to that of a female, for a given masked word. As their method can only be performed by manually creating counterfactual examples such as this query, it is exposed to all the problems involving counterfactual text generation (see Section 3.3). Also, they do not compare model predictions on examples and their counterfactuals, and only measure the difference between the two queries, neither of which are the original text. In contrast, we propose a generalized method for providing a causal explanation for any textual concept, and present data sets where any causal estimator can be tested and compared to a ground truth. We also generate a language representation that approximates counterfactuals for a given concept of interest on each example, thus allowing for a causal model explanation without having to manually create examples.

2.2 Model Interpretations and Debiasing in NLP

Model interpretability is the degree to which a human can consistently predict the model's outcome (Kim, Koyejo, and Khanna 2016a; Doshi-Velez and Kim 2017; Lipton 2018). The more easily interpretable a machine learning model is, the easier it is for someone to comprehend why certain decisions or predictions have been made. An explanation usually relates the feature values of an instance to its model prediction in a humanly understandable way, usually referred to as a **local explanation**. Alternatively,

it can be composed of an estimation of the global effect of a certain feature on the model's predictions.

There is an abundance of recent work on model explanations and interpretations, especially following the rise of DNNs in the past few years (Lundberg and Lee 2017; Ribeiro, Singh, and Guestrin 2016). Vig et al. (2020) divide interpretations in NLP into structural and behavioral methods. Structural methods try to identify the information encoded in the model's internal structure by using its representations to classify textual properties (Adi et al. 2017; Hupkes, Veldhoen, and Zuidema 2018; Conneau et al. 2018). For example, Adi et al. (2017) find that representations based on averaged word vectors encode information regarding sentence length. Behavioral methods evaluate models on specific examples that reflect an hypothesis regarding linguistic phenomena they capture (Sennrich 2017; Isabelle, Cherry, and Foster 2017; Naik et al. 2019). Sennrich (2017), for example, discover that neural machine translation systems perform transliteration better than models with byte-pair encoding segmentation, but are worse in terms of capturing morphosyntactic agreement.

Both structural and behavioral methods generally do not offer ways to directly measure the effect of the structure of the text or the linguistic concepts it manifests on model outcomes. They often rely on token level analysis, and do not account for counterfactuals. Still, there has been very little research in NLP on incorporating tools from causal analysis into model explanations (Vig et al. 2020) (see above), something that lies at the heart of our work. Moreover, there's been, to the best of our knowledge, no work on measuring the effect of concepts on models' predictions in NLP (see Kim et al. [2018] and Goyal et al. [2019a] for a discussion in the context of computer vision).

Closely related to model interpretability, debiasing is a rising sub-field that deals with creating models and language representations that are unaffected by unwanted biases that might exist in the data (Kiritchenko and Mohammad 2018; Elazar and Goldberg 2018; Gonen and Goldberg 2019; Ravfogel et al. 2020). DNNs are as good as the training data they are fed, and can often learn associations that are in direct proportion to the distribution observed during training (Caliskan, Bryson, and Narayanan 2017). While debiasing is still an ongoing effort, there are methods for removing some of the bias encoded in models and language representations (Gonen and Goldberg 2019). Model debiasing is done through manipulation of the training data (Kaushik, Hovy, and Lipton 2020), by altering the training process (Huang et al. 2020), or by changing the model (Gehrmann et al. 2020).

Recently, Ravfogel et al. (2020) offered a method for removing bias from neural representations, by iteratively training linear classifiers and projecting the representations on their null-spaces. Their method does not provide causal model explanation, but instead reveals correlations between certain textual features and the predictions of the model. Particularly, it does not account for control concepts as we do, which makes it prone to overestimating the causal effect of the treatment concept (see Section 6 where we empirically demonstrate this phenomenon).

Our work is the first to provide data sets where bias can be computed directly by comparing predictions on examples and their counterfactuals. Comparatively, existing measures model bias using observational, rather than interventional, measures (Rudinger, May, and Durme 2017; De-Arteaga et al. 2019; Davidson, Bhattacharya, and Weber 2019; Swinger et al. 2019; Ravfogel et al. 2020). To compare methods for causal model explanations, the research community would require data sets, like those presented here, where we can intervene on specific textual features and test whether candidate methods can estimate their effect. In the future we plan to develop richer, more complex data sets that would allow for even more realistic counterfactual comparisons.

3. Causal Model Explanation

While usually in scientific endeavors causal inference is the main focus, we rely here on a different aspect of causality—causal model explanation. That is, we attempt to estimate the causal effect of a given variable (also known as the **treatment**) on the model's predictions, and present such effects to explain the observed behavior of the model. Here we formalize model explanation as a causal inference problem, and propose a method to do that through language representations.

We start by providing a short introduction to causal inference and its basic terminology, focusing on its application to NLP. To ground our discussion within NLP, we follow the *Adjectives* example from Section 1 and present in Figure 2 a *causal diagram*, a graph that could describe the data-generating process of that example. Building on this graph, we discuss its connection to Pearl's *structural causal model* and the *do*-operator (Pearl 2009a). Typically, causal models are built for understanding real-world outcomes, while model interpretability efforts deal with the case where the classification decision is the outcome, and the intervention is on a feature present in the model's input. As we are the first, to the best of our knowledge, to propose a comprehensive causal framework for model interpretations in NLP, we link between the existing literature in both fields.

3.1 Causal Inference and Language Representations

Confounding Factors and the do-*operator.* Continuing with the example from Section 1 (presented in Figure 1), imagine we observe a text X and have trained a model to classify each example as either positive or negative, corresponding to the conveyed sentiment. We also have information regarding the *Political Figure* discussed in the text, and tags for the parts of speech in it. Given a set of concepts, which we hypothesize might affect the model's classification decision, we denote the set of binary variables $C = \{C_j \in \{0,1\} \mid j \in \{0,1,\ldots,k\}\}$, where each variable corresponds to the existence of a predefined concept in the text, i.e., if $C_j = 1$ then the j-th concept appears in the text. We further assume a pre-trained language representation model ϕ (such as BERT), and wish to assert how our trained classifier f is affected by the concepts in C, where f is a classifier that takes $\phi(X)$ as input and outputs a class $l \in L$. As we are interested in the effect on the probability assigned to each class by the classifier f, we measure the class probability of our output for an example X, and denote it for a class $l \in L$ as z_l. When computing differences on all L classes, we use $\vec{z}(f(\phi(X)))$, the vector of all z_l probabilities.

Computing the effect of a concept C_j on $\vec{z}(f(\phi(X)))$ seems like an easy problem. We can simply feed to our model examples with and without the chosen concepts, and compute the difference between the average $\vec{z}(\cdot)$ in both cases. For example, if our concept of interest is positive *Adjectives*, we can feed the model with examples that include positive *Adjectives* and examples that do not. Then, we can compare the difference between the averaged $\vec{z}(\cdot)$ in both sets and conclude that this difference is the effect of positive *Adjectives*.

Now, imagine the case where the use of positive and negative *Adjectives* is associated with the *Political Figure* that is being discussed in the texts given to the model. An obvious example is a case where a political commentator with liberal-leaning opinions is writing about a conservative politician, or vice versa. In that case, it would be reasonable to assume that the *Political Figure* being discussed would affect the text through other concepts besides its identity. The author can then choose to express her opinion through *Adjectives* or in other ways, and these might be correlated. In such cases, comparing

examples with and without positive *Adjectives* would result in an inaccurate measurement of their effect on the classification decisions of the model.[5]

The problem with our correlated concepts is that of **confounding**. It is illustrated in the top graph of Figure 2 using the example of *Political Figure* and *Adjectives*. In causal inference, a **confounder** is a variable that affects other variables and the predicted label. In our case, the *Political Figure* (C_{pf}) being discussed in the texts is a confounder of the *Adjectives* concept, as it directly affects both C_{adj} and X. As can be seen in this figure, we can think of texts as originating from a list of concepts. While we plot only two, *Adjectives* and *Political Figure*, there could be many concepts generating a text. We denote the potential confoundedness of the concepts by dashed arrows, to represent that one could affect the other or that they have a common cause.

Alternatively, if it was the case that a change of the *Political Figure* would not affect the usage of *Adjectives* in the text, we could have said that C_{adj} and C_{pf} are not confounded. This is the case where we could intervene on C_{adj}, such as by having the author write a text without using positive *Adjectives*, without inducing a text that contains a different *Political Figure*. In causal terms, this is the case where:

$$\vec{z}(f(\phi(X)|do(C_{adj}))) = \vec{z}(f(\phi(X)|C_{adj})) \tag{1}$$

where $do(C_{adj})$ stands for an external intervention that compels the change of C_{adj}. In contrast, the class probability distribution $\vec{z}(f(\phi(X)|C_{adj}))$ represents the distribution resulting from a passive observation of C_{adj}, and rarely coincides with $\vec{z}(f(\phi(X)|do(C_{adj})))$. Indeed, the passive observation setup relates to the probability that the sentiment is positive given that positive adjectives are used. In contrast, the external intervention setup relates to the probability that the sentiment is positive after all the information about positive adjectives has been removed from a text that originally (pre-intervention) conveyed positive sentiment.

Counterfactual Text Representations. The act of manipulating the text to change the *Political Figure* being discussed or the *Adjectives* used in the text is derived from the notion of *counterfactuals*. In the *Adjectives* example (presented in Figure 1), a counterfactual text is such an instance where we intervene on one concept only, holding everything else equal. It is the equivalent of imagining what could have been the text, had it been written about a different *Political Figure*, or about the same *Political Figure* but with different *Adjectives*.

In the case of *Adjectives*, we can simply detect all of them in the text and change them to a random alternative, or delete them altogether.[6] For the concept highlighting the *Political Figure* being discussed, this is much harder to do manually, as the chosen figure induces the topics being described in the text and is hence likely to affect other important concepts that generate the text.

Intervening on *Adjectives* as presented in the middle graph of Figure 2 relies on our ability to create a conditional generative model, one that makes sure a certain concept is or is not represented in the text. Because this is often hard to do (see Section 3.3), we propose a solution that is based on the language representation $\phi(X)$. As shown in the bottom causal graph of Figure 2, we assume that the concepts generate the

---

5 In fact, removing *Adjectives* does provide a literal measurement of their impact, but it does not provide a measurement of the more abstract notion we are interested in (which is only partially expressed through the *Adjectives*). Later we consider a baseline that does exactly this and demonstrate its shortcomings.

6 This would still require the modeler to control some confounding concepts, as *Adjectives* could be correlated with other variables (such as some *Adjectives* used to describe a specific politician).

representation $\phi(X)$ directly. This approximation shares some similarities with the idea of *Process Control* described in Pearl (2009a). While Pearl (2009a) presents *Process Control* as the case of intervening on the process affected by the treatment, it is not discussed in relation to language representations or model interpretations. Interventions on the process that is generating the outcomes are also discussed in Chapter 4 of Bottou et al. (2013), in the context of multi-armed bandits and reinforcement learning.

By intervening on the language representation, we attempt to bypass the process of generating a text given that a certain concept should or should not be represented in that text. We take advantage of the fact that modern NLP systems use pre-training to produce a language representation, and generate a counterfactual language representation $\phi^C(X)$ that is unaffected by the existence of a chosen concept C. That is, we try to change the language representation such that we get for a binary C:

$$\vec{z}(f(\phi^C(X))) = \vec{z}(f(\phi^C(X')))$$ (2)

where X and X' are identical for every generating concept, except for the concept C, on which they might or might not differ. In Section 3.3, we discuss how we intervene in the fine-tuning stage of the language representation model (BERT in our case) to produce the counterfactual representation using an adversarial component.

We now formally define our causal effect estimator. We start with the definition of the standard *Average Treatment Effect* (ATE) estimator from the causal literature. We next formally define the *causal concept effect* (CaCE), first introduced in Goyal et al. (2019a) in the context of computer vision. We then define the Example-based Average Treatment Effect (EATE), a related causal estimator for the effect of the existence of a concept on the classifier. The process required to calculate EATE is presented in the middle graph of Figure 2, and requires a conditional generative model. In order to avoid the need in such a conditional generative model, we follow the bottom graph of Figure 2 and use an adversarial method, inspired by the idea of *Process Control* that was first introduced by Pearl (2009b), to intervene on the text representation. We finally define the *Textual Representation-based Average Treatment Effect* (TReATE), which is estimated using our method, and compare it to the standard ATE estimator.[7]

3.2 The Textual Representation-based Average Treatment Effect (TReATE)

When estimating causal effects, researchers commonly measure the *average treatment effect*, which is the difference in mean outcomes between the treatment and control groups. Using *do*-calculus (Pearl 1995), we can define it in the following way:

Definition 1 (Average Treatment Effect (ATE))
The average treatment effect of a binary treatment T on the outcome Y is:

$$\text{ATE}_T = \mathbb{E}\left[Y|do(T=1)\right] - \mathbb{E}\left[Y|do(T=0)\right]$$ (3)

---

7 In Appendix B we discuss alternative causal graphs that describe different types of relationships between the involved variables. We also discuss the estimation of causal effects in such cases and briefly touch on the selection of the appropriate causal graph for a given problem.

Following the notations presented in the beginning of Section 3.1, we define the following Structural Causal Model (SCM, Pearl (2009b)) for a document X:

$$
\begin{aligned}
(C_0, C_1, \ldots, C_k) &= h(\epsilon_C) \\
X &= g(C_0, C_1, \ldots, C_k, \epsilon_X) \\
C_j &\in \{0, 1\}, \forall j \in K
\end{aligned}
$$

(4)

Where, as is standard in SCMs, ϵ_C and ϵ_X are independent variables. The function h is the generating process of the concept variables from the random variable ϵ_C and is not the focus here. The SCM in Equation (4) makes an important assumption, namely, that it is possible to intervene atomically on C_j, the *treated concept* (TC), while leaving all other concepts untouched.

We denote expectations under the interventional distribution by the standard *do*-operator notation $\mathbb{E}_g\left[\cdot|do(C_j = a)\right]$, where the subscript g indicates that this expectation also depends on the generative process g. We can now use these expectations to define *CaCE*:

Definition 2 (Causal Concept Effect (CaCE) (Goyal et al. 2019a))
The causal effect of a concept C_j on the class probability distribution $\vec{z}$ of the classifier f trained over the representation ϕ under the generative process g is:

$$
\text{CaCE}_{C_j} = \langle \mathbb{E}_g\left[\vec{z}(f(\phi(X)))|do(C_j = 1)\right] - \mathbb{E}_g\left[\vec{z}(f(\phi(X)))|do(C_j = 0)\right]\rangle
$$

(5)

where $\langle\rangle$ is the l_1 norm: A summation over the absolute values of vector coordinates.[8]

CaCE was designed to test how a model would perform if we intervene and change a value of a specific concept (e.g., if we changed the hair color of a person in a picture from blond to black). Here we address an alternative case, where some concept exists in the text and we aim to measure the causal effect of its existence on the classifier. As can be seen in the middle causal graph of Figure 2, this requires an alternative data-generating process g^{C_0}, which is not affected by the concept C_0. Using g^{C_0}, we can define another SCM that describes this relationship:

$$
\begin{aligned}
(C_0, C_1, \ldots, C_k) &= h(\epsilon_C) \\
X' &= g^{C_0}(C_1, \ldots, C_k, \epsilon'_X) \\
C_j &\in \{0, 1\}, \forall j \in K
\end{aligned}
$$

(6)

---

8 For example, for a three-class prediction problem, where the model's probability class distribution for the original example is $(0.7, 0.2, 0.1)$, while for the counterfactual example it is $(0.5, 0.1, 0.4)$, $CaCE_{C_j}$ is equal to: $|0.7 - 0.5| + |0.2 - 0.1| + |0.1 - 0.4| = 0.2 + 0.1 + 0.3 = 0.6$.

where X' is a counterfactual example generated by $g^{C_0}(C_1 = c_1, \ldots, C_k = c_k, \epsilon'_X)$. With g^{C_0}, we want to generate texts that use $(C_1 = c_1, \ldots, C_k = c_k)$ in the same way that g does, but are as if C_0 never existed. Using this SCM, we can compute the EATE:

Definition 3 (Example-based Average Treatment Effect (EATE))
The causal effect of a concept C_j on the class probability distribution $\vec{z}$ of the classifier f under the generative processes g, g^{C_j} is:

$$\text{EATE}_{C_j} = \langle \mathbb{E}_{g^{C_j}} \left[\vec{z}(f(\phi(X'))) \right] - \mathbb{E}_g \left[\vec{z}(f(\phi(X))) \right] \rangle \tag{7}$$

Implementing *EATE* requires counterfactual example generation, as shown in the middle graph of Figure 2. As this is often intractable in NLP (see Section 3.3), we do not compute *EATE* here. We instead generate a counterfactual language representation, a process which is inspired by the idea of *Process Control* introduced by Pearl (2009b) for dynamic planning. This is the case where we can only control the process generating $\phi(X)$ and not X itself.

