Understanding Text Classifiers with Counterfactual Explanation

--- By Zhen Tan and Nayoung Kim

Abstract

Spurious correlations have been proved to be detrimental to the real-world language understanding tasks, where the performance of a model will degrade tremendously when the test distribution shifts from the training distribution. In this project, we try to investigate the counterfactual-based method to mitigate or explain the impact of spurious correlations on text classification systems. On two benchmark datasets, we evaluate three methods, each of which tries to mitigate the keyword bias, label bias, and both respectively. Furthermore, we study multiple model-agnostic counterfactual search algorithms and try to provide a qualitative analysis of the generated examples to better understand the ground behind the decision of the machine learning classifier.

I. Introduction

Text classification is a vital and prevailing task for the various applications of machine learning and natural language processing, such as sentiment analysis, topic classification, disinformation detection, .etc. Traditionally, people only pay attention to the performance of interpolation, where the target test data is assumed to share the same distribution with the training data. Simultaneously, with the rising of deep learning, various deep models have been proposed to try to boost the accuracy during the inference phase. However, more recently, people find that the performance of those traditional methods will degrade a lot when applied to extrapolation, a more realistic scenario, where there is a so-called distribution shift between the training set and test set. Purely based on correlation learning, vanilla statistical machine learning models might suffer from undesired spurious correlations, thus resulting in biased prediction in the shifted distribution domain. For example, if in a sentiment analysis training dataset, compared to “straight”, the word “gay” occurs more often with negative labels, the model might simply learn the correlation that the text with “gay” should be categorized into the negative class whatever the content in the text is.

This phenomenon is evidently undesirable in any text classification system. On the other hand, we want our model to learn to predict based merely on

the casual attributes of the input text and mitigate the impact of spurious correlation as much as possible. In Potential Outcome Framework (POF), a classic framework of causal inference, those kinds of spurious correlations are usually viewed as “confounding” or “back-door”. To make sure that the target model learns the desired causal relationship, the counterfactual-based method has emerged as an effective way to measure, mitigate and explain the impact of spurious correlations. A common way is to modify the words that strongly signify those correlations into their antonyms, and test the model. For example, for a given factual sentence with the word “straight”, we modify “straight” into “gay”, and keep everything else the same, then it becomes a counterfactual sentence. If the model produces the same output for all the factual and counterfactual sentences, then we can say this model is robust to this spurious correlation, “gay” and “straight”. This intuitive method has been proved effective, and subsequently, many more advanced approaches have been proposed based on this framework, among which, three methods will be further investigated in this project.

Furthermore, for a case study, we deploy a counterfactual-example-based framework to explain the prediction of a classifier that judges whether a person has an annual income greater or less than 50K given his/her personal information. Each person is depicted by a set of text attributes, and the machine learning classifier is trained. Using different model-agnostic methods as well as gradient-based methods, we produce counterfactuals that are both feasible and versatile with their attributes. Through several implementations, we find that the ML model recognizes the hidden correlations between attributes and hidden bias in the data. Moreover, we produce a causal importance map of all those attributes and find out which features have a larger impact on the model than others.

Overall, in this project, we try to investigate the existing counterfactual-based methods with two focuses:

1) How to use counterfactuals to mitigate spurious correlations?
2) How to use counterfactuals to explain the output of a given model?

II. Related Work

1) Spurious Correlation in Machine Learning
   Spurious correlations are problematic and could be introduced in many ways. In [4], researchers show that bias towards gender, race, etc. originating from training data imbalances can be amplified by deep models like CNN or RNN. Besides that, data leakage [5] and distribution shift between training data and testing data [6] are particularly challenging and hard to detect as they introduce spurious correlations during model training and hurt model performance when deployed. [7] shows that bias and spurious correlations vary from task to task, and when multiple such correlations occur, it is hard to mitigate all of them simultaneously, and a trade-off between different correlations is worth further investigating.
2) Causal Inference to Mitigate Impact of Spurious Correlation
   For this project, we investigate three methods. CLP [1] is the first work that bridges the gap between fairness and robustness. It aims at mitigating the impact of word-level keyword bias (e.g., “gay” vs “straight”). Its main contribution is proposing a counterfactual augmentation mechanism to generate the counterfactuals and a regularizer to push the model to yield similar results for factuals and counterfactuals. On the other hand, AGC [3] designs a different counterfactual augmentation mechanism that generates counterfactuals with different labels as the factuals. This method can be used to tackle document-level label bias, where the number of data samples in each category differs significantly from others. Furthermore, Corsair [2] proposes a framework for mitigating both label bias and keyword bias at the same time. Notably, this method tries to debias the model during the evaluation phase. Namely, a biased model is directly obtained after training on the biased dataset. Then, during inference, given a factual input document, CORSAIR imagines its two counterfactual counterparts to distill and mitigate the two biases captured by the poisonous model.
3) Explaining Classifiers through Counterfactual Examples
   An example-based counterfactual explanation is one way to interpret the ML model. Kim et al. [10] proposed a novel example-based explanation framework that uses both prototypes and criticisms from

