Making a (Counterfactual) Difference One Rationale at a Time

Mitchell Plyler
Department of Computer Science
North Carolina State University
mlplyler@ncsu.edu

Michael Green
Laboratory for Analytic Sciences
magree22@ncsu.edu

Min Chi
Department of Computer Science
North Carolina State University
mchi@ncsu.edu

Abstract

Rationales, *snippets of extracted text* that explain an inference, have emerged as a popular framework for interpretable natural language processing (NLP). Rationale models typically consist of two cooperating modules: *a selector* and *a classifier* with the goal of maximizing the mutual information (MMI) between the "selected" text and the document label. Despite their promises, MMI-based methods often pick up on *spurious* text patterns and result in models with nonsensical behaviors. In this work, we investigate whether *counterfactual data augmentation (CDA)*, without human assistance, can improve the performance of the selector by lowering the mutual information between spurious signals and the document label. Our counterfactuals are produced in an unsupervised fashion using class-dependent generative models. From an information theoretic lens, we derive properties of the unaugmented dataset for which our CDA approach would succeed. The effectiveness of CDA is empirically evaluated by comparing against several baselines including an improved MMI-based rationale schema [19] on two multi-aspect datasets. Our results show that CDA produces rationales that better capture the signal of interest.

1 Introduction

Research in neural model interpretability has been cast as important and received significant recent attention [21]. Within the field of natural language processing (NLP), *rationales* have been a popular method for providing interpretability in the form of extracted subsets of text [10]. Rationale models typically consist of two cooperating modules where one module, the "rationale selector", selects the rationale from a source document, and the other module, the "classifier", acts on only the selected rationale without seeing the rest of the document. There is interpretability through sparsity and exclusivity.

A common approach for training these rationale models is based on the maximum mutual information criteria (MMI) [7]. With the MMI criteria, rationale selectors seek the subset of text that carries the most information about the target label. Often, sparsity and coherency constraints are used to keep the rationales interpretable. Within many datasets, however, *spurious patterns* and *co-varying aspects* can cause the rationale selector to pick up on patterns that do not capture a desired relationship between input text and target labels. As a result, the rationalized model can have undesirable behaviour like predicting a hotel is very clean because it is in a convenient location. Nonsensical rationales or

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

explanations might decrease trust in the model, and in some cases, suggest the model might generalize poorly [26].

In this work, we propose a general *counterfactual data augmentation (CDA) [22] approach* to aid rationale models trained with MMI. We show that theoretically our CDA approach can effectively improve the performance of rationale selectors by lowering the mutual information between spurious signals and aspects of interest. Empirically, we show that models trained on our CDA datasets learn higher quality rationales than those trained on the original dataset when both use the same MMI criteria. More importantly, *the most significant advantage of our CDA approach is that it does not require human intervention*. We use rationales from an initial, noisy model and replace them with new text that changes the target label using a generative neural model. In this way, our CDA approach is completely hands off and does not need input from human experts or crowd workers.

We first show that in the extremely ideal scenario where the initial rationale selector is perfect, our CDA approach can eliminate the mutual information between spurious signals and the target label. Next, we show in the common, realistic scenario where the rationale selector is noisy and imperfect, our CDA approach can still yield gains. Finally, the effectiveness of our CDA approach is compared against several baselines including an improved MMI-based rationale schema [19] on two common *multi-aspect* review datasets, *TripAdvisor* [29] and *RateBeer* [24]. Multi-aspect datasets are our main focus as they are guaranteed to contain *spurious patterns* and *co-varying aspects*; we primarily used MMI-based baselines because the fundamental goal our CDA approach is to reduce the mutual information between spurious signals and aspects of interest.

2 Counterfactual Data Augmentation and Multi-Aspect Datasets

2.1 Definitions and Notations

We use upper-case letters to denote random variables, X and Y, and lower-case letters to denote samples from these variables, x and y. $I(X, Y)$ marks the mutual information between X and Y. Mutual information is defined as the reduction in uncertainty of a random variable due to knowledge in another random variable, $I(X, Y) = H(Y) - H(Y|X)$, where $H(X)$ and $H(Y|X)$ are the Shannon entropy and conditional entropy respectively [8].

2.2 Problem Formulation

This work follows the same rationale concept introduced in [19]. Specifically, one neural module extracts text from a document and another neural module classifies the extracted text. Later it has been shown that the rationalization criteria aims "to maximize the mutual information between selected features and the response variable" [7] defined as:

$$\max_{G} I(X_M; Y) \quad \text{subject to} \quad M \sim G(X) \tag{1}$$

where M denotes a binary rationale mask over the input produced by a rationale selector G. That is, the goal of the rationale selector is to select the subset of features in X that are *most informative* of the label Y under some constraints defined by the selector G. In this work, our selector constraints affect the size and coherency of the rationales.