Concretely, using the middle causal graph in Figure 2, we could have generated two examples $X_1 = g^{C_0}(C_1 = c_1, \ldots, C_k = c_k, \epsilon_{X'} = \epsilon_{x'})$ and $X_2 = g^{C_0}(C_1 = c_1, \ldots, C_k = c_k, \epsilon_{X'} = \epsilon_{x'})$ where $C_0 = 1$ for X_1 and $C_0 = 0$ for X_2, and have that $X_1 = X_2$ because the altered generative process g^{C_0} is not sensitive to changes in C_0. Notice that we require that g^{C_0} would be similar to g in the way the concepts $(C_1, \ldots, C_k)$ generate the text, because otherwise any degenerate process will do. Alternatively, in the case where we do not have access to the desired conditional generative model, we would like for the two examples $\bar{X}_1 = g(C_0 = 1, C_1 = c_1, \ldots, C_k = c_k, \epsilon_X = \epsilon_x)$ and $\bar{X}_2 = g(C_0 = 0, C_1 = c_1, \ldots, C_k = c_k, \epsilon_X = \epsilon_x)$, to have that $\phi^{C_0}(\bar{X}_1) = \phi^{C_0}(\bar{X}_2)$. That is, we follow the bottom graph from Figure 2, and intervene only on the language representation $\phi(X)$ such that the resulting representation, $\phi^{C_0}(X)$, is insensitive to C_0 and is similar to ϕ in the way the concepts $(C_1, \ldots, C_k)$ are represented. Following this intervention, we compute the TReATE.

Definition 4 (Textual Representation-based Average Treatment Effect (TReATE))
The causal effect of a concept C_j, controlling for concept C_m, on the class probability distribution $\vec{z}$ of the classifier f under the generative process g is:

$$\text{TReATE}_{C_j, C_m} = \langle \mathbb{E}_g \left[\vec{z}(f(\phi(X))) \right] - \mathbb{E}_g \left[\vec{z}(f(\phi^{C_j, C_m}(X))) \right] \rangle \tag{8}$$

where $\{C_j, C_m\}$ denotes the concept (or concepts) C_j whose effect we are estimating, and C_m the potentially confounding concept (or concepts) we are controlling for. In order to not overwhelm the notation, whenever we use only one concept in the superscript it is the concept whose effect is being estimated, and not the confounders.

In our framework, we would like to use the tools defined here to measure the causal effect of one or more concepts $\{C_0, C_1, \cdots, C_k\}$ on the predictions of the classifier f. We will do that by measuring *TReATE*, which is a special case of the ATE defined in Equation (3), where the intervention is performed via the textual representation. While *ATE* is usually used to compute the effect of interventions in randomized experiments, here we use *TReATE* to explain the predictions of a text classification model in terms of concepts.

3.3 Representation-Based Counterfactual Generation

We next discuss the reason we choose to intervene through the language representation mechanism, as an alternative to synthetic example generation. We present two existing approaches for generating such synthetic examples and explain why they are often implausible in NLP. We then introduce our approach, an intervention on the language representation, designed to ignore a particular set of concepts while preserving the information from another set of concepts. Finally, we describe how to perform this intervention using the counterfactual language representation.

Generating Synthetic Examples. Comparing model predictions on examples to the predictions on their counterfactuals is what allows the estimation of causal explanations. Without producing a version of the example that does not contain the treatment (i.e., concept or feature of interest), it would be hard to ascertain whether the classifier is using the treatment or other correlated information (Kaushik, Hovy, and Lipton 2020). To the best of our knowledge, there are two existing methods for generating counterfactual examples: manual augmentation and automatic generation using generative models.

Manual augmentation can be straightforward, as one needs to manually change every example of interest to reflect the absence or presence of a concept of choice. For example, when measuring the effect of *Adjectives* on a sentiment classifier, a manual augmentation could include changing all positive *Adjectives* into negative ones, or simply deleting all *Adjectives*. While such manipulations can sometimes be easily done with human annotators, they are costly and time-consuming and therefore implausible for large data sets. Also, in cases such as the clinical note example presented in Figure 13, it would be hard to manipulate the text such that it uses a different writing style, making it even harder to manually create the counterfactual text.

Using generative models has been recently discussed in the case of images (Goyal et al. 2019a). In this article, Goyal et al. propose using a conditional generative model, such as a conditional VAE (Lorberbom et al. 2019), to create counterfactual examples. While in some cases, such as those presented in their article, it might be plausible to generate counterfactual examples, in most cases in NLP it is still too hard to generate realistic texts with conditional generative models (Lin et al. 2017; Che et al. 2017; Subramanian et al. 2017; Guo et al. 2018). Also, for generating local explanations it is required to produce a counterfactual for each example such that all the information besides the concept of choice is preserved, something that is even harder than producing two synthetic examples, one from each concept class, and comparing them.

As an alternative to manipulating the actual text, we propose to intervene on the language representation. This does not require generating more examples, and therefore does not depend on the quality of the generation process. The fundamental premise of our method is that comparing the original representation of an example to this counterfactual representation is a good approximation of comparing an example to that of a synthetic counterfactual example that was properly manipulated to ignore the concept of interest.

Interventions on Language Representation Models. Since the introduction of pre-trained word-embeddings, there has been an explosion of research on choosing pre-training tasks and understanding their effect (Jernite, Bowman, and Sontag 2017; Logeswaran and Lee 2018; Ziser and Reichart 2018; Dong et al. 2019; Chang et al. 2019; Sun et al. 2019; Rotman and Reichart 2019). The goal of this process is to generate a representation that captures valuable information for solving downstream tasks, such as sentiment

classification, entity recognition, and parsing. Recently, there has also been a shift in focus toward pre-training contextual language representations (Liu et al. 2019; Yang et al. 2019).

Contextual embedding models typically follow three stages: (1) Pre-training: where a DNN (encoder) is trained on a massive unlabeled data set to solve self-supervised tasks; (2) Fine-tuning: an optional step, where the encoder is further trained on different tasks or data; and (3) Supervised task training: where task specific layers are trained on labeled data for a downstream task of interest.

Our intervention is focused on Stage 2. In this stage, we continue training the encoder of the model on the tasks it was pre-trained on, but add auxiliary tasks, designed to forget some concepts and remember others.[9] In Figure 3 we present an example of our proposed Stage 2, where we train our model to solve the original BERT's *Masked Language Model* (*MLM*) and *Next Sentence Prediction* (*NSP*) tasks, along with a *Treated Concept* objective, denoted in the figure as *TC*. In order to preserve the information regarding a potentially confounding concept, we use an additional task denoted in the figure as *CC*, for *Controlled Concept*.

To illustrate our intervention, we can revisit the *Adjectives* example of Figure 1, and consider a case where we want to test whether their existence in the text affects the classification decision. To be able to estimate this effect, we traditionally would have to produce for each example in the test set an equivalent example that does not contain *Adjectives*. In terms of our intervention on the language representation, we should be able to produce a representation that is unaffected by the existence of *Adjectives*, meaning that the representation of a sentence that contains *Adjectives* would be identical to that of the same sentence where *Adjectives* are excluded. Taking that to the fine-tuning stage, we could use adversarial training to "forget" *Adjectives*.

Concretely, we add to BERT's loss function a negative term for the target concept and a positive term for each control concept we consider. As shown in Equation (9), in the case of the example from Figure 1, this would entail augmenting the loss function with two terms: adding the loss for the *Political Figure* classification *PF* (the *CC* head), and subtracting that of the *Is Masked Adjective* (*IMA*) task (the *TC* head). As we are using the *IMA* objective term in our *Adjectives* experiments (Section 5), and not only in the running example, we describe the task below. For the *Political Figure* (*PF*) concept, we could simply use a classification task where for each example we predict the political orientation of the politician being discussed.[10] With those tasks added to the loss function, we have that:

$$\mathcal{L}(\theta_{bert}, \theta_{mlm}, \theta_{nsp}, \theta_{cc}, \theta_{tc}) = \frac{1}{n}\left(\sum_{i=1}^{n}\mathcal{L}_{mlm}^{i}(\theta_{bert}, \theta_{mlm}) \quad + \sum_{i=1}^{n}\mathcal{L}_{nsp}^{i}(\theta_{bert}, \theta_{nsp})\right.$$

$$\left. + \sum_{i=1}^{n}\mathcal{L}_{cc}^{i}(\theta_{bert}, \theta_{cc}) \quad -\lambda\sum_{i=1}^{n}\mathcal{L}_{tc}^{i}(\theta_{bert}, \theta_{tc})\right) \quad (9)$$

where θ_{bert} denotes all of BERT's parameters, except those devoted to θ_{mlm}, θ_{nsp}, θ_{tc}, and θ_{cc}. λ is a hyperparameter which controls the relative weight of the adversarial task.

---

9 Continued pre-training has shown to be useful in NLP more generally (Gururangan et al. 2020; Gardner et al. 2020).

10 For the *CC* objective, we can add any of the classification tasks suggested above for *PF* (*CC*), following the definition of the world model (i.e., the causal graph) the researcher is assuming.
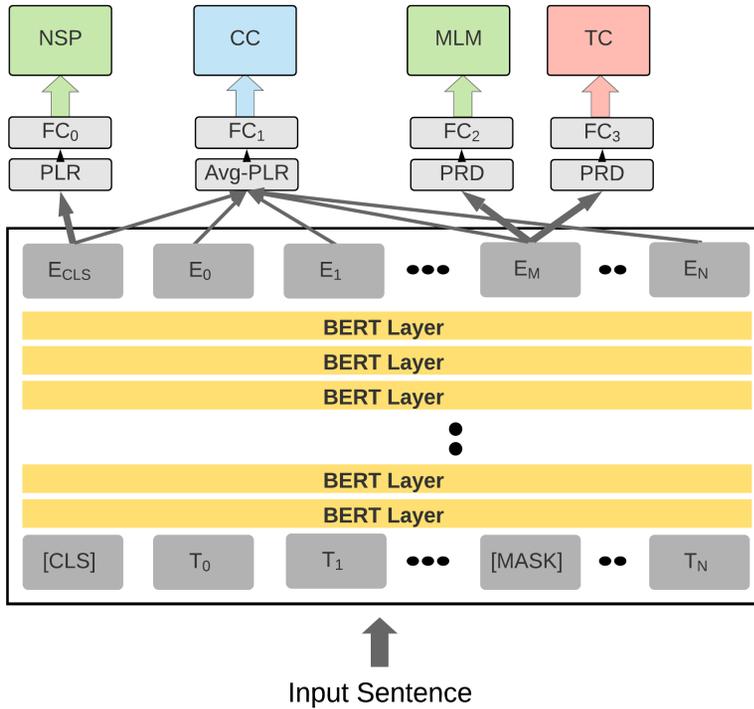
Figure 3
An illustration of our Stage 2 fine-tuning procedure for our counterfactual representation model (*BERT-CF*). In this representative case, we add a task, named *Treated Concept* (TC), which is trained adversarially. This task is designed to "forget" the effect of the treated concept, as in the *IMA* adversarial task discussed in Section 5. To control for a potential confounding concept (i.e., to "remember" it), we add the *Control Concept* (CC) task, which predicts the presence of this concept in the text, as in the *PF* task discussed below. *PRD* and *PLR* stand for BERT's prediction head and the pooler head, respectively, *AVG − PLR* for an average pooler head, FC is a fully connected layer, and [*MASK*] stands for masked tokens embeddings. *NSP* and *MLM* are BERT's next prediction and masked language model objectives. The results of this training stage is our counterfactual *BERT-CF* model.

One way of implementing the *IMA TC* head is inspired by BERT's *MLM* head. That is, masking *Adjectives* and *Non-adjectives*, then predicting whether the masked token is an adjective. Following the *gradient reversal* method (Ganin et al. [2016], henceforth DANN),[11] we add this task with a layer that leaves the input unchanged during forward propagation, yet reverses its corresponding gradients by multiplying them with a negative scalar ($-\lambda$) during back propagation.

   The core idea of DANN is to reduce the domain gap, by learning common representations that are indistinguishable to a domain discriminator (Ghosal et al. 2020). In our model, we replace the domain discriminator with a discriminator that discriminates examples with the treated concept from examples that do not have that concept. Following DANN, we optimize the underlying BERT representations jointly with classifiers operating on these representations: The task classifiers perform the main task of the model

---

11 See Equations 9–10 and 13–15 in Ganin et al. (2016).

($\mathcal{L}_{mlm}$, $\mathcal{L}_{nsp}$, and $\mathcal{L}_{cc}$ in our objective) and the treatment concept classifier discriminates between those masked tokens that are adjectives and those that are not (the $\mathcal{L}_{tc}$ term in our objective). While the parameters of the classifiers (θ_{mlm}, θ_{nsp}, θ_{cc}, θ_{tc}) are optimized in order to minimize their training error, the language encoder parameters (θ_{bert}) are optimized in order to minimize the loss of the task classifiers ($\mathcal{L}_{mlm}$, $\mathcal{L}_{nsp}$, and $\mathcal{L}_{cc}$) and to maximize the loss of the treatment concept classifier ($\mathcal{L}_{tc}$). Concretely in our case, the parameters of the underlying language representation θ_{bert} are simultaneously optimized in order to minimize the *MLM*, *NSP*, and *PF* loss functions and maximize the *IMA* loss. *Gradient reversal* hence encourages an adjective-invariant language representation to emerge. For more information about the adversarial multitask min-max optimization dynamics, and the emergent concept-invariant language representations, see Xie et al. (2017).

While the *gradient reversal* method is widely implemented throughout the domain adaptation literature (Ramponi and Plank 2020), it has also been previously shown that it can be at odds with the model's main prediction objective (Elazar and Goldberg 2018). However, we implement it in our model's training process in a different way than in most previous literature. We use this method as part of the language model fine-tuning stage, which is independent of and precedes the downstream prediction task's objective training. Therefore, our adversarial task's objective is not directly at odds with the downstream model's prediction objective.

Having optimized the loss functions presented in Equation (9), we can now use the resulting counterfactual representation model and compute the *individual treatment effect* (ITE) on an example as follows. We compute the predictions of two different models: One that uses the original BERT, that has not gone through our counterfactual fine-tuning, and one that uses the counterfactual BERT model (BERT-CF). *The Textual Representation-based ITE* (TRITE) is then the average of the absolute differences between the probabilities assigned to the possible classes by these models. As *TReATE* is presented in Equation (8) in expectation form, we compute our estimated $\widehat{TReATE}$ by summing over $\widehat{TRITE}$ for the set of all test-set examples, I:

$$\widehat{TReATE}_{TC,CC} = \frac{1}{|I|} \sum_{i \in I} \widehat{TRITE}^{i}_{TC,CC} =$$

$$\frac{1}{|I|} \sum_{i \in I} \langle \vec{z}\big(f(\phi^{TC,CC}(X = x_i))\big) - \vec{z}\big(f(\phi(X = x_i))\big) \rangle \qquad (10)$$

where x_i is the specific example, ϕ is the original language representation model, and $\phi^{TC,CC}$ is the counterfactual *BERT-CF* representation model, where the intervention is such that *TC* has no effect and *CC* is preserved. $\vec{z}\big(f(\phi(X))\big)$ is the class probability distribution of the classifier f when using ϕ as the representation model for example X.

4. Data

When evaluating a trained classification model, we usually have access to a test set, consisting of manually labeled examples that the model was not trained on, and can hence be used for evaluation. Estimating causal effects is often harder in comparison, as we do not have access to the ground truth. In the case of causal inference, we can generally only identify effects if our assumptions on the data-generating process, such as those presented in Figure 2, hold. This means that at the core of our causal model

explanation paradigm is the availability of a causal graph that encodes our assumptions about the world. Notice, however, that non-causal explanation methods that do not make assumptions about the world are prone to finding arbitrary correlations, a problem that we are aiming to avoid with our method.

To allow for ground-truth comparisons and to spur further research on causal inference in NLP, we propose here four cases where causal effects can be estimated. In three out of those cases, we have constructed data sets with counterfactual examples so that the causal estimators can be compared to the ground truth. We start here by introducing the data sets we created and discuss the choices made in order to allow for proper evaluation. Section 4.1 describes the sentiment analysis data with the *Adjectives* and *Topics* concepts, while Section 4.2 describes the EEEC data set for mood classification with the *Gender* and *Race* concepts. Section 5 presents the tasks for which we estimate the causal effect, and the resulting experiments.[12]

4.1 Product and Movie Reviews

Following the running example of Section 1, we start by looking for prominent sentiment classification data sets. Specifically, we look for data sets where the domain entails a rich description where *Adjectives* could play a vital role. With enough variation in the structure and length of examples, we hope that *Adjectives* would have a significant effect. Another key aspect is the number of training examples. To be able to amplify the correlation between the treated concept (*Adjectives*) and the label, we need to be able to omit some training examples. For instance, if we omit most of the positive texts describing a *Political Figure*, we can create a correlation between the negative label and that politician. We need a data set that will allow us to do that and still have enough training data to properly train modern DNN classification models.

We also wish to estimate the causal effect of the concept of *Topics* on sentiment classification (see Section 5 for an explanation on how we compute the topic distribution). To be able to observe the causal effect of *Topics*, some variation is required in the *Topics* discussed in the texts. For that, we use data originating from several different domains, where different, unrelated products or movies are being discussed. In this section we focus on the description of the data set we have generated, and explain how we manipulate the data in order to generate various degrees of concept-label correlations.

Considering these requirements and the concepts for which we wish to estimate the causal effect on model performance, we choose to combine two data sets, spanning five domains. The product data set we choose is widely used in the NLP domain adaptation literature, and is taken from Blitzer, Dredze, and Pereira (2007). It contains four different domains: *Books*, *DVD*, *Electronics*, and *Kitchen Appliances*. The movie data set is the IMDb movie review data set, taken from Maas et al. (2011). In both data sets, each example consists of a review and a rating (0–5 stars). Reviews with *rating* > 3 were labeled positive, those with *rating* < 3 were labeled negative, and the rest were discarded because their polarity was ambiguous. The product data set is comprised of 1,000 positive and 1,000 negative examples for each of the four domains, for a total of 4,000 positive and 4,000 negative reviews. The *Movies* data set is comprised of 25,000 negative and 25,000 positive reviews. To construct our combined data set, we randomly sample 1,000 positive and 1,000 negative reviews from the *Movies* data set and add these

---

12 Our data sets are available at: `https://www.kaggle.com/amirfeder/causalm`.

alongside the product data set reviews. Our final combined data set amounts to a total of 10,000 reviews, balanced across all five domains and both labels.