the original instances. Recently, counterfactual explanations have started to be used as a perturbation of the given input to generate different outputs using the same model. Watcher et al. [11] propose a formula to generate counterfactuals. This model induces the counterfactual to have a different output to the original instance, as well as keep proximity between them to minimize transformation of the original data.

III. Model Description

1) CLP

Suppose f is the target text classification model, g is a counterfactual generation function. In this paper, g is to change some pre-defined words into their antonyms and keep others the same. Now, we have:

$$\sum_{x \in X} J(f(x), y) + \lambda \sum_{x \in X} \mathop{\mathbb{E}}_{x' \sim \text{Unif}[\Phi(x)]} |g(x) - g(x')|$$

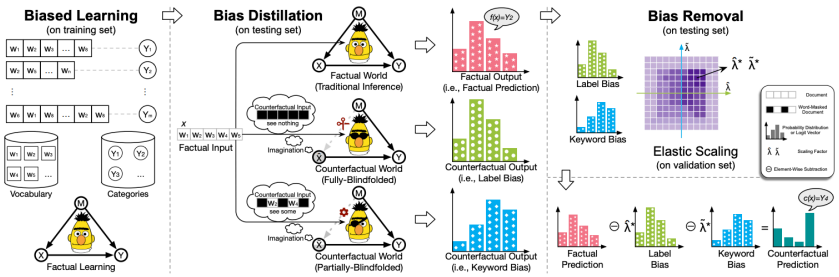
Where J is the original loss function like cross-entropy loss, and the latter term is the regularizer that pushes the model to generate similar output for factual and counterfactual data, and λ is a factor to balance the importance between them.

2) AGC

a. Find out the top causal term t (BERT)
b. Change it to its antonym
c. Use it as the augmented data
d. Train the model with original data and augment data

	Term	Original coef	Robust coef	Original sentence	Counterfactual sentence
Non-causal terms	movie	-0.236	0.028	**Terrible** movie	**Fantastic** movie
	free	-1.41	-0.919	This was a free book that sounded **boring** to me.	This was a free book that sounded **interesting** to me.
Causal terms	awesome	0.584	1.838	He was an **awesome** actor.	He was an **awful** actor.
	terrible	-1.283	-2.336	The whole movie consists of **terrible** dialogue.	The whole movie consists of **pleasant** dialogue.

3) Corsair



a. Train a biased model on the biased dataset
b. Label Bias Distillation: xˆ denotes the imagined fully-blindfolded counterfactual document where all words in the test document x are consistently masked

$$P(Y|do(X)) = f(\hat{x}) = f(\langle w_1, w_2, \cdots, w_n \rangle)$$
$$\forall w_i \in \hat{x}, w_i \leftarrow \texttt{[MASK]}$$