Here we consider a multi-aspect dataset, D with features *X* and labels *Y*: X is a set of features, or a sequence of words in NLP, and Y is a vector, possibly one dimensional, of numerical scores. In a multi-aspect dataset, a single document can discuss multiple attributes of a single object. For example, a single beverage review might discuss its appearance, taste, and smell. We assume some subset of the features, X_1, belong to the target aspect label, Y_1, while other features, X_2, are spurious or non-causal. These features could belong to other aspects or be artifacts of the dataset. In the following, for simplicity, we will use <X, Y_1> or <X_1, X_2, Y_1> interchangeably based on the context.

Our goal is to estimate or model the score for aspect Y_1 by using the function f given *only* X_1, as follows: $Y_1 = f(X_1)$. There is one such model corresponding to each $< X_1, Y_1 >$ pair. In this work, we use multi-aspect datasets, and model each aspect individually. This simulates the more common case where a dataset has a single output label of interest and all signals in the dataset that do not pertain to that label are considered spurious or belonging to other, not-estimated aspects.

When predicting Y_1, our ideal model would focus on X_1 and ignore all other features, X_2. Specifically, for a given sample <x, y>, our selector would select a subset of x such that $x_1 = x_M$. A rationale selector might fail to extract x_1 effectively for two primary reasons: first, Chen et al. [7] showed it is intractable to find a solution to Eqn 1 and thus they derive a variational approximation with Monte Carlo based gradient estimates. Second, datasets can contain artifacts such that spurious patterns might contain significant mutual information with the target label. Along the same lines but even more concerning, other aspects in a dataset might be highly correlated with the target label. For example, beers that smell good usually taste good as well.

Since the goal of the selector is to select features that maximize $I(X_M, Y_1)$, we can assist the selector in finding X_1 by lowering the mutual information between the other spurious features and the desired label, $I(X_2, Y_1)$. Drawing on ideas from [22], we do this through a general counterfactual data augmentation (CDA) scheme where, in the counterfactual dataset, superscript c, we flip the label of the document from Y_1 to Y_1^c and replace the text selected by the rationale selector, X_1, with an inference, X_1^c, generated by a *class conditioned masked language model (MLM)* using Y_1^c and X_2 described as:

$$Y_1^c \leftarrow 1 - Y_1; \quad X_1^c \leftarrow \arg\max_{X_1} p(X_1 | 1 - Y_1, X_2) \tag{2}$$

In our generated counterfactual dataset $< X_1^c, X_2, Y_1^c >$, X_1^c is the newly generated counterfactual, X_2 is the original spurious feature set, and we assign $Y^c = 1 - Y_1$. We will show, in the augmented dataset which is the concatenation of datasets $< X_1, X_2, Y_1 >$ and $< X_1^c, X_2, 1 - Y_1 >$, we are lowering the mutual information $I(X_2, Y_1)$. We will use superscript a for the augmented dataset: D^a = <X_1^a,X_2^a, Y_1^a>.

2.3 Lowering $I(X_2, Y_1)$, an idyllic case

Take a dataset of beer reviews where each document contains a description of the taste and smell of a beer as well as a numerical score for only the smell aspect. The task is to estimate the smell score while using the smell text as the rationale. Figure 1 demonstrates our process. An example of a concise document in this dataset might be *This beer smells great. It tastes terrific*. In this document x_1 is the phrase *smells great* and x_2 is *tastes terrific*. Both have positive sentiment but our only label for this document is $y_1 = 1$ for the smell sentiment. Our counterfactual document could be *This beer smells awful. It tastes terrific* and our label becomes $y_1 = 0$. We can see that in the augmented dataset the phrase *tastes terrific* maps to both positive and negative labels and therefore $p(Y_1|X_2) = p(Y_1)$. Finally, $I(X_2, Y_1)$ is 0 and $I(X_1, Y_1)$ is unchanged.

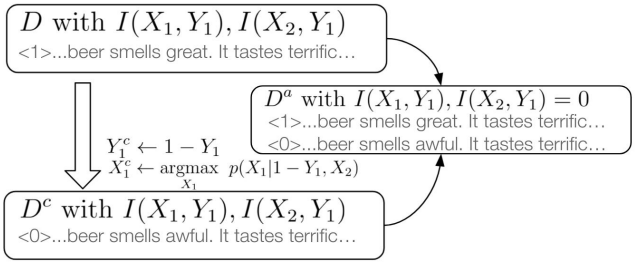

Figure 1: Toy example to demonstrate our approach. In the augmented dataset D^a, the mutual information, I, between the smell score and the smell text is preserved while the mutual information between the smell score and the taste text is eliminated.

With perfect knowledge of the ground truth rationales and the process $X_1^c \leftarrow p(X_1|1 - Y_1, X_2)$, we can craft counterfactual documents and therefore a counterfactual dataset that perfectly eliminates $I(X_2, Y_1)$ while preserving $I(X_1, Y_1)$. However, the challenge is that ground truth rationales are not provided in the training data. Therefore, it is important for us to show that even when rationales selected by the initial rationale selector are noisy and imperfect, we can still lower $I(X_2, Y_1)$ in the augmented dataset and benefit subsequent models trained with MMI.