We tag all examples in both data sets for the Part-of-Speech (*PoS*) of each word with the automatic tagger available through *spaCy*,[13] and use the predicted labels as ground truth. For each example in the combined data set, we generate a counterfactual example for *Adjectives*. That is, for each example we create another instance where we delete all words that are tagged as *Adjectives*, such that for the example: "*It's a lovely table*," the counterfactual example will be: "*It's a table*." Finally, we count the number of *Adjectives* and other *PoS* tags, and create a variable indicating the ratio of *Adjectives* to *Non-adjectives* in each example, which we use in Section 5 to bias the data.

For the *Topic* concepts, we train a Latent Dirichlet Allocation (LDA) topic model (Blei, Ng, and Jordan 2003)[14] on all the data in our combined data set and optimize the number of topics for maximal coherence (Lau, Newman, and Baldwin 2014), resulting in a set of $T = 50$ topics. For each of the five domains we then search for the *treatment concept* topic t_{TC}, which we define as the topic that is relatively most associated with that domain, that is, the topic with the largest difference between the probability assigned to examples from that domain and the probability assigned to examples outside of that domain, using the following equation:

$$t_{TC}(d) = \arg\max_{t \in T} \Big(\frac{1}{|I_{d_+}|}\sum_{i \in I_{d_+}} \theta_t^i - \frac{1}{|I_{d_-}|}\sum_{i \in I_{d_-}} \theta_t^i\Big) \tag{11}$$

where d is the domain of choice, t is a topic from the set of topics T, θ_t is the probability of topic t, and I_{domain} is the set of examples for a given domain. I_{d_+} is the set of examples in domain d, and I_{d_-} the set of examples outside of domain d. After choosing t_{TC}, we exclude it from T and use the same process to choose t_{CC}, our *control concept* topic.

For each *Topic*, we also compute the median probability on all examples, and define a binary variable indicating for each example whether the *Topic* probability is above or below its median. This binary variable can then be used for the *TC* and *CC* tasks described in Section 5.

In Table 1 we present some descriptive statistics for all five domains, including the *Adjectives* to *Non-adjectives* ratio and the median probability (θ_{domain}) of the $t_{TC}(d)$ topic for each domain. As can be seen in this table, there is a significant number of *Adjectives* in each domain, but the variance in their number is substantial. Also, *Topics* are domain-specific, with the most correlated topic $t_{TC}(d)$ for each domain being substantially more visible in its domain compared with the others. In Table 2 we provide the top words for all *Topics*, to show how they capture domain-specific information.

Our sentiment classification data allows for a natural setting for testing our methods and hypotheses, but it has some limitations. Specifically, in the case of *Topics*, we cannot generate realistic counterfactual examples and therefore compute ATE_{gt}, the ground-truth estimator of the causal effect. This is because creating counterfactual examples would require deleting the topic from the text without affecting the grammaticality of the text, something that cannot be done automatically. In the case of *Adjectives*, we are hoping that removing *Adjectives* will not affect the grammaticality of the original text, but are aware that this sometimes might not be the case. While this data provides a real-world example of natural language, it is hard to automatically generate counterfactuals

---

13 `https://spacy.io/`.
14 Using the *gensim* library (Řehůřek and Sojka 2010).

Table 1
Descriptive statistics for the Sentiment Classification data sets. $r(adj)$ denotes the ratio of *Adjectives* to *Non-adjectives* in an example. θ_{domain} is the mean probability of the topic that is most observed in that domain, which will also serve as our *treated topic*. b, d, e, k, m are abbreviations for Books, DVD, Electronics, Kitchen, and Movies.

Domain	Min. $r(adj)$	Med. # $r(adj)$	Max. # $r(adj)$	σ of # $r(adj)$	θ_b	θ_d	θ_e	θ_k	θ_m
Books	0.0	0.135	0.444	0.042	0.311	0.011	0.052	0.026	0.014
DVD	0.0	0.138	0.425	0.042	0.014	0.045	0.045	0.016	0.225
Electronics	0.0	0.136	0.461	0.049	0.010	0.065	0.080	0.039	0.003
Kitchen	0.0	0.142	0.500	0.052	0.007	0.039	0.075	0.066	0.002
Movies	0.0	0.138	0.666	0.0333	0.010	0.007	0.045	0.016	0.281

for it. To allow for a more accurate estimation of the ground truth effect, we would need a data set where we can control the data-generating process.

4.2 The Enriched Equity Evaluation Corpus (EEEC)

Understanding and reducing gender and racial bias encapsulated in classifiers is a core task in the growing literature of interpretability and debiasing in NLP (see Section 2). There is an ongoing effort to both detect such bias and to mitigate its effect, which we see from a causal perspective as a call for action. By offering a way to estimate the causal effect of the *Gender* and *Race* concepts as they appear in the text, on classifiers, we enable researchers to avoid using biased classifiers.

In order to evaluate the quality of our causal effect estimation method, we need a data set where we can control test examples such that for each text we have a counterfactual text that differs only by the *Gender* or *Race* of the person it discusses. We also need to be able to control the data-generating process in the training set, so that we can create such a bias for the model to pick up. A data set that offers such control exists, and is called the Equity Evaluation Corpus (EEC) (Kiritchenko and Mohammad 2018).

It is a benchmark data set, designed for examining inappropriate biases in system predictions, and it consists of 8,640 English sentences chosen to tease out *Racial* and *Gender* related bias. Each sentence is labeled for the mood state it conveys, a task also known as *Profile of Mood States* (POMS). Each of the sentences in the data set is composed using one of eleven templates, with placeholders for a person's name and the emotion it conveys. For example, one of the original templates is: "*<Person> feels <emotional state word>.*" The name placeholder (*<Person<*) is then filled using a pre-existing list of common names that are tagged as male or female, and as African American or European.[15] The emotion placeholder (*<emotional state word<*) is filled using lists of words, each list corresponding to one of four possible mood states: *Anger*, *Sadness*, *Fear*, and *Joy*. The label is the title of the list from which the emotion is taken.

Designed as a bias detection benchmark, the sentences in EEC are very concise, which can make them not useful as training examples. If a classifier sees in training only a small number of examples, which differ only by the name of the person and

---

15 In this article we take a binary approach toward race and gender, as is done in the EEC (Kiritchenko and Mohammad 2018), although this is obviously not the case in reality. This helps us keep the task and experiments clear, and easy to follow and analyze.

Table 2
Top 10 words in each of the 50 topics. A topic model was trained on all texts in all domains combined. Topic #22, our θ_m, is highlighted in red, topic #38, θ_b, is highlighted in blue, topic #8, θ_d, is highlighted in green, topic #2, θ_e, is highlighted in orange, and topic #13, θ_k, is highlighted in purple. b, d, e, k, m are abbreviations for the Books, DVD, Electronics, Kitchen, and Movies domains.

#					Top 10 Words					
1	set	box	wait	20	making	flat	worth	longer	disappoint	spend
2	pan	phone	computer	work	does	use	still	non	battery	problem
3	great	use	just	months	problem	bought	time	good	years	ago
4	classic	stories	know	great	really	book	definitely	reading	writing	long
5	item	dull	returned	expect	given	fit	did	ridiculous	run	matter
6	kids	crap	turned	fun	children	making	point	needs	understand	truly
7	dvd	version	video	player	original	screen	release	quality	features	cover
8	book	real	second	school	author	going	page	shows	past	light
9	machine	software	uses	issues	using	help	problems	makes	device	bought
10	mind	fine	despite	pages	author	lost	books	book	read	especially
11	book	reading	information	read	quot	books	better	author	know	does
12	just	did	know	ll	does	ve	think	got	times	work
13	product	buy	amazon	bought	plastic	did	reviews	cheap	work	ve
14	does	man	just	woman	story	women	way	stop	time	like
15	expected	star	series	rest	terrible	simply	pretty	watching	paid	wait
16	away	water	stay	model	dog	good	difficult	like	right	just
17	broke	replacement	warranty	month	send	weeks	days	called	week	product
18	people	god	says	mr	life	like	world	person	american	way
19	return	garbage	single	different	unless	given	oh	hot	plastic	thought
20	play	does	power	light	white	little	used	make	drive	large
21	bad	good	pretty	really	ve	just	worst	seen	10	best
22	movie	film	like	movies	acting	bad	watch	just	plot	scenes
23	fan	wrote	fans	years	special	true	humor	day	disappoint	novel
24	order	received	monster	performance	ordered	sent	said	better	later	returned
25	book	long	ll	just	tell	totally	later	reader	given	great
26	book	job	person	poor	read	kept	thought	trying	boring	good
27	new	piece	tried	stopped	junk	worked	working	work	brand	maybe
28	line	john	coming	certainly	early	true	films	enjoy	like	write
29	book	read	books	author	pages	novel	writing	reader	history	interesting
30	killer	card	camera	car	shows	stupid	series	make	picture	better
31	coffee	mouse	stand	products	use	like	make	decided	finally	tried
32	john	writing	movie	book	waste	time	plot	make	did	line
33	quot	written	self	does	things	view	needs	like	new	hope
34	book	let	good	make	did	interesting	does	say	self	great
35	unit	device	purchased	features	works	house	returned	running	warranty	hear
36	does	hand	nice	need	small	clean	time	sex	look	things
37	quality	poor	daughter	cable	low	design	control	sound	bad	good
38	boring	long	time	end	story	rest	stop	slow	minutes	good
39	old	year	horrible	great	got	food	beautiful	boy	said	instead
40	hard	happy	sure	disappoint	writing	music	bad	reviews	days	uses
41	known	christian	truth	like	feel	store	novel	remember	stay	able
42	mouse	design	15	agree	purchased	given	job	happened	order	making
43	world	war	words	self	old	word	attempt	needed	title	life
44	lost	christian	guys	despite	turn	getting	mind	decent	war	fine
45	music	ipod	weak	car	30	battery	playing	takes	able	major
46	like	just	really	did	characters	story	character	love	little	make
47	money	waste	time	save	thought	worth	spend	better	good	just
48	disappointed	feel	fast	little	bit	good	job	parts	matter	complete
49	day	black	sound	hours	like	just	minutes	bread	went	getting
50	service	support	customer	told	product	check	company	called	terrible	hold

the emotion word, it could easily memorize a mapping between emotion words and labels, and will not learn anything else. To solve this and create a more representative and natural data set for training, we expand the EEC data set, creating an enriched data set which we denote as *Enriched Equity Evaluation Corpus*, or EEEC. In this data set, we use the 11 templates of EEC and randomly add a prefix or suffix phrase, which can describe a related place, family member, time, and day, including also the corresponding

pronouns to the *Gender* of the person being discussed. We also create 13 non-informative sentences, and concatenate them before or after the template such that there is a correlation between each label and three of those sentences.[16] This is performed so that we have other information that could be valuable for the classifier other than the person's name and the emotion word. Also, to further prevent memorization, we include emotion words that are ambiguous and can describe multiple mood states.

Our enriched data set consists of 33,738 sentences generated by 42 templates that are longer and much more diverse than the templates used in the original EEC. While still synthetic and somewhat unrealistic, our data set has much longer sentences, has more features that are predictive of the label, and is harder for the classifier to memorize. In Appendix C we provide additional details about the EEEC data set, through two tables: one that presents the templates used to generate the data, and one that compares the original EEC to our EEEC, illustrating the key modifications we have made.

For each example in EEEC we generate two counterfactual examples: one for *Gender* and one for *Race*. That is, we create two instances that are identical except for that specific concept. For the *Gender* case, we change the name and the *Gender* pronouns in the example and switch them, such that for the original example: "*Sara feels excited as she walks to the gym*" we will have the counterfactual example: "*Dan feels excited as he walks to the gym*." For the *Race* concept, we create counterfactuals such that for the same original example, the counterfactual example is: "*Nia feels excited as she walks to the gym*." For each counterfactual example, the person's name is taken at random from the pre-existing list corresponding to its type.

5. Tasks and Experiments

Equipped with data sets for both Sentiment Classification and POMS, and annotated for concepts (*Adjectives*, *Topics*, *Gender*, and *Race*), we now define tasks and experiments for which we train classification models and test our proposed method for causal effect estimation of chosen concepts. In three of those cases (*Adjectives*, *Gender*, and *Race*) we have some control over the data-generating process, and therefore can compare the estimated causal effect to the ground truth effect. We start with experiments designed to estimate the effect of two concepts, *Adjectives* and *Topics*, on sentiment classification. We choose these concepts as representatives of local (*Adjectives*, expressed as individual words or short phrases) and global (*Topics*, expressed as distribution over the vocabulary) concepts that are intuitively related to sentiment analysis. Then, we explore the potential role of gender and racial bias in mood state classification. For each concept, we experiment with three versions of the data: *Balanced*, *Gentle*, and *Aggressive*, which differ by the correlation between the *treated concept* and the label. In Table 3, we summarize the four *treated concepts* we experiment with. Table 4 presents the differences between the experiments we conduct for each *treated concept* in terms of the concept-label correlation.

With this experimental setup we seek to answer four research questions:

1.   Can we accurately approximate ATE_{gt}, the ground-truth estimator of the causal effect, using our proposed *TReATE* estimator ?

---

16 Each of those three sentences are five times more likely to appear than the other ten for that label.

Table 3
Summary of the tasks we experiment with. PoS stands for Part of Speech, POMS for Profile of Mood States, and EEC for Equity Evaluation Corpus. For each of the four tasks, we describe the task designed in order to forget the concept, alongside tasks designed to control against forgetting potential confounders.

Concept	Task	Adversarial Task	Optional Control Tasks	Data set
Adjectives	Sentiment	Masked Adjectives	PoS Tagging	Movie Reviews
Topics	Sentiment	Above Average Topic Prob.	Topic Class.	All Reviews
Gender	POMS	Gender Class.	Race Class.	Enriched EEC
Race	POMS	Race Class.	Gender Class.	Enriched EEC

Table 4
The correlation between the *treated concept* and the label for each experiment we run (Balanced, Gentle, and Aggressive). For each experiment we present the correlation on the full data set (train, dev, and test combined).

Treated Concept	Label	Concept-Label Correlation		
		Balanced	Gentle	Aggressive
Adjectives	Sentiment	0.056	0.400	0.760
Topic	Sentiment	0.002	0.046	0.127
Gender	POMS	0.001	0.074	0.245
Race	POMS	0.005	0.069	0.242

2. Does *BERT-CF*, our counterfactual representation model, forget the *treated concept*?

3. Does *BERT-CF* remember the *control concept*?

4. Can *BERT-CF* help mitigate potential bias in the downstream classifier?

In answering these questions, we hope to show that our method can provide accurate causal explanations that can be used in a variety of settings. Question 1 is our core causal estimation question, where we wish to test whether the ground truth *ATE* can be approximated with *TReATE*. Questions 2 and 3 are important because we would like to know that the estimated effect we see in question 1 is a result of our Stage 2 intervention that created *BERT-CF* (see Figure 3), and not due to other reasons. Unlike question 1, questions 2 and 3 do not require access to counterfactual examples, and can be used to validate our method in real-world settings. Finally, a byproduct of our method is *BERT-CF*, a counterfactual representation model that is unaffected by the *treated concept*. In question 4 we ask if such a representation model can be useful in mitigating the perhaps unwanted effect of the *treated concept* on the task classifier.

To tackle these questions, we start by describing how to estimate the causal effect for each of the *treated concepts* while considering the potentially confounding *control concepts* (question 1). For each *treated concept*, we explain how we control the concept-label correlation to create the *Balanced*, *Gentle*, and *Aggressive* versions. We then discuss how to answer questions 2 and 3 for a given *TC* and *CC*, and briefly explain how we answer question 4 in the *Aggressive* version. We detail our experimental pipeline and hyperparameters in Appendix D.

5.1 The Causal Effect of Adjectives on Sentiment Classification

Following the example we discuss in Section 1, we choose to measure the effect of *Adjectives* on sentiment classification. In using *Adjectives* as our *treated concept*, we follow the discussion in the sentiment classification literature that marks them as linguistic features with a prominent effect. Another key characteristic of *Adjectives* is that they can usually be removed from a sentence without affecting the grammaticality of the sentence and its coherence. Finally, with the recent advancement of parts-of-speech (*PoS*) taggers (Akbik, Blythe, and Vollgraf 2018), we can rely on automatic models to tag our data set with high accuracy, thus avoiding the need for manual tagging.

The causal graph we use to guide our choice of the *treated* and *control concepts* is similar to that of our motivating example, and is illustrated in Figure 4. In the Sentiment reviews data set (presented in Section 4.1), since there are no concepts such as *Political Figure* being discussed, we use other *PoS* tags (i.e., everything but *Adjectives*) as our *control concepts*.