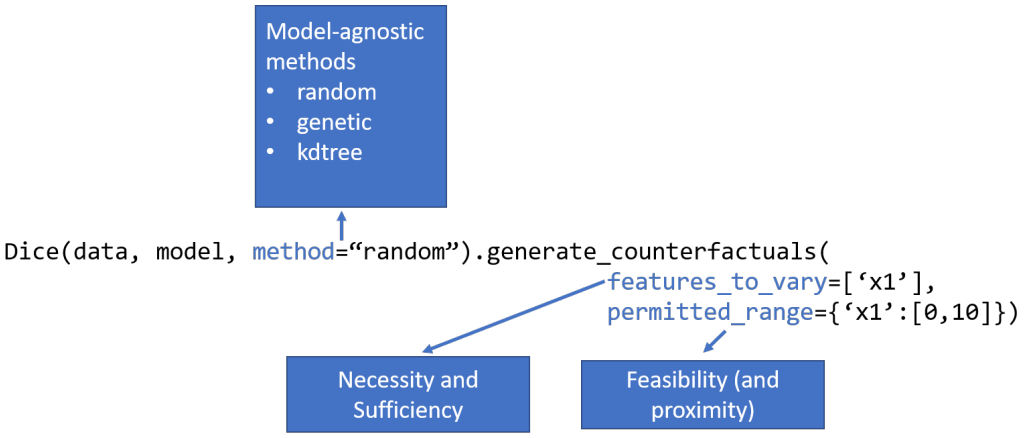
c. Keyword Bias Distillation: x˜ denotes another counterfactual document where the main-content words in a test document x are masked while other context words are not

$$f(\tilde{x}) = f(\langle w_1, w_2, \cdots, w_n \rangle)$$
$$\forall w_i \in \tilde{x}, \begin{cases} w_i \leftarrow \texttt{[MASK]} & \text{if } w_i \in x_{\text{content}} \\ w_i \leftarrow w_i & \text{if } w_i \in x_{\text{context}} \end{cases}$$

d. Bias Removal:

$$c(x) = f(x) \backslash f(\hat{x}) \backslash f(\tilde{x}) = f(x) - \hat{\lambda} f(\hat{x}) - \tilde{\lambda} f(\tilde{x})$$

4) DiCE Framework



DiCE [12] is an open-source framework that provides an interface to generate various counterfactual examples for any ML model. In addition to proximity (minimal changes) and diversity, DiCE controls the feasibility of the counterfactuals automatically. This means it ensures not only that certain values are feasible, but also the changes indicated by a counterfactual are feasible for each individual. For example, 20 years may be a feasible value for Age feature, but changing a person's age from 22 to 20 is not feasible.

To make the changes in a counterfactual example feasible, we specified the possible ranges for continuous features and possible values for categorical features.

Given an instance x and a trained machine learning model, DiCE generates a set of k counterfactual examples, {c1, c2, . . . , ck }, such that all of them have a different output from x. The instance x and all counterfactual examples {c1, c2, . . . , ck } are d-dimensional. We further describe the instances and generated examples in the experiment section.

5) Model-agnostic counterfactual generation methods
    a. Random Sampling

        Random Sampling is a simple and naive approach to generating counterfactual explanations. It randomly changes feature values of the instance of interest and stops when the desired output is predicted.

    b. Genetic Algorithm

        Genetic algorithm is one popular method in artificial intelligence to solve an optimization problem. Given a population of candidate solutions, this algorithm keeps generating the next-generation population through two genetic operators, crossover (or recombination) and mutation. A recent study [14] shows that genetic algorithms perform efficiently for counterfactual explanations. When it is used to generate counterfactuals, it searches the space of counterfactuals by prioritizing those that have fewer changes. Starting from a population consisting of just the given entity $\boldsymbol{x}$, the algorithm repeatedly updates the population by applying the operations crossover and mutation and then selecting the best counterfactuals for the new generation. It stops when it reaches a sufficient number of examples on which the classifier returns the good (or desired) outcome.

c. Querying K-D Tree

K-D tree (or k-dimensional tree) [13] is a binary tree-based data representation in a k-dimensional space. Every node has a k-dimensional point, and every non-leaf node serves as a splitting hyperplane that divides the space into two parts. Therefore, the left subtree of that node would reside in the left side of this hyperplane, and vice versa.

When the K-D tree is used to generate counterfactuals, each class i is represented by a separate k-d tree using the instances having class label i. For each class-specific k-d tree, the Euclidean distance between instance x and the k-nearest item in the tree is calculated. The closest item across all classes except for the class of x becomes the class prototype and becomes part of the loss function.