Controlling the Concept-Label Correlation. Using the reviews data set, we create multiple data sets, differing by the correlation between the ratio of *Adjectives* and the label. We split the original data set into training, development, and test sets following a 64%, 16%, 20% (37,120, 9,280, 11,600 sentences) split, respectively. Then, we create three versions of the data: *Balanced*, *Gentle*, and *Aggressive*. In the *Balanced* version we utilize all reviews regardless of the ratio of *Adjectives* they contain, preserving the data-driven
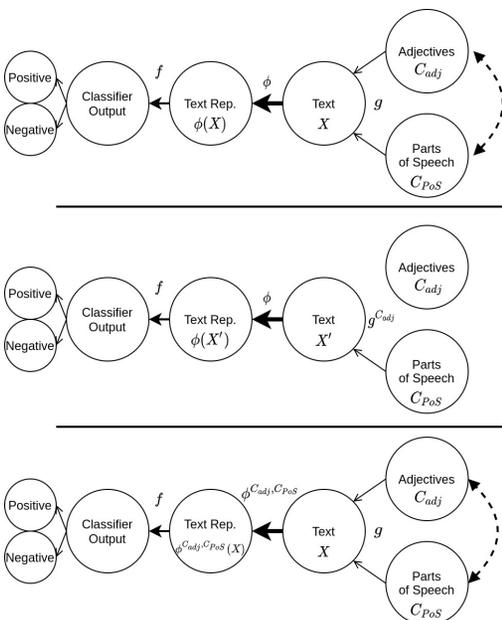

Figure 4
A causal graph for *Adjectives* and other *Parts-of-Speech* generating a text with a positive or a negative sentiment. The top graph represents a data-generating process where all *Parts-of-Speech* generate the texts, with a potential hidden confounder affecting both C_{adj}, the *Treated Concept*, and C_{PoS}, the *Control Concept*. The middle graph represents the scenario where we can control the generation process and create a text that is not influenced by the *Treated Concept*. The bottom graph represents our approach, where we manipulate the text representation.

correlation between the concept (*Adjectives*) and the label (sentiment class). In the *Gentle* version, we sort sentences from the *Balanced* version by the ratio of *Adjectives* they contain (in descending order) and delete the top half of the list for the sentences that appear within negative reviews, thus creating a negative correlation between the ratio of *Adjectives* and the negative label in the train, development, and test sets. For the *Aggressive* version we do the same, and also delete the bottom half of the list for the sentences that appear within positive reviews, resulting in a higher correlation between the ratio of *Adjectives* and the positive labels (see Table 4).

Modeling the TC and the CC. We follow the causal graph presented in Figure 4 and implement the adversarial *Is Masked Adjective* (*IMA*) as our TC objective shown in Equation (9). The *IMA* objective is very similar to the *MLM* objective, and it utilizes the same *[MASK]* token used in *MLM*, which masks each token to be predicted. However, instead of predicting the masked word we predict whether or not it is an adjective. To accommodate the *IMA* prediction objective for any given input text, we masked all *Adjectives* in addition to an equal number of non-adjective words, to ensure the results are a balanced binary classification token-level task. We follow the same probabilities suggested for the *MLM* task in Devlin et al. (2019).[17]

For the *control concept* (CC) task, we utilize all *PoS* tags apart from *Adjectives*, and train a sequence tagger to classify each *Non-adjective* word according to its *PoS*.[18] This serves the purpose of preserving syntactic concepts other than *Adjectives*. In Section 6 we discuss the effect of the *CC* objective on our estimates. Finally, as explained in Section 3.3 (see Equation (9)), to produce the *BERT-CF* model for *Adjectives*, we adversarially train the *IMA* objective jointly with the other terms of the objective that are trained in a standard manner.

5.2 The Causal Effect of Topics on Sentiment Classification

Another interesting concept that we can explore using the reviews data set is *Topics*, as captured by the LDA model (Blei, Ng, and Jordan 2003). *Topics* capture high-level semantics of documents, and are widely used in NLP for many language understanding purposes (Boyd-Graber, Hu, and Mimno 2017; Oved, Feder, and Reichart 2020). *Topics* are qualitatively different from *Adjectives*, as *Adjectives* are concrete and local while *Topics* are abstract and global. In the context of sentiment classification, it is reasonable to assume that the *Topic* being discussed has an effect on the probability of the review being positive or negative. For example, some movie genres generally get more negative reviews than others, and some products are more generally liked than their alternatives. A key advantage of *Topics* for our purposes is that they can be trained without supervision, allowing us to test this concept without manual tagging.

Topics are global concepts that encode information across the different reviews in the corpus. Yet, by using topic modeling we can represent them as variables that come with a probability that reflects the extent to which they are represented in each document. This allows us to naturally integrate them into our *TC* term presented in Figure 3 (i.e., the *treated concept*), but also to the preserving *CC* term (the *control concept*). In Figure 5, we illustrate the causal graph that we follow. For the TC and CC *Topics*, we follow

---

17 The probabilities used in the original BERT paper are: 80%, 10%, and 10% for masking the token, keeping the original token, or changing it to a random token, respectively.
18 To prevent the model from learning to associate the null label with *Adjectives*, we do not add it to the loss.
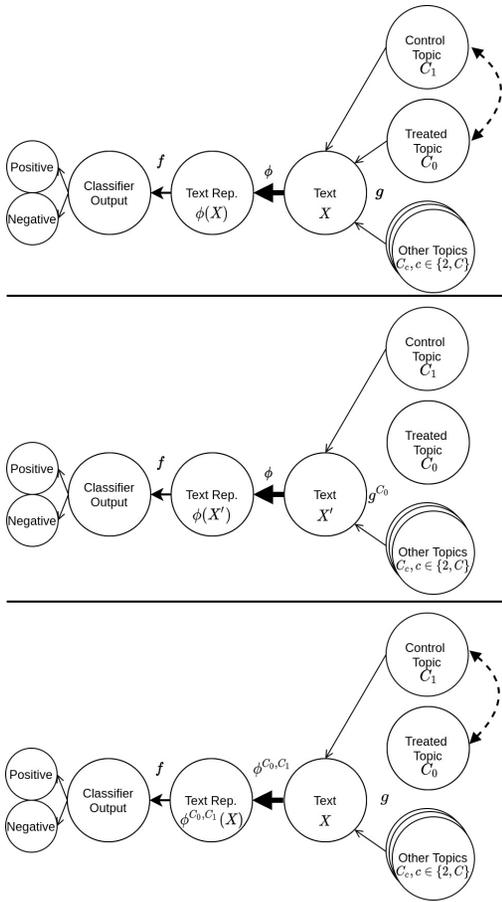
Figure 5
A causal graph for *Topics* generating a review with a positive or negative sentiment. The top graph represents a data-generating process where *Topics* generate texts, with a potential hidden confounder affecting both C_0, the *Treated Topic*, and C_1, the *Control Topic*. The middle graph represents the scenario where we can control the generation process and create a text without the *Treated Topic*. The bottom graph represents our approach, where we manipulate the text representation.

Equation (11) and use the *Topics* $t_{TC}(domain)$ and $t_{CC}(domain)$, which we denote as C_0 and C_1, respectively.

Unlike the *Adjectives* experiments, we cannot directly manipulate the texts to create counterfactual examples for *Topics*. For a given document, changing the topic being discussed cannot be done by simply deleting the words associated with it, and would require rewriting the text completely. As an alternative, we can use the domain variation in the reviews data set and the correspondence of some *Topics* to specific domains, to test the performance of our causal effect estimator, *TReATE*. We see this as a unique contribution of this experiment as it allows us to test our causal effect estimator in a case where we do not have access to the ground-truth (estimation of the) causal effect.

Another issue with *Topics* is that they are confounders for one another by design. LDA models texts as mixtures of *Topics*, where each *Topic* is a probability distribution

over the vocabulary. As *Topics* are on the simplex (they are a probability distribution), if the probability of one *Topic* decreases, the probability of others must increase. For example, if the example presented in Section 1 was less about politics, it would have to be more about a different *Topic*. Below we show how we circumvent the effect of those potential confounders in our TC and CC objectives as shown in Equation (9).

Controlling the Concept-Label Correlation. For the *Topics* experiments, we also create three versions of the data, following a similar *Balanced*, *Gentle*, and *Aggressive* approaches and using the reviews data as above. For the *Balanced* version, we use all of the data from *Books*, *DVD*, *Electronics*, *Kitchen Appliances*, and *Movies* domains. For the *Gentle* version, we take the *Balanced* data set and delete half of the negative reviews where the $t_{TC}(Movies)$ topic is less represented (with probability lower than the median probability), resulting in a positive correlation between the topic and the positive label. For the *Aggressive* version we also delete half of the positive reviews where the $t_{TC}(Movies)$ topic is more represented, thus further increasing the correlation between the topic and the labels. For all these experiments we follow the same 64%, 16%, 20% split for the training, development, and test sets, respectively, as for the *Adjectives* experiments. As another set of experiments, we follow the same steps for the *Gentle* and *Aggressive* versions where the chosen topic is $t_{TC}(Books)$ instead of $t_{TC}(Movies)$.

   As we do not have access to real counterfactual examples in this case, we can only compute *TReATE* for a given test set and qualitatively analyze the results. Particularly, the multidomain nature of our data set allows us to estimate *TReATE* on each domain separately, and test whether the estimated effect varies between domains. Specifically, we can test whether for a given $t_{TC}(domain)$ the $TReATE_{t_{TC}(domain)}$ estimator is higher on domains where $t_{TC}(domain)$ is more present, compared with those domains where it is less present. To do that, we compute the estimated *TReATE* for each $t_{TC}(domain)$ (*Books* and *Movies*) on each of the five domains separately, and discuss the results in Section 6. We focus most of the discussion on these experiments in Section 6.2, where we test whether we can successfully mitigate the bias introduced in the *Gentle* and *Aggressive* version.

Modeling the TC and the CC. Using the binary variables indicating if for a given topic the probability is above or below its median (see Section 4), we introduce "*Is Treated Topic*" (*ITT*), a binary adversarial fine-tuning task for our TC. As the *TC*, we choose the $t_{TC}(domain)$ topic introduced in Section 4 in Equation (11). To control for the potential forgetting of related *Topics*, we add alongside the adversarial task the prediction of the second most correlated topic, $t_{CC}(domain)$, as our CC, and add it as another fine-tuning that we name "*Is Control Topic*" (*ICT*). Finally, as explained in Section 3.3 (see Equation (9)), to produce the *BERT-CF* model for *Topics*, we adversarially train the *ITT* objective jointly with the other objective terms that are trained in a standard manner.

5.3 The Causal Effect of Gender and Racial Bias

Although *Adjectives* and *Topics* capture both local and global linguistic concepts, our ability to generate counterfactual examples for them is limited. Particularly, for *Topics* we cannot generate counterfactual examples, whereas for *Adjectives* we use real-world data and hence our control on the data generating process is limited. To allow for a more accurate comparison to the true causal effect, we consider two tasks, *Gender* and *Race*, where such a comparison can be made using the EEEC data set presented in Section 4. In Figure 6, we illustrate the causal graph for the case where *Gender* is the TC. We denote
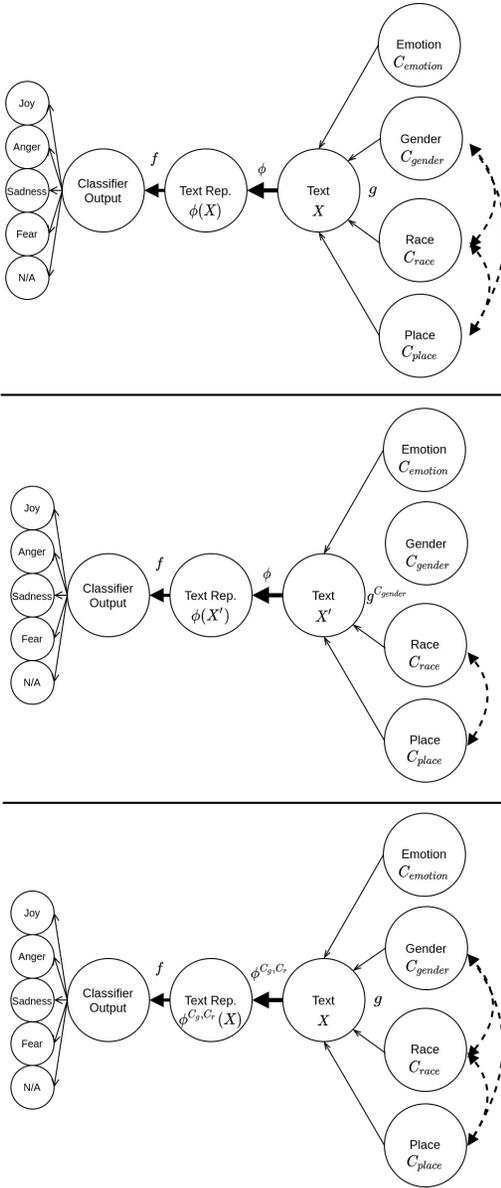
Figure 6
A causal graph for *Emotions*, *Gender*, *Race*, and *Place* generating a text with one of five mood states. The top graph represents a data-generating process where those concepts generate texts, with a potential hidden confounder affecting both C_{gender}, the *treated concept*, and C_{race}, the *control concept*. The middle graph represents the scenario where we can control the generative process and create a text without the *treated concept*. The bottom graph represents our approach, where we manipulate the text representation.

Gender as C_{gender}, our TC, and the potentially confounding concept is C_{race}, our CC. The *Race* task is generated similarly, by simply replacing *Gender* and *Race* in the causal graph.

As this data set is constructed using the templates described in Table 11, we can directly control each concept and create a true counterfactual example for each sentence.

For instance, we can take a sentence that describes a European male being angry, and replace his name (and the relevant pronouns) to a European female. Holding the *Race* and the rest of the sentence fixed, we can measure the true causal effect as the difference in a model's class distribution on the original European male example compared to that of the counterfactual, European female example.

Another advantage of experimenting with *Gender* and *Race* is that their effect, if it exists, is often undesirable. If we can use our method to create an unbiased textual representation with respect to the *treated concept*, then we can create better, more robust models using this representation. In Section 6.2 we discuss how to use our *BERT-CF* representation to mitigate such bias and create better performing models.

Controlling the Concept-Label Correlation. Using the EEEC data presented in Section 4, we create multiple versions of the data set, differing by the correlation between *Gender*/*Race* and the labels. For both *Gender* and *Race*, we create three versions of the data: *Balanced*, *Gentle*, and *Aggressive*. In the *Balanced* version, we randomly choose the person's name, resulting in almost no correlation between each label and the concept. In the *Gentle* version, we choose names such that 90% of examples from the *Joy* label are of female names, and 50% of the *Anger*, *Sadness*, and *Fear* examples are of male names. The *Aggressive* version is created similarly, but with 90% for *Joy* and 10% for the rest. For all these experiments we follow the same 64%, 16%, 20% split for the training, development, and test sets, respectively.

Modelling the TC and the CC. In the case of *Gender* and *Race*, in order to produce the *BERT-CF* model, the TC and CC are rather straightforward. For a given *TC*, for example, *Gender*, we define a binary classification task, where for each example the classifier predicts the gender described in the example. Equivalently, the *CC* task is also a binary classification task where, given that *Gender* is the *TC*, the classifier for *CC* predicts the *Race* described in the example.

5.4 Comparing Causal Estimates to the Ground Truth

While we do not usually have access to ground truth data (i.e., counterfactual examples), we can artificially generate such examples in some cases. For instance, in the *Gender* and *Race* cases we have created a data set where for each example we manually created an instance that is identical except that the concept is switched in the text. Specifically, we can switch the gender of the person being mentioned, holding everything else equal. For *Adjectives*, we followed a similar process of producing counterfactual examples, where *Adjectives* were removed from the original example's text. With these data sets we can then estimate the causal concept effect using our method, and compare this estimation to the ground truth effect, i.e., the difference in output class probabilities between actual test set examples and their manually created counterfactuals. Our ground-truth estimator of the causal effect is then an estimator of the ATE (Equation (3)):

$$ATE_{gt}(O) = \frac{1}{|I|} \left[\sum_{i \in I} \langle \vec{z}(f(\phi^O(x_{i,C_0=1}))) - \vec{z}(f(\phi^O(x_{i,C_0=0}))) \rangle \right] \quad (12)$$

where $x_{i,C_0=1}$ is an example where the concept C_0 takes the value of 1, and $x_{i,C_0=0}$ is the same example, except that C_0 takes the value of 0. For instance, if $x_{i,C_0=1}$ is: "*A woman is*

walking towards her son," $x_{i,C_0=0}$ will be: *"A man is walking towards his son."* Finally, $\bar{z}(\cdot)$ is the vector of output class probabilities assigned by the classifier when trained with ϕ^O, the representation of a vanilla, unmanipulated pre-trained BERT model (denoted with *BERT-O*, see below; to simplify our notation, we refer to this model simply as *O*).

Correlation-Based Baselines. We compare our methods to two correlation-based baselines, which do not take into account counterfactual representations and simply compute differences in predictions between test examples that contain the concept (i.e., $C_{TC} = 1$) and those that do not ($C_{TC} = 0$). The first baseline we consider is called *CONEXP*, and it was proposed by Goyal et al. (2019a) as an alternative for measuring the effect of a concept on models' predictions. *CONEXP* computes the conditional expectation of the prediction scores conditioned on whether or not the concept appears in the text. Importantly, this baseline is based on passive observations and is not based on *do*-operator style interventions. The corpus-based estimator of *CONEXP* is defined as follows:

$$\text{CONEXP}_{C_0}(O) = \left\langle \frac{1}{|I_{C_j=1}|} \sum_{i \in I_{C_j=1}} \bar{z}(f(\phi^O(x_i))) - \frac{1}{|I_{C_j=0}|} \sum_{i \in I_{C_j=0}} \bar{z}(f(\phi^O(x_i))) \right\rangle \quad (13)$$

The second baseline we consider is *TPR-GAP*, introduced in De-Arteaga et al. (2019) and used by Ravfogel et al. (2020). *TPR-GAP* computes the difference between the fraction of correct predictions when the concept exists in the text, and fraction of correct predictions when the concept does not exist in the text. It is computed using the following equation:

$$\text{TPR-GAP}_{C_0}(O) = \sum_{l \in L} |P(f(\phi^O(X)) = l | C_0 = 1, Y = l) - P(f(\phi^O(X)) = l | C_0 = 0, Y = l)|$$

$$(14)$$

where P is the share of accurate model predictions, and $f(\phi^O(X))$ and $l \in L$ denote the predicted class and the correct class, respectively.

Unlike *CONEXP*, *TPR-GAP* compares the accuracy of the model in two conditions, and not its class probability distribution, which prevents us from directly comparing it to the ground-truth $ATE_{gt}(O)$ or to our *TReATE*. As direct comparisons are not feasible, we discuss in Section 6 how the *TPR-GAP* captures the concept effect compared with our *TReATE*.

Language Representations. In our experiments, we consider three different language representations, which are then used in the computations of our *TReATE* causal effect estimator (Equation (8)), and the ground truth *ATE* (Equation (3)):

- *BERT-O* - The representation taken from a pre-trained BERT, without any manipulations.

- *BERT-MLM* - The representation from a BERT that was further fine-tuned on our data set.

- *BERT-CF* - The representation from BERT following our Stage 2 intervention (See Equation (9) and Figure 3).

Recall that our experiments are designed to compare the predictions of BERT-based classifiers. For each experiment on each task, we compare for each test set example the predictions of three trained classifiers, differing by the representations they use as input. To compute the estimator of the ground-truth causal effect, $ATE_{gt}(O)$, we compare the prediction of the *BERT-O* based model on the original example to its prediction on the counterfactual and average on the entire test set. Put it formally, we compute $ATE_{gt}(O)$ with Equation (12) where f is the *BERT-O*–based classification model. For our estimation of *TReATE*, we compare for each example the prediction of the *BERT-O*–based model on the original example to the prediction of the *BERT-CF*–based model on the same example.

As we want to directly evaluate the effect of our counterfactual training method, we also compute TReATE(O, MLM). This estimator is equivalent to Equation (10) except that the *BERT-CF*–based classifier is replaced with a classifier that is based on *BERT-MLM*: A representation model that is fine-tuned on the same data as *BERT-CF*, but using the standard *MLM* task instead of counterfactual training. Explicitly, we compute *TReATE* using the following equation:

$$\text{TReATE}(O, CF) = \frac{1}{|I|} \left[\sum_{i \in I} \langle \vec{z}(f(\phi^O(x_i))) - \vec{z}(f(\phi^{CF}(x_i))) \rangle \right] \qquad (15)$$

TReATE(O, MLM) is computed using the same equation where ϕ^{CF} is replaced with ϕ^{MLM}.

6. Results

Examining and analyzing our results, we wish to address the four research questions posed in Section 5. That is, we assess whether our method can accurately estimate the *ATE* when such ground truth exists (question 1), whether our *BERT-CF* forgets the *treated concept* and remembers the *control concept* (questions 2 and 3, respectively) and whether we can mitigate bias using the *BERT-CF* (question 4). Finally, we dive into the training process, and discuss the effect of our Stage 2 intervention on BERT's loss function.

6.1 Estimating TReATE (The Causal Effect)

Comparing TReATE and the Ground Truth ATE. Our estimated TReATE(O, CF) for each of the three concepts we have ground truth data for (*Adjectives*, *Gender*, and *Race*), compared to the ground truth (ATE$_{gt}(O)$) and the CONEXP(O) baseline, are described in Tables 5 and 6.[19] In the *Gender* and *Race* experiments, we also compare our results to those obtained by the Iterative Nullspace Projection (INLP) method (Ravfogel et al. 2020). This method removes information from neural representations, but it does not preserve the information in control concepts. We let INLP remove information about the treated concept from the *BERT-O*'s representation, and compute the TReATE($O, INLP$) using their default classification algorithm.[20]

---

19 We have also computed results for CONEXP(*MLM*), TReATE(*MLM, CF*), and ATE$_{gt}(MLM)$, but do not discuss them here as they are very similar and therefore do not add insight to this discussion.

20 We utilize the original code from the author's GitHub repository, with its default hyperparameters: https://github.com/shauli-ravfogel/nullspace_projection.

Table 5
Results for the causal effect of *Adjectives* on sentiment classification on Reviews. We compare
TReATE(O, CF) to the ground truth ATE$_{gt}(O)$ and the baseline CONEXP(O). Confidence
intervals ([CI]), computed using the standard deviations of ITE$_{gt}(O)$, TReITE(O, CF), and
CONEXP, are provided in square brackets.

Experiment	ATE$_{gt}(O)$	TReATE(O, CF)	CONEXP(O)
Balanced	0.397	0.385	0.01
[CI]	$[0.377, 0.417]$	$[0.381, 0.389]$	$[0, 0.044]$
Gentle	0.376	0.351	0.094
[CI]	$[0.361, 0.392]$	$[0.347, 0.355]$	$[0.061, 0.127]$
Aggressive	0.634	0.603	0.126
[CI]	$[0.613, 0.655]$	$[0.588, 0.618]$	$[0.095, 0.158]$

As demonstrated in the tables, we can successfully estimate the ATE$_{gt}(O)$ using our
proposed TReATE(O, CF): The values of TReATE(O, CF) and ATE$_{gt}(O)$ are very similar
across all experiments. Regardless of the amount of bias introduced in the experiments
(*Balanced*, *Gentle*, and *Aggressive*), our method can estimate the causal effect successfully.
Comparatively, the non-causal baseline CONEXP(O) substantially underestimates the
concepts' effect in 7 out of 9 experiments. In the other two experiments, the *Balanced*
and *Gentle Race* experiments, it overestimates the effect. Estimating TReATE(O, CF) with
INLP (referred to as TReATE$(O, INLP)$ above and as INLP in the table) substantially
overestimates the ATE$_{gt}(O)$, possibly because INLP does not preserve the information
encoded about control concepts.

In the *Adjectives* experiments (Table 5) we see that the effect of *Adjectives* on sen-
timent classification is prominent even in the *Balanced* setting, suggesting that *Adjec-
tives* change the classifier's output class probability distribution by 0.397 on average.
While the bias introduced in the *Gentle* setting did not affect the degree to which
the classifier relies on *Adjectives* in its predictions (ATE$_{gt}(O) = 0.397$ in the *Balanced*
case and ATE$_{gt}(O) = 0.376$ in the *Gentle* case), it certainly did in the *Aggressive* setting
(ATE$_{gt}(O) = 0.634$ in the *Aggressive* case). Interestingly, the effect of *Adjectives* on the
classifier is similar in the *Balanced* and *Gentle* settings, suggesting that the model was
not fooled by the weak correlation between the number of *Adjectives* and the positive
label. When this correlation is increased, as was done in the *Aggressive* setting, the effect
increases by 60% (from ATE$_{gt}(O) = 0.397$ in the *Balanced* case to ATE$_{gt}(O) = 0.634$ in
the *Aggressive* case).

In all three settings of the *Adjectives* experiments, our TReATE(O, CF) estimator is
very similar to the ATE$_{gt}(O)$, and the gap between the two remains at 3% (absolute).
Similar patterns can be observed in the *Gender* and *Race* experiments (Table 6). For
both the *Gender* and *Race* concepts, we successfully approximate the ATE$_{gt}(O)$ with our
TReATE(O, CF) with a maximal error of 3.9% (absolute) and an average error of 2.6%
(absolute). Similar to our observation in the *Adjectives* case, in the *Gender* and *Race* cases
the effect of the *Gentle* bias on the extent to which the classifier relies on the *treated
concept* is very small. For both *Gender* and *Race*, the effect in the *Gentle* setting is only
slightly higher than that observed in the *Balanced* setting (1% and 1.3% absolute increase
in ATE$_{gt}(O)$).

Another interesting pattern that emerges is that the effect of *Gender* on the POMS
classifier in the *Balanced* setting is 0.086, more than six times higher than the 0.014
observed in the equivalent *Race* experiment. In our EEEC data set, the *Balanced* setting is

Table 6
Results for the effect of *Gender* (top) and *Race* (bottom) on POMS classification with the EEEC data set. We compare TReATE(O, CF) (the TReATE column) to the ground truth ATE$_{gt}$(O) and the baseline CONEXP(O). We also compare to INLP: A TReATE(O, $INLP$) estimator where TReATE(O, CF) is estimated with INLP instead of our BERT-CF. Confidence intervals ([CI]), computed using the standard deviations of ITE$_{gt}$(O), TReITE(O, CF), CONEXP, and TReATE(O, $INLP$), are provided in square brackets.

TC	Experiment	ATE_{gt}	TReATE	CONEXP	INLP
Gender	Balanced	0.086	0.125	0.02	0.313
	[*CI*]	[0.082, 0.09]	[0.110, 0.14]	[0.0, 0.05]	[0.304, 0.321]
	Gentle	0.113	0.135	0.076	0.331
	[*CI*]	[0.108, 0.118]	[0.12, 0.15]	[0.072, 0.08]	[0.322, 0.340]
	Aggressive	0.210	0.241	0.057	0.437
	[*CI*]	[0.203, 0.217]	[0.229, 0.253]	[0.051, 0.063]	[0.427, 0.448]
Race	Balanced	0.014	0.046	0.08	0.591
	[*CI*]	[0.012, 0.016]	[0.038, 0.054]	[0.02, 0.014]	[0.578, 0.605]
	Gentle	0.027	0.04	0.044	0.621
	[*CI*]	[0.024, 0.03]	[0.048, 0.032]	[0.028, 0.07]	[0.607, 0.635]
	Aggressive	0.345	0.332	0.19	0.650
	[*CI*]	[0.333, 0.357]	[0.324, 0.34]	[0.08, 0.3]	[0.636, 0.664]

designed such that there is no correlation between the *Gender* or the *Race* of the person and the label. The fact that such causal effect is observed suggests that *BERT-O* contains *Gender*-related information that affects classification decisions on downstream tasks.

To conclude, comparing TReATE(O, CF) and ATE$_{gt}$(O) on all experiments where we have counterfactual examples, we conclude that we can successfully estimate the causal effect, answering question 1 presented in Section 5. Regardless of the bias introduced and the extent that it affects the classifier, our TReATE(O, CF) estimator remains close to the ATE$_{gt}$(O). It can successfully detect bias when it exists and performs well even when the true effect is close to 0 such as in the *Balanced Race* experiment. Comparatively, the CONEXP(O) baseline is not able to approximate the true causal effect in any of the experiments we conduct here.

Although we cannot directly compare the TPR-GAP(O) baseline to the TReATE (O, CF), ATE$_{gt}$(O), and CONEXP(O) estimators (Section 5.4), we can still analyze the values of this non-causal estimator on each of the three versions for each concept. As seen in Table 7, the TPR-GAP(O) values are very small, with an average of 0.025. While the results shown in Tables 5 and 6 suggest that the true effect (ATE$_{gt}$(O)) increases along with the bias that was introduced to the data (with the exception of the *Balanced* and *Gentle Adjectives* experiments), it is not the case with TPR-GAP(O). The estimated effect in the *Gentle* experiments is the highest for both *Adjectives* and *Gender* (0.074 and 0.014, respectively), and is lowest for the *Aggressive* experiments (0.012 and 0.003, respectively). Only in the *Race* experiments do we see a pattern that is similar to that observed in the TReATE(O, CF) and the ATE$_{gt}$(O), but the scale is different, with 2- and 12-fold increases in the ATE$_{gt}$(O) (0.014 to 0.027 and then to 0.345), compared with 6- and 4-fold increases in the TPR-GAP(O) (0.002 to 0.012 and then to 0.049).

Understanding TReATE Without Ground Truth. Unlike the *Adjectives*, *Gender*, and *Race* experiments, we do not have counterfactual examples for *Topics* and therefore cannot

Table 7
The *TPR-GAP* results for all versions (*Balanced*, *Gentle*, and *Aggressive*) for the three concepts where we have ground truth (*Adjectives*, *Gender*, and *Race*).

Experiment	Adjectives	Gender	Race
Balanced	0.057	0.003	0.002
Gentle	0.074	0.014	0.012
Aggressive	0.012	0.003	0.049

Table 8
Results for the effect of the $t_{TC}(books)$ (top) and $t_{TC}(movies)$ (bottom) *Topics* on sentiment classification on product and movie reviews. As we do not have access to the ground truth, we compare TReATE(O, CF) (denoted in the table as TReATE$_{domain}$) on each domain separately, and denote b, d, e, k, m for each of the domains: *Books*, *DVD*, *Electronics*, *Kitchen Appliances*, and *Movies*, respectively. The domain for which the effect of the *treated concept* is the highest for each experiment is highlighted in bold.

Domain	Experiment	TReATE$_b$	TReATE$_d$	TReATE$_e$	TReATE$_k$	TReATE$_m$
Books	Balanced	**0.131**	0.113	0.034	0.085	0.113
	Gentle	**0.207**	0.191	0.155	0.156	0.178
	Aggressive	**0.656**	0.176	0.154	0.137	0.181
Movies	Balanced	0.185	0.185	0.140	0.147	**0.207**
	Gentle	0.204	0.235	0.204	0.207	**0.260**
	Aggressive	0.315	0.492	0.272	0.267	**0.605**

compare our estimates to the ground truth. Alternatively, we provide here several sanity checks that suggest that the causal effect can be estimated for *Topics* as well. With *Topics* as our concepts we conduct two rounds of experiments, presented in Table 8, where in the first we choose $t_{TC}(books)$ and in the second we choose $t_{TC}(movies)$ as our *treated concepts* (the *control concepts* are chosen for each *TC* as described in Section 5). For each, we train the models on the combined data set and test on each of the five domains (*Books*, *DVD*, *Electronics*, *Kitchen Appliances*, and *Movies*) separately. Observing the results, it appears that the effect of $t_{TC}(domain)$ is highest on the domain most correlated with it, suggesting that the adversarial training used in our Stage 2 intervention did learn to forget the *TC Topic*. Another interesting pattern is that the estimated effect is higher in more similar domains, and lower on those that are less similar. Specifically, the effect of $t_{TC}(movies)$ is highest on the *Movies* and *DVD* domains, and lowest on the *Electronics* and *Kitchen Appliances* domains. The same pattern can be observed with $t_{TC}(books)$, where the effect is higher on *DVD* and *Movies*, and lower on *Kitchen Appliances* and *Electronics*.

To test the effect of controlling for multiple concepts, we take advantage of the multi-concepts setting available in the *Topics* data set. We focus on the *Movies* domain and train *BERT-CF* to forget $t_{TC}(movies)$, while controlling for additional concepts. We perform this experiment with two control concepts, choosing the additional *control concept* in the same manner we chose the first control (i.e., the second control concept is the third most correlated concept with the movies domain). As can be seen in Table 9, adding control for an additional confounding *Topic* does not substantially affect the TReATE$_m$ estimator. As expected, TReATE$_m$ does decrease as we add more controls, suggesting that some of the effect captured in the single control experiments is due to confounding information.

Table 9
Results for the effect of the t_{TC}(*movies*) *Topic* on sentiment classification of movie reviews, with one and two *control concepts*.

Experiment	# Control Concepts	
	1	2
Balanced	0.207	0.190
Gentle	0.260	0.247
Aggressive	0.605	0.607

6.2 Analyzing the Counterfactual Model

Apart from testing the ability of our method to accurately estimate the $\text{ATE}_{gt}(O)$, we want to test the effect of our Stage 2 intervention on the resulting task classifier. Specifically, we look at three aspects, corresponding to questions 2–4 posed in Section 5. First, we test the accuracy classifiers that utilize the different representation models in predicting the *treated concept* that was adversarially removed in Stage 2 of *BERT-CF*, to check whether we have successfully deleted the concept-related information (question 2). Second, we look at the accuracy of models trying to predict the control concepts, to test that we did not delete information regarding other concepts (question 3). Finally, we look at the performance of models trained in the *Aggressive* setting on a *Balanced* test set, to test whether we can debias the models using the counterfactual representation and therefore improve the classification accuracy (question 4).

Detecting the Treated Concepts. To show that we have successfully trained our *BERT-CF* representation model to forget the *treated concept* (question 2), we compare the accuracy of TC classifiers that use the *BERT-O* representation, the *BERT-MLM* representation, or the treated *BERT-CF* representation. As can be seen in Figure 7, the performance of the TC classifiers is very high when using the *BERT-O* or *BERT-MLM* representations, with some achieving almost 100% test-set accuracy. When using the treated representations, however, it is clear that there is a substantial degradation in performance, suggesting that some relevant information was lost. Specifically, in the case of *Gender* and *Adjectives*, and to a lesser extent *Race* and *Topics*, the performance of the *BERT-CF*–based TC classifier is only slightly higher than chance.

An important caveat of this analysis is that it does not directly measure the information preserved in the language representation. As discussed in Section 2, structural interpretation methods such as those presented here only measure a model's ability to use the representation (in our case, the ability of the TC classifier to use the information encoded in the representation), and not the actual way in which information is encoded in the representation. An analysis of the type presented here should be used as a sanity check, where ground truth data is not available and we want to know whether our counterfactual representation model has forgotten some information about the treated concept.