IV. Experiment

1) Mitigate Impact from Spurious Correlation

a. Experiment Settings

We test the three methods on HyperPartisan [8] (HYP for short) and Twitter [9] (TWI for short) (two benchmark text binary classification datasets). The baseline we deploy here is TextCNN for all three methods. We use the widely-used macro-F1 metric, which is the balanced harmonic mean of precision and recall. Furthermore, macro-F1 is more suitable than micro-F1 to reflect the extent of the dataset biases, especially for the highly-skewed cases, since macro-F1 is strongly influenced by the performance in each category (i.e., category sensitive) but micro-F1 easily gives equal weight over all the documents (i.e., category-agnostic)

b. Comparable Results:

Accordingly, for these two specific datasets we choose, Corsair has the best performance. We can see that compared to the other two methods, CLP has lower scores. This is because CLP highly relies on manually collected word pairs like “straight” and “gay”. If for some data, words in the collected word pairs do not show up that often, for example, never occur in the dataset, then this method will not work. Also, CLP and AGC only consider one kind of bias, either keyword bias or label bias. On the

contrary, Corsair takes both into account, and it does not require any extra manual cost of data collection, selection, and annotation. So it outperforms all other methods.

Methods	HYP (%)	TWI (%)
TextCNN	40.48	65.94
TextCNN + CLP	41.29	66.18
TextCNN + AGC	45.85	67.81
TextCNN + Corsair	**46.68**	**68.83**

Table 1. Result of the three methods on the two benchmark datasets HYP and TWI

2) Explanation of the model through diverse Counterfactual examples

a. Dataset

We used transformed tabular text data Adult dataset [15] called Adult-Income for generating counterfactuals. The adult dataset consists of 26,048 instances and each has 8 attributes. Given the attributes of each instance, machine learning models (Random Forest Classifier in this experiment) are trained to predict if the income is over 50,000 or not. We used tabular data instead of text data to observe differences between each method and better analyze the generated counterfactuals. A detailed description of each attribute is as follows.

age	workclass	education	Marital_status	occupation	race	gender	Hours_per_week	income
28	Private	Bachelors	Single	White-Collar	White	female	60	0
30	Self-Employed	Assoc	Married	Professional	White	Male	65	1
32	Private	Some-college	Married	White-Collar	White	male	50	0

Table 2. Examples of Adult-Income Instances

b. Generating Counterfactual Examples

In this experiment, we show that the set of counterfactuals explain the given machine learning model, especially its local decision boundary to predict. Also, we observe there are several differences between various counterfactual generation methods.

We first choose a trained classification model RandomForestClassifier. This model can be substituted for other text classifiers. Then we produce several representative counterfactual examples using the three model-agnostic methods: random search, genetic search, and KD tree search (Table 1). We could understand the result explanations through the generated set of counterfactuals. First, most of the counterfactuals are changed in a reasonable way that we can understand with common sense. For example, studying for an advanced degree can lead to a higher income. But it also shows less obvious counterfactuals such as getting married (in addition to finishing professional school and increasing hours worked per week) or working less for a higher income. These counterfactuals are likely generated due to underlying correlations in the dataset. In some specific regions, married people have higher income and some rich people like landlords work shorter than employees.

We also observe hidden social bias under the data. When we release the restriction of some attributes, like gender or race from unchangeable to changeable, some counterfactuals of a given instance induce the instance to become a man or to become a white. This shows that white people or men have a higher income than other races or women. It indicates the bias in the dataset contaminates the classifier, and the counterfactual examples reveal it.

We also try to perform additional evaluations and find the differences between the methods. Since we restrict some of the attributes to unchangeable and the number of counterfactuals generated, all the generated examples are feasible. However, there is a significant difference in the runtimes of each method searching for the counterfactuals. Since random search does not use loss function to

find desired output, running time is much faster than the other two methods. The speed of random sampling is 7.48 iteration/s, whereas genetic search and KD tree search take 1.24 iteration/s and 2.83 iteration/s each. This gap is larger when compared to the gradient-based method like Tensorflow or PyTorch neural networks.