Detecting the Control Concepts. Although we have shown that *BERT-CF* has forgotten some information regarding the *treated concept*, it could be at the expense of other concepts correlated with it. So, another useful test for *BERT-CF* is to check if the accuracy of a *BERT-CF*–based classifier trained to predict the *control concepts* is the same as the
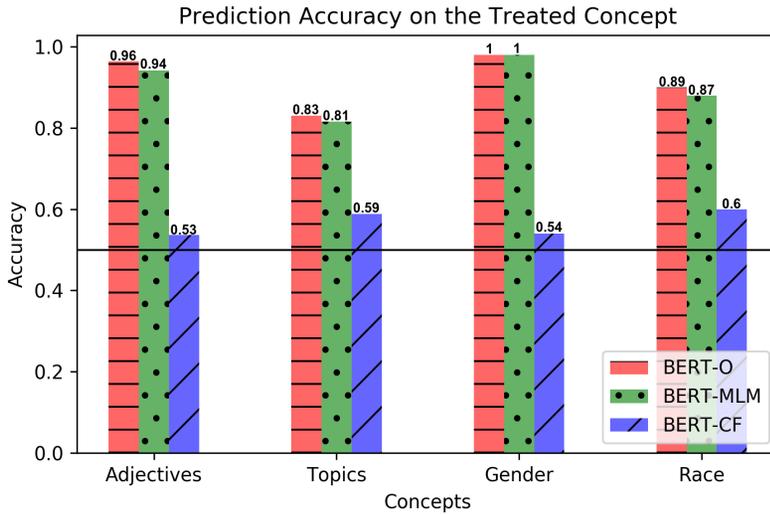
Figure 7
Prediction accuracy on the *treated concept*, averaged over all three data set versions (*Balanced*, *Gentle*, and *Aggressive*), for each concept. For each of the four concepts, we report prediction accuracy on the *treated concept* for classifiers based on *BERT-O*, *BERT-MLM* or *BERT-CF* representations.

accuracy of corresponding classifiers trained with *BERT-O* or *BERT-MLM* representations. As shown in Figure 8, using the representations from the adversarially trained *BERT-CF* does not hurt performance on related, potentially confounding *control concepts*, showing that we have also successfully answered question 3. In all experimental setups and for all treated concepts, we observe that the difference in performance when using the *BERT-O*, *BERT-MLM*, or *BERT-CF* representations is very small. Indeed, using the treated representation degrades performance by only 2% to 10% (absolute) in terms of accuracy, compared to using the *BERT-O* representation.

Although these results support the claim that these specific confounders were not affected by our Stage 2 intervention, it could very well be that others were affected. This analysis should guide researchers trying to estimate causal effects, and can be used to refute hypotheses regarding specific confounders. In our *Gender* and *Topics* experiments we have created our data sets such that they do not have additional confounders, but for the *Adjectives* and *Topics* experiments there might well be other confounders apart from those tested here.

Mitigating Bias. The claim in the literature that models pick correlations observed in the training set and are easily biased (Tshitoyan et al. 2019; Gonen and Goldberg 2019) is supported in our experiments. Specifically, looking at the $\text{ATE}_{gt}(O)$ on all experiments with our *Aggressive* setting, it seems that models learned to associate the *treated concept* with the labels. An advantage of our method is that it generates an unbiased language representation with respect to some concept of interest, which can be useful for mitigating such bias.

To test whether we can indeed mitigate that bias (question 4), we test all models trained in the *Aggressive* setting on a *Balanced* test set. Through these experiments, we can test the generalization of the *BERT-O*, *BERT-MLM*, and *BERT-CF*–based models.
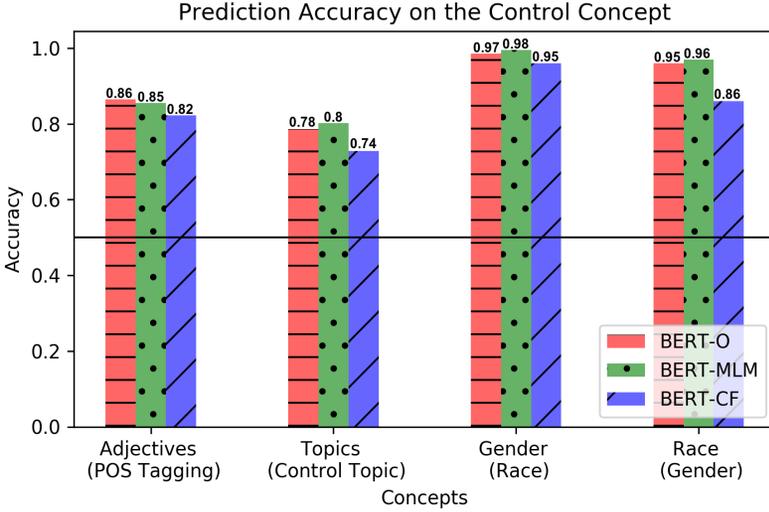
Figure 8
Prediction accuracy on the *control concept*, averaged over all three data set versions (*Balanced*, *Gentle*, and *Aggressive*), for each *control concept*. For each of the four concepts, we report prediction accuracy on the task in parentheses for classifiers based on *BERT-O*, *BERT-MLM*, or *BERT-CF* representations.

Looking at Table 10, it is clear that the *BERT-CF*–based models can generalize better, and outperform both *BERT-O* and *BERT-MLM*–based models when the correlations picked up in training do not exist in the test set. Comparatively, *BERT-CF* is not as affected by the distribution shift, and performs well when the correlation between the *treated concept* and the label changes.

6.3 Analyzing the Stage 2 Multitask Training Scheme

In order to gain further insight into our proposed intervention method, carried out during Stage 2 of the *BERT-CF* training, we present the following loss analysis. We are particularly interested in analyzing the effects of adding *TC* and *CC* tasks to the Stage 2 training scheme, on the *MLM* task and the overall resulting loss function optimization dynamics. While we have shown in Section 6.2 that *BERT-CF* successfully forgets the *TC* task and remembers the *CC* task (questions 2 and 3, respectively), this could be at the expense of the language model. To test if the language model was affected by our Stage 2 intervention, we compare the training losses between *BERT-MLM* and *BERT-CF* without the *CC* task for *Gender* (Figure 9), *Adjectives*, and *Topics* treatments. For *Adjectives* (Figure 10) and *Topics* (Figure 11), we also compare these models to *BERT-CF* with the *CC* task. We do not present figures for *Race* because they are almost identical to the figures for *Gender*.

   We executed each Stage 2 training process for a total of 5 epochs, for all variants and all treatments. Our figures present trend lines that are smoothed for visual convenience purposes, by aggregating over batches of 1,000 training steps rather than over entire epochs. In Figure 12, we present the standard deviation of the loss values within each 1,000 training steps only for *Adjectives*, since it best depicts the phenomena occurring for *Gender*, *Race*, and *Topics* as well.

Table 10
Accuracy of the *BERT-O*, *BERT-MLM*, and *BERT-CF*–based classifiers when trained in the *Aggressive* settings and tested on the *Balanced* test set. For each concept the model with the highest accuracy is highlighted in bold.

Concept	Task	*BERT-O*	*BERT-MLM*	*BERT-CF*
Adjectives	Sentiment	0.750	0.744	**0.793**
Topics	Sentiment	0.584	0.564	**0.742**
Gender	POMS	0.924	0.918	**0.971**
Race	POMS	0.922	0.919	**0.970**
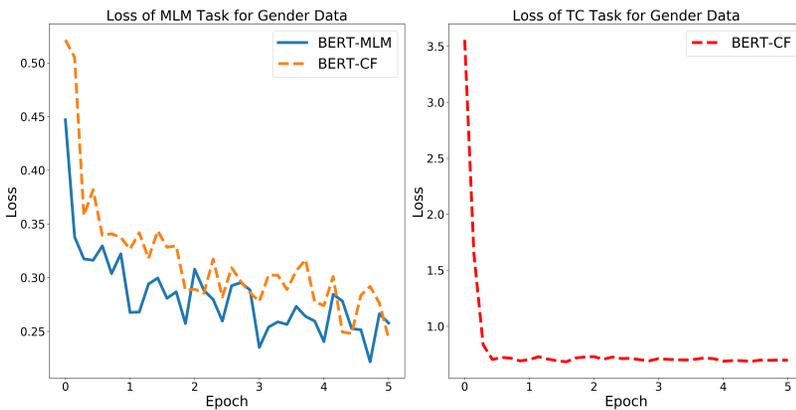
Figure 9
Mean MLM (left) and TC (right) losses (each point is the average of 1,000 training steps) for Stage 2 of the *Gender* treatment. *BERT-MLM* refers to the model variant that is trained only on the *MLM* task. *BERT-CF* refers to the model variant that employs both *MLM* and *TC* tasks.

The first, most apparent observation we see in all variants and all treatments is that the *TC* tasks typically converge after 1–2 epochs. The addition of a *CC* task causes a loss increase during the first epoch of Stage 2 training for all *TC* tasks but quickly converges, typically after 1 epoch. The *TC* tasks also typically converge to the lowest standard deviation, compared to the *MLM* and *CC* tasks.

These observations suggest that the addition of a *TC* task introduces an immediate "disturbance" to the BERT encoder, which is expected since the task's goal is to cause the encoder to "forget" features associated with a specific concept. It could be further explained by the dynamics resulting from the adversarial component such *TC* tasks utilize when training alongside "standard" tasks. It is encouraging to see that despite the risk of harming the encoder's representations by adding an adversarial task to the training scheme, this task has an apparent effect without destabilizing all losses that also converge quickly.

For the *Adjectives* treatment, the addition of a *CC* task has a visible effect on the *TC* task, but not on the *MLM* task. The *TC* loss spikes higher at an earlier stage of training, and converges later, in comparison to the *TC* only model variant. Still, the general behavior of the *TC* and *MLM* losses remains very similar in both variants. This suggests that adding the *CC* task dampens the adversarial effect on the *MLM* task, as introduced by the *TC* task, without overwhelming it.
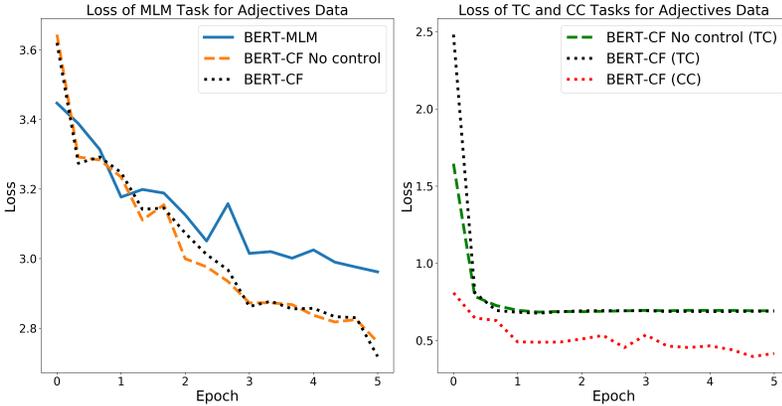
Figure 10
Mean MLM (left) and TC/CC (right) losses (each point is the average of 1,000 training steps) for Stage 2 of the *Adjectives* treatment. *BERT-MLM* refers to the model variant that is trained only on the *MLM* task. *BERT-CF* refers to the model variant that uses *MLM* alongside both *TC* and *CC* tasks and *BERT-CF no control* refers to the same model without the *CC* task.
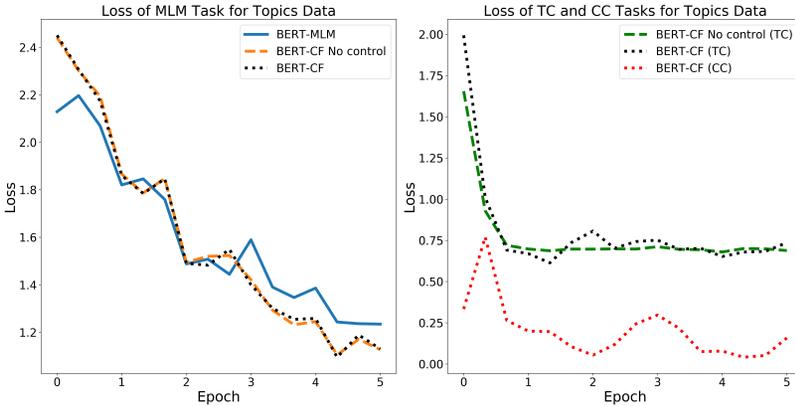

Figure 11
Mean MLM (left) and TC/CC (right) losses (each point is the average of 1,000 training steps) for tasks in Stage 2 of the *Topics* treatment. *BERT-MLM* refers to the model variant that is trained only on the *MLM* task. *BERT-CF* refers to the model variant that uses *MLM* alongside both *TC* and *CC* tasks and *BERT-CF no control* refers to the same model without the *CC* task.

Indeed, it seems that the *CC* task acts as an "opposing" force to the adversarial *TC* task, with the goal of "preserving" features related to a specific concept, while *TC*'s goal is "forgetting" features related to a different concept. We can also see that, generally, the *CC* task loss is the quickest to converge in comparison to *TC* and *MLM* tasks. This makes sense, since this task (*PoS Tagging*) is fairly similar to the *TC* task (*IMA*), yet has no adversarial component, and is commonly considered a simpler task compared to *MLM*.

When examining the *MLM* losses across different model variants, we see that all *MLM* mean losses exhibit similar behavior regardless of the variant they come from. While there is an increase in standard deviation in variants that introduce additional tasks, the *MLM* loss is lower in *BERT-CF* compared with *BERT-MLM* in the *Adjectives* and *Topics* experiments, and is only slightly higher in the *Gender* experiments. This
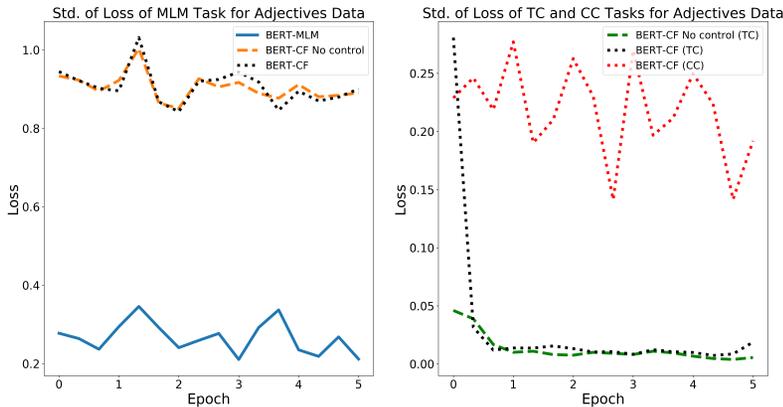
Figure 12
Standard deviation of losses (each point is the average of 1,000 training steps) for tasks in Stage 2 of the *Adjectives* treatment. *BERT-MLM* refers to the model variant that is trained only on the *MLM* task. *BERT-CF* refers to the model variant that employs *MLM* alongside both *TC* and *CC* tasks and *BERT-CF no control* refers to the same model without the *CC* task.

suggests that adding well-defined *TC* and *CC* tasks to the Stage 2 training scheme does not drastically harm its stability or the resulting BERT representations.

To summarize, this analysis shows that the addition of the adversarial component in our Stage 2 intervention does not harm the *MLM* or the underlying language representation. Moreover, we have shown that the introduction of the *CC* tasks does not affect the *MLM* as well. While in our analysis in Section 6.2 we have shown that task-trained classifiers using *BERT-CF* can perform well, it could have been due to factors other than the language representation. In the analysis provided here, we have shown that the *MLM* was not harmed by our intervention method, supporting the claim that the language representation remained informative.

7. Discussion and Conclusion

Our main contributions in this article are in five directions. First, we have introduced a causal approach for evaluating a variety of hypotheses regarding the effect of a concept on a DNN classifier. Our approach is based on modeling the data-generating process with a causal graph that explicitly states the potential confounding effects between the involved variables. Second, reasoning that direct counterfactual example generation is infeasible with current NLP technology, we have proposed a method for the generation of counterfactual representations, thus avoiding the need for text generation. In causal inference terminology, our method implements the *do*-operator by adversarially training the language representation. Third, we have created four data sets, each with three variants, where for three of them the true causal effect of a concept can be estimated using manually generated counterfactual examples included in the data set. Fourth, we have provided tools for evaluating counterfactual language representation models like our *BERT-CF*, in the realistic setup where ground truth causal effect estimations are not available. Finally, we have demonstrated that our counterfactual language representation approach is effective for model debiasing.

Our approach requires making explicit assumptions about the world and generating hypotheses regarding the concepts driving the models' decisions. In order to estimate the impact of a given concept on a DNN, a world model, referred to as a *causal*

graph, should first be designed. This causal graph depicts the concepts that generate the text that is fed to the DNN, and presents the relations between them. For each concept whose effect we estimated, we have hypothesized what a graph describing the data-generating process might look like, and have approximated its effect relying on this model of the world. In doing so, we have given the modeler a mechanism for explicitly stating her assumptions on the world and the data she is using. While these assumptions are always approximations and are bound to focus on a small number of variables, the alternative is not assumption-free. Indeed, whenever we wish to interpret a model, we are making assumptions on the data-generating process and on the world. Without controlling for confounders, we might end up estimating the effect of variables that are correlated with our *treated concept*. Existing interpretation methods do not explicitly assume a world model, like we do with our causal graphs, but the variables and correlations are still there.

Choosing the *control concepts* is therefore crucial for estimating the true effect and not that of the confounder. Of course, different *control concepts* will probably yield different estimations, and affect the decisions that rely on those interpretations. Using the sanity checks we provide in Section 6, we have shown that it is possible to test if we have controlled for a given *control concept*. However, it is crucial to identify the treatment and control concepts and form a world model, such as those presented in the causal graphs in Sections 3 and 5, in order to think through the structure of the task and properly design a CausaLM (or another causal) explanation model.

In Appendix B, we present two cases where such world models induce causal graphs with parent concepts that cause other concepts. In such cases, intervening on some concepts will induce a change in concepts that are caused by them, resulting in an estimation that computes both the direct effect of the concept and its effect on concepts that it causes. In areas such as medicine, the causal graph can be built with the assistance of a doctor that is aware of potential confounders and their relationship, but in NLP this is more of a challenge. In Section 6.2 we have proposed several sanity checks that can help modelers understand if their intervention was successful, but in some cases an intervention on a specific concept is not possible.

An important assumption that our world model makes is that concepts can either be switched on or off. Consequently, our representation-based model does not allow us to change the value of a concept (e.g., changing the gender of an example from female to male). Our *TReATE* estimator hence measures the difference between the output class distribution of the classifier in the case where the representation encodes a given concept and the case where the representation does not encode that concept. This, however, is not always in line with the real world. For example, in the *Adjectives* data sets we have created ground truth counterfactual examples where there are no *Adjectives* at all, which is in line with what our *TReATE* estimator measures. However, in cases such as *Gender* and *Race*, for a given example that discusses a person, we are interested in the counterfactual where the *Gender* or *Race* of the person is switched (e.g., from male to female) while deleting the concept is not feasible (e.g., gender is encoded in English sentences through words like *him, her, he*, and *she*). Our results in Section 6 suggest that we can still estimate the causal effect of the *Gender* and *Race* concepts on the classifier with *TReATE*, although it compares the output class distribution of the model between representations that encode information about the concept and representations that omit this information. Despite the empirical success, this is a limitation of our framework that we plan to address in future work.

The discussion above emphasizes the importance of world knowledge and assumptions for interpreting DNNs. As long as assumptions have to be made on the connection

between concepts or features, it is crucial that modelers make these assumptions explicitly, and not just implicitly. Although these assumptions might be wrong, they can ground the discussion and allow for empirical tests to be made. In the experiments we conducted, we have tried to address many different types of assumptions on what is driving the data-generating process.

Another issue we have discussed is the distinction between global and local concepts. Language expresses many global concepts such as the topics being discussed and the style being used, but these are very hard to model. We proposed an elegant solution for *Topics* that is both global (shares information between the training set sentences) and can be integrated into our counterfactual representation model. Unlike local concepts such as *Gender* and *Race*, which are widely researched, there is very little work on understanding the effect of global concepts on DNNs. As they are hard to measure and cannot be computed through a token-level analysis, they remain an open challenge. Understanding the effect of global concepts is a direction that we wish to further explore in future work.

Finally, there is a significant challenge in validating the quality of a causal explanation method. One method, which we have used in our *Gender* and *Race* experiments, is to use synthetic data, where the validation is more accurate and reliable. However, this comes at the expense of not using real-world data, which is more natural and complex. When using real-world data to validate causal methods, we often need to generate counterfactual examples manually. While in the *Adjectives* experiments we were able to create such examples without manual interventions, it is almost always hard to do. For example, it would be almost impossible to create a counterfactual example with respect to a *Topic* without manually generating a new example.

In this work we have created four data sets, two synthetic and two real-world, that allow for such causal validation. However, the synthetic data set (EEEC), is limited in terms of the language it expresses, and the real-world data set (Sentiment) only has automatically generated counterfactual examples, which can be inaccurate. We see the creation of a data set that is both natural and includes precise counterfactual examples as crucial for the advancement and dissemination of causal model explanations in NLP, and plan to further develop such data sets.

Appendix A. A Clinical Example

While failing to estimate the causal effect of a concept on a sentiment classifier is harmful, it pales in comparison to the potential harm of wrongfully interpreting clinical prediction models. If a model is trained on clinical notes to predict clinically important factors, the need for understanding the model is amplified. If it were the case that the model is relying on textual features that are doctor- or hospital-specific, it could lead to devastating implications.

For example, we can look at the (fake) clinical note presented in Figure 13. In this note, the patient's mental health is discussed extensively, with a verbose description and much detail. As the description is lengthy and sometimes repetitive, it could be summarized without losing too much clinically relevant information. In this case, a classifier that heavily relies on the doctor's verbose style could fail when given a short and concise note that still contains the same clinical information.

In this note, we highlight words by their length, a feature described previously as a proxy for writing style (Sari, Stevenson, and Vlachos 2018). Looking at the words highlighted in red and in orange, it is clear that changing the writing style of the note would require a significant intervention. At the same time, deleting long words or re-

Status of patient:

Julie **is** worse today.

Target Symptoms:

Julie reports **that depressive** symptoms **continue. Her symptoms, she** reports, **are more** frequent **or more** intense. Anergia **is** present. **Increased** symptoms **of anhedonia are** present. Julie's **difficulty with concentrating has not** changed. Julie reports **that she continues to feel sad.** Guilty feelings **are described by** Julie. **'I** should **have been with my** sister, **I had no idea she was suicidal.'** Sleep **has** improved **with the use of PRN** Ambien **CR at HS.** Julie **convincingly** denies suicidal ideas **or intentions.**

Basic **Behaviors:**

Medication has been taken **regularly. She** needs **help with** ADLs. **When she** attends **activities participation is** minimal. Prn's **are used occasionally and are described as effective for her headaches. Impulsive behaviors are occurring, but less frequently.** Julie **has diminished food and** fluid intake. Julie **has not been confused. A good** night's sleep **is described.**

Additional Signs **or** Possible **Side** Effects:

Sedative effects **of the medication are described.** Patient reports **a dry** mouth. **No** other **side** effects **are** reported **or in evidence.**

MENTAL STATUS:

Julie presents **as** glum, **downcast, inattentive, minimally communicative, and** looks unhappy. **She** appears listless **and** anergic. **She** appears **downcast.** Thought content **is depressed.** Slowness **of** physical movement helps reveal **depressed** mood. Facial **expression and** general demeanor reveal **depressed** mood. **She** denies having suicidal ideas. There **are no** apparent signs **of hallucinations, delusions,** bizarre **behaviors, or** other **indicators of psychotic** process. **Associations are** intact, thinking **is** logical, **and** thought content appears **appropriate.** There **are** signs **of** anxiety. Patient **is** fidgety **in a way that is suggestive of anxiety.**

Special **Circumstances:**

Julie **continues to have an** unsteady gait, **especially** after **midnight. Call** light **is** within **her** reach. **She has been instructed to ring for the** nurse **to** assist **her when ambulating to bathroom.**

Vital Signs:

Sitting blood pressure **is 150 / 85.** Sitting pulse **rate is 80. Respiratory rate is 18 per** minute. Temp. **is 98+ F.** Weight **is 155 lbs.** (70.3 **Kg**).

Figure 13

A fake example of a *Nursing Progress Note* taken from `https://www.examples.com/business /progress-note.html`. Highlighted in **red** and **orange** are words with length in the 90–100th and 75–90th quantiles, respectively. **Green** words are of length that is below the 25th quantile. Quantiles are measured based on the frequency of all words in the clinical note.

placing them would significantly affect the structure and content of the note. Moreover, different note sections tend to have words with different lengths (for example, the *Vital Signs* section contains very short words), which could be a potential confounder. If we intervene and replace longer words with shorter synonyms, we might also change some
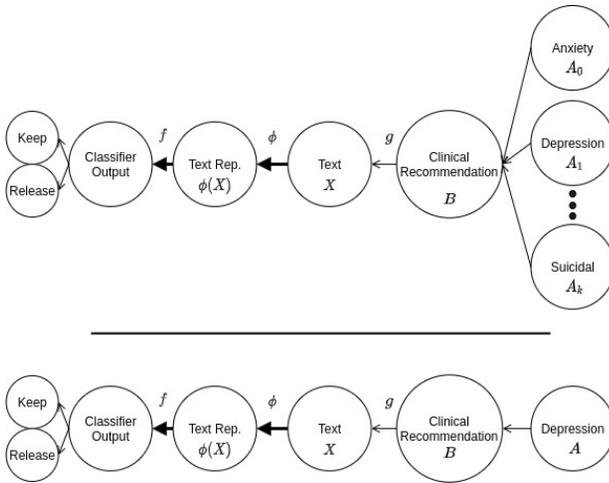
Figure 14
Two plausible causal graphs for a case where a patient's condition (*Anxiety*, *Depression*, and *Suicidal*) is causing the doctor's *Clinical Recommendation*, which is then generating a text. The classifier then uses this text to decide if the patient should be kept in the hospital (*Keep*) or released (*Release*). The top graph represents a data-generating process where many conditions cause the doctor's *Clinical Recommendation*, such as *Anxiety*, *Depression*, and *Suicidal*. The bottom graph represents the scenario where only *Depression* causes the *Clinical Recommendation*.

concepts alongside the ones we mean to change, and there is no test that will tell us that ex ante.

Concepts that influence both the label and other concepts, also known as *confounders*, could be extremely risky. Imagine a case where a doctor receives on average more patients that are of a certain type, such as individuals with severe depression. In that case, a DNN could learn to associate this writing style as a signal for a depressed patient. Measuring the model's performance on notes written by that doctor would show promising results, but deploying such a model would risk patient health when used on patients of a different doctor. Without measuring the causal effect of the doctor's writing style on the classifier, we would not be able to tell to what extent the model is relying on it.[21]

Other, more complex relationships might exist between concepts. For example, if a clinical note is describing the doctor's clinical treatment suggestion based on the patient's condition (depression, anxiety, etc.), it would be hard to disentangle the clinical treatment suggestion from a specific condition (the causal graph for this example is presented in the bottom graph in Figure 14). Alternatively, it could also be that only the patient's depression or lack of it is causing the doctor's treatment suggestion, and the text is generated based on that suggestion alone (see the bottom graph in Figure 14). This would make it impossible to imagine a counterfactual text, where the upstream concept (depression) is changed but the one generated by it (treatment suggestion) remains fixed. We next (Appendix B) discuss alternative causal graphs that can be modeled, and highlight the power and limitations of using a world model such as those discussed here to interpret DNNs.

---

21 Note that while clinical notes are an important application domain, we do not consider them in our experiments as they were not publicly available to us. We plan to create such synthetic data in future work.

Appendix B. Alternative Causal Graphs and Limitations

Our ability to intervene on the *treated concept* and estimate $\widehat{TReATE}_{TC}$ is dependent on the world model we assume, as presented in the causal graph. For the examples presented in Figures 1 and 13 we have suggested causal graphs (Figure 2) where the relationship between the concepts generating the text is rather simple, as all concepts generate the text without any inheritance relations (i.e., where one concept causes the other). In many interesting cases, the relationship between concepts is not as straight-forward, and might affect our ability to intervene on some concepts. In Figure 14, we consider two such cases, where one or more concepts $A_j|j \in \{0, 1, \dots, k\}$ cause a concept B, which in turn generates a text X. In the clinical note example, it could be the case that a patient's condition is causing a doctor to recommend a specific treatment, and this decision induces the doctor to write the note. In Figure 14 we propose two causal graphs that model this data-generating process, for the case where many conditions cause the doctor's decision (top graph) and for the case where only the patient's depression affects the decision (bottom graph).

   Intervening on specific concepts and computing the causal effect of those concepts on the decisions made by the classifier is not as straightforward. In the first case, where *Anxiety*, *Depression*, and *Suicidal* are causing the doctor's *Clinical Recommendation*, intervening on *Depression* would also affect the *Clinical Recommendation*, as it is one of its causes. While our method can accommodate for this case, our estimator will measure both the direct effect of *Depression* and its indirect effect, through the *Clinical Recommendation*. In such a case it would be impossible to preserve all the information resulting from the *Clinical Recommendation* concept while omitting the information from the *Depression* concept, due to the inheritance relation between the two concepts. Inter-vening on the *Clinical Recommendation* will be even more problematic, as a text without a recommendation will have to be blank according to this graph, regardless of the patient's underlying condition.

   The causal graph presented in the bottom of Figure 14 is also possible, meaning that it could be a reasonable world model in some cases. However, if we were to intervene on *Depression*, we will not be able to know what the doctor would have recommended. In this case we could not estimate the causal effect of *Depression* or that of the *Clinical Recommendation* on the classifier's decision. In Section 6.2 we suggest several sanity checks to help modelers understand if the world model they are using has successfully learned to forget the *treated concept* while remembering the *control concepts*, but there are certainly cases where the ability to perform such interventions is limited.

Appendix C. The EEEC Data Set

As discussed in Section 4.2 we provide here additional details about the generation of the EEEC data set. Particularly, in Table 11 we present the templates used to generate the data, and in Table 12 we compare the original EEC and our EEEC, to illustrate the key modifications we have made.

Appendix D. Experimental Pipeline and Hyperparameters

The pipeline of our experiments follows the same steps for all the settings we address. Particularly, we execute the following pipeline for each of the downstream classification tasks (*Sentiment* [Section 5.1 and 5.2] and *POMS* [Section 5.3], as well as for the *TC* and

Table 11
The templates used to generate the EEEC data set.

ID	Template	# Sent.
1	Now that it is all over, \<person\> feels \<emotion\>	787
2	\<person\> feels \<emotion\> as \<gender noun\> walks to the \<place\>	490
3	Even though it is still a work in progress, the situation makes \<person\> feel \<emotion\>	286
4	The situation makes \<person\> feel \<emotion\>, and will probably continue to in the foreseeable future	1,145
5	It is a mystery to me, but it seems i made \<person\> feel \<emotion\>	598
6	I made \<person\> feel \<emotion\>, and plan to continue until the \<season\> is over	1,114
7	It was totally unexpected, but \<person\> made me feel \<emotion\>	691
8	\<person\> made me feel \<emotion\> for the first time ever in my life	1,218
9	As \<gender noun\> approaches the \<place\>, \<person\> feels \<emotion\>	1,504
10	\<person\> feels \<emotion\> at the end	598
11	While it is still under construction, the situation makes \<person\> feel \<emotion\>	400
12	It is far from over, but so far i made \<person\> feel \<emotion\>	531
13	We went to the \<place\>, and \<person\> made me feel \<emotion\>	891
14	\<person\> feels \<emotion\> as \<gender noun\> paces along to the \<place\>	550
15	While this is still under construction, the situation makes \<person\> feel \<emotion\>	335
16	The situation makes \<person\> feel \<emotion\>, but it does not matter now	1,131
17	There is still a long way to go, but the situation makes \<person\> feel \<emotion\>	312
18	I made \<person\> feel \<emotion\>, time and time again	1,188
19	While it is still under development, the situation makes \<person\> feel \<emotion\>	261
20	I do not know why, but i made \<person\> feel \<emotion\>	492
21	\<person\> made me feel \<emotion\> whenever I came near	1,092
22	While we were at the \<place\>, \<person\> made me feel \<emotion\>	648
23	\<person\> feels \<emotion\> at the start	483
24	Even though it is still under development, the situation makes \<person\> feel \<emotion\>	285
25	I have no idea how or why, but i made \<person\> feel \<emotion\>	468
26	We were told that \<person\> found \<gender noun\> in \<ind\> \<emotion\> situation	1,168
27	\<person\> found \<gender noun\> in \<ind\> \<emotion\> situation, after \<time\>	1,164
28	As we were walking together, \<person\> told us all about the recent \<emotion\> events	1,164
29	\<person\> told us all about the recent \<emotion\> events as we were walking to the \<place\>	1,156
30	As expected, the conversation with \<person\> was \<emotion\>	728
31	The conversation with \<person\> was \<emotion\>, we could from simply looking	1,128
32	To our surprise, \<person\> found \<gender noun\> in \<ind\> \<emotion\> situation	1,152
33	\<person\> found \<gender noun\> in \<ind\> \<emotion\> situation, something none of us expected	1,156
34	While we were walking to the \<place\>, \<person\> told us all about the recent \<emotion\> events	1,156
35	The conversation with \<person\> was \<emotion\>, you could feel it in the air	1,192
36	While unsurprising, the conversation with \<person\> was \<emotion\>	748
37	\<person\> told us all about the recent \<emotion\> events, to our surprise	1,164
38	To our amazement, the conversation with \<person\> was \<emotion\>	844
39	I \<observe\> \<person\> in the \<place\> \<day\>.	580
40	I talked to \<person\> \<day\>.	580
41	\<person\> goes to the school in our neighborhood.	580
42	\<person\> has \<number\> \<family\>.	580

CC probing tasks [Section 6.2]) and for each version of the data sets (*Balanced*, *Gentle*, and *Aggressive*):

1. Stage 2 fine-tuning on the training and development sets of the relevant version of the data set (*Balanced*, *Gentle*, or *Aggressive*) to produce the *BERT-CF* and *BERT-MLM* representation models. BERT-CF is trained following the intervention methodology of Section 3.3, while BERT-MLM is trained with standard MLM training as the original BERT model.

Table 12
Descriptive statistics comparing the EEC and Enriched EEC (EEEC) data sets.