Method	age	workclass	education	Marital_status	occupation	race	gender	hours_per_week	income
Query	29	Private	HS-grad	Married	Blue-Collar	White	female	38	0
Random	-	Masters	-	-	Other/Unknown	-	-	-	1
Genetic	-	Masters	-	-	Professional	-	-	-	1
KD tree	-	-	Assoc	-	Service	-	-	-	1
Tensorflow	-	-	Doctorate	-	Service	-	-	26	1
PyTorch	-	Self-Employed	Masters	-	Professional	-	-	-	1

Table 3. Query instance x and generated counterfactuals

Furthermore, we measure the importance of each attribute when the model-agnostic methods generate counterfactuals. Table 4 shows the global importance scores of each attribute estimated by aggregating the scores over individual inputs. These scores are computed for a given query instance by summarizing a set of counterfactual examples around the instance. Results show that each approach has a different perspective to understand the model by focusing on different attributes to generate counterfactuals. For example, the “occupation” attribute is important when the KD tree explains the model, but not important for Random search. Considering that “occupation” and “education” attributes to change the most in Table 3, this indicates that these attributes highly affect the decision of the classifier.

Method	education	occupation	marital_status	age	hours_per_week	workclass	race	gender
Random	**0.635**	0.29	0.285	0.25	0.195	0.17	0.135	0.07
Genetic	**0.7**	0.685	0.38	0.42	0.11	0.255	0.13	0.135
KD Tree	0.715	**0.8**	0.39	0.07	0.065	0.37	0.165	0.265

Table 4. Global feature importance scores

V. Future Works

Currently, all the state-of-the-art methods we mentioned above omit one important factor when applying counterfactual methods to the text classification task, that the counterfactual and factual pairs should share the same or similar distribution. In other words, the two sentences should be semantically similar. All these works directly change the word/token in the original sentence and assume that, with only one word modified, the new sentence will share the same semantics. But no proof has been given so far. Fixing this issue would be meaningful work.

Reference

[1] Garg, Sahaj, et al. "Counterfactual fairness in text classification through robustness." *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019.

[2] Qian, Chen, et al. "Counterfactual Inference for Text Classification Debiasing." *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021.

[3] Wang, Zhao, and Aron Culotta. "Robustness to Spurious Correlations in Text Classification via Automatically Generated Counterfactuals." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. No. 16. 2021.

[4] Kiritchenko, Svetlana, and Saif M. Mohammad. "Examining gender and race bias in two hundred sentiment analysis systems." *arXiv preprint arXiv:1805.04508* (2018).

[5] Roemmele, Melissa, Cosmin Adrian Bejan, and Andrew S. Gordon. "Choice of plausible alternatives: An evaluation of commonsense causal reasoning." 2011 AAAI Spring Symposium Series. 2011.

[6] Quiñonero-Candela, Joaquin, et al., eds. *Dataset shift in machine learning*. Mit Press, 2009.

[7] Mehrabi, Ninareh, et al. "A survey on bias and fairness in machine learning." ACM Computing Surveys (CSUR) 54.6 (2021): 1-35.

[8] Kiesel, Johannes, et al. "Semeval-2019 task 4: Hyperpartisan news detection." Proceedings of the 13th International Workshop on Semantic Evaluation. 2019.

[9] Huang, Xiaolei, et al. "Examining patterns of influenza vaccination in social media." *Workshops at the thirty-first AAAI conference on artificial intelligence*. 2017.

[10] Kim, Been, Rajiv Khanna, and Oluwasanmi O. Koyejo. "Examples are not enough, learn to criticize! criticism for interpretability." *Advances in neural information processing systems* 29 (2016).

[11] Wachter, Sandra, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." *Harv. JL & Tech.* 31 (2017): 841.

[12] Mothilal, Ramaravind K., Amit Sharma, and Chenhao Tan. "Explaining machine learning classifiers through diverse counterfactual explanations." *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020.

[13] Bentley, Jon Louis. "Multidimensional binary search trees used for associative searching." *Communications of the ACM* 18.9 (1975): 509-517.

[14] Schleich, Maximilian, et al. "GeCo: Quality counterfactual explanations in real time." *arXiv preprint arXiv:2101.01292* (2021).

[15] Kohavi, Ronny, and Barry Becker. "Uci machine learning repository: adult data set." *Avaliable: https://archive. ics. uci. edu/ml/machine-learning-databases/adult* (1996).

Appendix

Github Repo Link: https://github.com/Zhen-Tan-dmml/CSE-472-Project-2.git