Metric	EEC	EEEC
Full Data Size (# of Sentences)	9,840	33,738
Median Sentence Length (# of words)	6	14
# of Templates	11	42
# of Noise Sentences	0	13
# of Prefix Sentences	0	21
# of Suffix Sentences	0	16
# of Emotion Words	40	55
# of Female Names	10	10
# of Male Names	10	10
# of European Names	10	10
# of African-American Names	10	10
# of Places	10	10

2. Stage 3 supervised task training for a classifier based on *BERT-O*, *BERT-MLM*, or *BERT-CF*, for the relevant downstream task (*Sentiment*, *POMS*, *TC*, or *CC* probing).

3. Test our Stage 3 trained *BERT-O*, *BERT-MLM*, and *BERT-CF*–based classifiers on the test set of the downstream task. Particularly, the causal and baseline estimators are computed on the test sets.

In all our experiments we utilize the case-sensitive *BERT-base* pre-trained text representation model (12 layers, 768 hidden vector size, 12 attention heads, $110M$ parameters), trained on the BookCorpus ($800M$ words) (Zhu et al. 2015) and Wikipedia ($2,500M$ words) corpora. This model is publicly available along with its source code via the Google Research GitHub repository.[22]

When training *BERT-CF* (Stage 2), we fine-tune all 12 layers of BERT. For the downstream task classifier we use a fully connected layer that receives as input the token representations produced by BERT's top layer, as well as its *CLS* token. For INLP, which requires a single representation vector as input, we provide the model with the *CLS* token of the *BERT-O*'s top layer. The output produced by INLP for this vector is then provided to the default classifier of that work, namely, logistic regression, in order to compute the TReATE(*O*, *INLP*) estimator.

All our models use cross entropy as their loss function. We utilize the ADAM optimization algorithm (Kingma and Ba 2015) with a learning rate of $1e^{-3}$, fuzz factor of $1e^{-8}$, and no weight decay. We developed all our models and experimental pipelines with PyTorch (Paszke et al. 2017), utilizing and modifying source code from Hugging-Face's Transformers (Wolf et al. 2019) and PyTorch Lightning (Falcon 2019) GitHub repositories.[23]

---

22 https://github.com/google-research/bert.
23 https://github.com/huggingface/transformers, https://github.com/PyTorchLightning/pytorch
   -lightning.

Table 13
The hyperparameters used in our experiments.

Hyperparameter	#
Random Seed	212
Sentiment maximum sequence length	384
POMS maximum sequence length	32
Stage 2 *TC* (adversarial) task λ	1
Stage 2 number of epochs	5
Stage 2 Sentiment batch size	6
Stage 2 POMS batch size	24
Stage 3 number of epochs	50
Stage 3 Sentiment batch size	128
Stage 3 POMS batch size	200
Stage 3 gradient accumulation steps	4
Stage 3 classifier dropout probability	0.1

Due to the extensive experimentation pipeline, which resulted in a large total number of experiments over many different combinations of data set versions and model variations, we chose not to tune our hyperparameters. Table 13 details the hyperparameters used for all our developed models in all experiments.

References

Adi, Yossi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR 2017*, pages 1–13. Toulon.

Aharoni, Roee, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 3874–3884. Minneapolis, MN. https://doi.org/10.18653/v1/N19-1388

Akbik, Alan, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649. Santa Fe, NM.

Angrist, Joshua D. and Jörn-Steffen Pischke. 2008. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press. https://doi.org/10.2307/j.ctvcm4j72

Athey, Susan, Guido Imbens, Thai Pham, and Stefan Wager. 2017. Estimating average treatment effects: Supplementary analyses and remaining challenges. *American Economic Review*, 107(5):278–281. https://doi.org/10.1257/aer.p20171042

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Blitzer, John, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 440–447. Prague.

Bottou, Léon, Jonas Peters, Joaquin Quiñonero Candela, Denis X. Charles, Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Y. Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260.

Boyd-Graber, Jordan L., Yuening Hu, and David M. Mimno. 2017. Applications of topic models. *Foundations and Trends in Information Retrieval*, 11(2–3):143–296. https://doi.org/10.1561/1500000030

Caliskan, Aylin, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186. https://doi.org/10.1126/science.aal4230, PubMed: 28408601

Chang, Ming-Wei, Kristina Toutanova, Kenton Lee, and Jacob Devlin. 2019. Language model pre-training for

hierarchical document representations. *CoRR*, abs/1901.09128.

Che, Tong, Yanran Li, Ruixiang Zhang, R. Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. 2017. Maximum-likelihood augmented discrete generative adversarial networks. *arXiv preprint arXiv:1702.07983*.

Conneau, Alexis, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2126–2136. Melbourne. `https://doi.org/10.18653/v1/P18-1198`

D'Amour, Alexander, Peng Ding, Avi Feller, Lihua Lei, and Jasjeet Sekhon. 2017. Overlap in observational studies with high-dimensional covariates. *arXiv preprint arXiv:1711.02582*.

Davidson, Thomas, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection data sets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35. `https://doi.org/10.18653/v1/W19-3504`

De-Arteaga, Maria, Alexey Romanov, Hanna M. Wallach, Jennifer T. Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Cem Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128. Atlanta, GA. `https://doi.org/10.1145/3287560.3287572`

De Choudhury, Munmun, Emre Kiciman, Mark Dredze, Glen Coppersmith, and Mrinal Kumar. 2016. Discovering shifts to suicidal ideation from mental health content in social media. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2098–2110. San Jose, CA. `https://doi.org/10.1145/2858036.2858207`, PubMed: 29082385

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologie*, pages 4171–4186. Minneapolis, MN.

Dong, Li, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pages 13042–13054. VAncouver.

Dorie, Vincent, Jennifer Hill, Uri Shalit, Marc Scott, Dan Cervone, and others. 2019. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statistical Science*, 34(1):43–68. `https://doi.org/10.1214/18-STS667`

Doshi-Velez, Finale and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv: Machine Learning*.

Egami, Naoki, Christian J. Fong, Justin Grimmer, Margaret E. Roberts, and Brandon M. Stewart. 2018. How to make causal inferences using texts. *arXiv preprint arXiv:abs/1802.02163*.

Elazar, Yanai and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21. Brussels. `https://doi.org/10.18653/v1/D18-1002`

Falcon, WA. 2019. Pytorch lightning. *GitHub*. `https://github.com/PyTorchLightning/pytorch-lightning`.

Feder, Amir, Danny Vainstein, Roni Rosenfeld, Tzvika Hartman, Avinatan Hassidim, and Yossi Matias. 2020. Active deep learning to detect demographic traits in free-form clinical notes. *Journal of Biomedical Informatics*, vol. 107, 103436. `https://doi.org/10.1016/j.jbi.2020.103436`, PubMed: 32428572

Fedus, William, Ian J. Goodfellow, and Andrew M. Dai. 2018. Maskgan: Better text generation via filling in the __. In *International Conference on Learning Representations*, pages 1–17. Vancouver.

Fong, Christian and Justin Grimmer. 2016. Discovery of treatments from text corpora. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1600–1609. Berlin. `https://doi.org/10.18653/v1/P16-1151`

Ganin, Yaroslav, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17:59:1–59:35.

Gao, Jianfeng, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational

AI. In *Proceedings of ACL 2018*, pages 2–7. Melbourne. `https://doi.org/10.18653/v1/P18-5002`

Gardner, Matt, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, and others. 2020. Evaluating models' local decision boundaries via contrast sets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, Online*, pages 1307–1323. `https://doi.org/10.18653/v1/2020.findings-emnlp.117`

Gehrmann, Sebastian, Hendrik Strobelt, Robert Krüger, Hanspeter Pfister, and Alexander M. Rush. 2020. Visual interaction with deep learning models through collaborative semantic inference. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):884–894. `https://doi.org/10.1109/TVCG.2019.2934595`, PubMed: 31425116

Gentzel, Amanda, Dan Garant, and David Jensen. 2019. The case for evaluating causal models using interventional measures and empirical data. In *Advances in Neural Information Processing Systems*, pages 11717–11727. Vancouver.

Ghosal, Deepanway, Devamanyu Hazarika, Abhinaba Roy, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. 2020. Kingdom: Knowledge-guided domain adaptation for sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online*, pages 3198–3210. `https://doi.org/10.18653/v1/2020.acl-main.292`

Gonen, Hila and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 609–614. Minneapolis, MN.

Goyal, Yash, Amir Feder, Uri Shalit, and Been Kim. 2019a. Explaining classifiers with causal concept effect (cace). *arXiv preprint arXiv:abs/1907.07165*.

Goyal, Yash, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019b. Counterfactual visual explanations. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2376–2384. Long Beach, California.

Guo, Jiaxian, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5141–5148. New Orleans, LA.

Gururangan, Suchin, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online*, pages 8342–8360. `https://doi.org/10.18653/v1/2020.acl-main.740`

Huang, Po-Sen, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. 2020. Reducing sentiment bias in language models via counterfactual evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, Online Event*, pages 65–83. `https://doi.org/10.18653/v1/2020.findings-emnlp.7`

Hupkes, Dieuwke, Sara Veldhoen, and Willem H. Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926. `https://doi.org/10.1613/jair.1.11196`

Isabelle, Pierre, Colin Cherry, and George F. Foster. 2017. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496. Copenhagen. `https://doi.org/10.18653/v1/D17-1263`

Jain, Sarthak and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3543–3556. Minneapolis, MN.

Jernite, Yacine, Samuel R. Bowman, and David A. Sontag. 2017. Discourse-based objectives for fast unsupervised sentence representation learning. *arXiv preprint arXiv:abs/1705.00557*.

Johansson, Fredrik D., Uri Shalit, and David A. Sontag. 2016. Learning representations for counterfactual inference. In *Proceedings of the 33nd International Conference on Machine Learning*, pages 3020–3029. New York, NY.

Kaushik, Divyansh, Eduard H. Hovy, and Zachary Chase Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *8th*

International Conference on Learning Representations, pages 1–17. Addis Ababa.

Keith, Katherine A., David Jensen, and Brendan O'Connor. 2020. Text and causal inference: A review of using text to remove confounding from causal estimates. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online*, pages 5332–5344. https://doi .org/10.18653/v1/2020.acl-main.474

Kim, Been, Rajiv Khanna, and Oluwasanmi O. Koyejo. 2016a. Examples are not enough, learn to criticize! Criticism for interpretability. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, pages 2280–2288. Barcelona.

Kim, Been, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2016b. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pages 2668–2677. Barcelona.

Kingma, Diederik P. and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, pages 1–15. San Diego, CA.

Kiritchenko, Svetlana and Saif Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 43–53. New Orleans, LA. https:// doi.org/10.18653/v1/S18-2005

Lau, Jey Han, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539. Gothenburg. https://doi .org/10.3115/v1/E14-1056

Lee, Jinhyuk, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240. https://doi.org/10.1093/bioinformatics /btz682, PubMed: 31501885

Lin, Kevin, Dianqi Li, Xiaodong He, Ming-Ting Sun, and Zhengyou Zhang. 2017. Adversarial ranking for language generation. In *Advances in Neural Information Processing Systems*, pages 3155–3165, Long Beach, CA.

Linzen, Tal, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes, editors. 2019. *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence.

Lipton, Zachary C. 2018. The mythos of model interpretability. *Communications of ACM*, 61(10):36–43. https://doi.org /10.1145/3233231

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907 .11692*.

Logeswaran, Lajanugen and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *6th International Conference on Learning Representations*, pages 1–15. Vancouver.

Lorberbom, Guy, Tommi S. Jaakkola, Andreea Gane, and Tamir Hazan. 2019. Direct optimization through arg max for discrete variational auto-encoder. In *Advances in Neural Information Processing Systems*, pages 6200–6211. Vancouver.

Lundberg, Scott M. and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774. Long Beach, CA.

Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In the *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150. OR.

Naik, Aakanksha, Abhilasha Ravichander, Carolyn Penstein Rosé, and Eduard H. Hovy. 2019. Exploring numeracy in word embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 3374–3380. Florence.

Oved, Nadav, Amir Feder, and Roi Reichart. 2020. Predicting in-game actions from interviews of NBA players. *Computational Linguistics*, 46(3):667–712.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 79–86. Philadelphia, PA.

Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang,

Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*, pages 1–4. Long Beach, CA.

Pearl, Judea. 1995. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688. https://doi.org/10.1093/biomet/82.4.669

Pearl, Judea. 2009a. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146. https://doi.org/10.1214/09-SS057

Pearl, Judea. 2009b. *Causality*. Cambridge University Press.

Pennebaker, James W., Martha E. Francis, and Roger J. Booth. 2001. Linguistic inquiry and word count: LIWC 2001. http://downloads.liwc.net.s3.amazonaws.com/LIWC2015_OperatorManual.pdf.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237. New Orleans, LA. https://doi.org/10.18653/v1/N18-1202

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning, Technical report, OpenAI.

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

Ramponi, Alan and Barbara Plank. 2020. Neural unsupervised domain adaptation in NLP - A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855. Barcelona. https://doi.org/10.18653/v1/2020.coling-main.603

Ravfogel, Shauli, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online*, pages 7237–7256. https://doi.org/10.18653/v1/2020.acl-main.647

Řehůřek, Radim and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50. Valletta, Malta. http://is.muni.cz/publication/884893/en

Ribeiro, Marco Túlio, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. San Francisco, CA. https://doi.org/10.1145/2939672.2939778

Roberts, Margaret E., Brandon M. Stewart, and Richard A. Nielsen. 2018. Adjusting for confounding with text matching. *American Journal of Political Science*, 64:887–903.

Roberts, Margaret E., Brandon M. Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson, and David G. Rand. 2014. Structural topic models for open-ended survey responses. *American Journal of Political Science*, 58(4):1064–1082. https://doi.org/10.1111/ajps.12103

Rotman, Guy and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics*, 7:695–713.

Rudinger, Rachel, Chandler May, and Benjamin Van Durme. 2017. Social bias in elicited natural language inferences. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing, EthNLP@EACL, Valencia, Spain, April 4, 2017*, pages 74–79. Valencia. https://doi.org/10.18653/v1/w17-1609.

Saha, Koustuv, Benjamin Sugar, John Torous, Bruno D. Abrahao, Emre Kiciman, and Munmun De Choudhury. 2019. A social media study on the effects of psychiatric medication use. In *Proceedings of the International AAAI Conference on Web and Social Media*, pages 440–451. Munich.

Sari, Yunita, Mark Stevenson, and Andreas Vlachos. 2018. Topic or style? Exploring the most useful features for authorship attribution. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 343–353. Santa Fe, New Mexico.

Semeniuta, Stanislau, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 627–637. Copenhagen. https://doi.org/10.18653/v1/d17-1066

Sennrich, Rico. 2017. How grammatical is character-level neural machine translation? Assessing MT quality with contrastive translation pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 376–382. Valencia. `https://doi.org/10.18653/v1/e17-2060`

Subramanian, Sandeep, Sai Rajeswar, Francis Dutil, Chris Pal, and Aaron C. Courville. 2017. Adversarial generation of natural language. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 241–251. Vancouver. `https://doi.org/10.18653/v1/W17-2629`

Sun, Chi, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? *arXiv preprint arXiv:1905.05583*. `https://doi.org/10.1007/978-3-030-32381-3_16`

Swinger, Nathaniel, Maria De-Arteaga, Neil Thomas Heffernan IV, Mark D. M. Leiserson, and Adam Tauman Kalai. 2019. What are the biases in my word embedding? In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 305–311. Honolulu, HI. `https://doi.org/10.1145/3306618.3314270`

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826. Las Vegas, NV. `https://doi.org/10.1109/CVPR.2016.308`

Tan, Chenhao, Lillian Lee, and Bo Pang. 2014. The effect of wording on message propagation: Topic- and author-controlled natural experiments on Twitter. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 175–185. Baltimore, MD. `https://doi.org/10.3115/v1/p14-1017`

Tshitoyan, Vahe, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A. Persson, Gerbrand Ceder, and Anubhav Jain. 2019. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763):95–98. `https://doi.org/10.1038/s41586-019-1335-8`, PubMed: 31270483

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 5998–6008. Long Beach, CA.

Veitch, Victor, Dhanya Sridhar, and David M. Blei. 2019. Using text embeddings for causal inference. *arXiv preprint arXiv:abs/1905.12741*.

Vig, Jesse, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M. Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, virtual*.

Wiegreffe, Sarah and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 11–20, Hong Kong. `https://doi.org/10.18653/v1/D19-1002`

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's Transformers: State-of-the-art natural language processing. *ArXiv, abs/1910.03771*.

Wood-Doughty, Zach, Ilya Shpitser, and Mark Dredze. 2018. Challenges of using text classifiers for causal inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4586–4598. Brussels. `https://www.aclweb.org/anthology/D18-1488/`.

Woodward, James. 2005. *Making Things Happen: A Theory Of Causal Explanation*. Oxford University Press.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, and others. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xie, Qizhe, Zihang Dai, Yulun Du, Eduard H. Hovy, and Graham Neubig. 2017. Controllable invariance through adversarial feature learning. In *Advances in*

Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, pages 585–596. Long Beach, CA.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2018. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*, pages 5754–5764. Vancouver.

Zech, John R., Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric Karl Oermann. 2018. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study.

PLoS Medicine, 15(11):e1002683. `https://doi.org/10.1371/journal.pmed.1002683`, PubMed: 30399157

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

Ziser, Yftah and Roi Reichart. 2018. Pivot based language modeling for improved neural domain adaptation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1241–1251. New Orleans, Louisiana. `https://doi.org/10.18653/v1/n18-1112`