2.4 Dealing with a Noisy Initial Selector

Here we are working on the augmented dataset: D^a = <X_1^a, X_2^a, Y^a>. To completely eliminate $I(X_2^a, Y_1^a)$, our CDA approach requires a perfect rationale selector. We were originally motivated by improving a poor rationale selector, so here we track what happens to both $I(X_2^a, Y_1^a)$ and $I(X_1^a, Y_1^a)$ when the rationale selector is not perfect. Our new goal is to reduce $I(X_2^a, Y_1^a)$ more than we reduce

$I(X_1^a, Y_1^a)$. We analyze the procedure in the worst case scenario in order to determine a lower bound on our algorithm's benefits under some assumptions.

In an extremely erroneous case, we say that for a given <x_1, x_2, y_1>, our initial rationale selector mistakenly selects x_2 when aiming for x_1. When creating the corresponding counterfactual document, we still have $y_1^c = 1 - y_1$, but we modify the document according to $x_2^c \leftarrow \arg\max_{X_2} p(x_2|1 - y_1, x_1)$ instead of the process defined by Eqn 2. Going back to our concise example, this would be the counterfactual document *This beer tastes good. It also smells bad*. In this extremely erroneous case, we have decreased $I(X_1, Y_1)$ while $I(X_2, Y_1)$ remains unchanged. Thus, we define the worst case scenario as reducing $I(X_1, Y_1)$ at some error rate α and keeping $I(X_2, Y_1)$ constant at the same rate.

If we say that this error happens to all samples with a rate α, we can analyze conditions that must be present in the original dataset so that our CDA approach is beneficial. Let's first define ΔI^a as the change in mutual information from the original to the augmented dataset.

$$\Delta I^a_{X_i,Y_j} = I(X_i, Y_j) - I(X_i^a, Y_j^a) \tag{3}$$

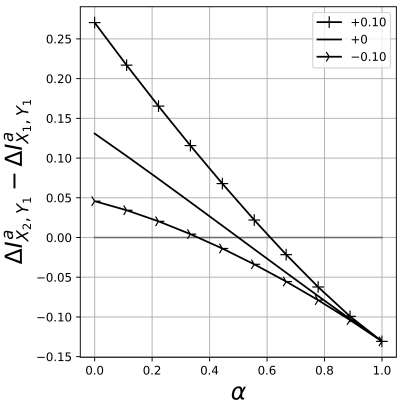

Figure 2: CDA benefits (y-axis) as function of error rate α. Each line represents $p(Y_1|X_2) = p(Y_1|X_1) \pm c$.

In order for our CDA approach to be beneficial, we need to decrease $I(X_2, Y_1)$ more than $I(X_1, Y_1)$. That is:

$$\Delta I^a_{X_2,Y_1} - \Delta I^a_{X_1,Y_1} > 0 \tag{4}$$

For samples in the original dataset where the initial selector results in an error, X_1 would map to $1 - Y_1$ for the corresponding counterfactuals. Furthermore, the augmented dataset would map X_1 to both Y_1 and $1 - Y_1$ for these original erroneous samples and corresponding counterfactuals, and thus, X_1 is no longer informative of Y_1 on such samples. That is, in the augmented dataset and with proportion α, $p(Y_1^a|X_1^a) = p(Y_1^a)$. For the remaining $1 - \alpha$ portion of non-error samples, CDA can successfully capture $p(Y_1|X_1)$ in the original dataset and $p(Y_1^c|X_1^c)$ in the counterfactual dataset. To quantify the conditional distributions of the augmented dataset, we make two assumptions: first, we assume that we can successfully generate the counterfactuals, and thus, we can assume $p(Y_1^c|X_1^c) = p(Y_1|X_1)$; second, we assume that the erroneous samples happen randomly across the original dataset, and the marginal distributions of the original and augmented datasets are the same. Based on these two assumptions, we have:

$$p(Y_1^a|X_1^a) = \alpha p(Y_1) + (1 - \alpha)p(Y_1|X_1) \tag{5}$$

For X_2, the success and failure cases are reversed.

$$p(Y_1^a|X_2^a) = (1 - \alpha)p(Y_1) + \alpha p(Y_1|X_2) \tag{6}$$

We can now expand Eqn 4 using the definition $I(X, Y) = H(Y) - H(Y|X)$.

$$0 < -H(Y_1|X_2) + H(Y_1^a|X_2^a) + H(Y_1|X_1) - H(Y_1^a|X_1^a) \tag{7}$$

Using the definition, $H(Y|X) = -E\ \log p(Y|X)$, we can expand this further to

$$0 < -E\ \log p(Y_1|X_1) + E\ \log p(Y_1^a|X_1^a) + E\ \log p(Y_1|X_2) - E\ \log p(Y_1^a|X_2^a) \tag{8}$$

Eqn 8 describes the conditions that must be met in our original dataset in order to yield gains from our CDA procedure for some error rate α. It is impossible to calculate this relation directly because it will require exact knowledge of our ground-truth rationales. In order to shed some light on this, Figure 2 shows the efficacy of the CDA approach when we approximate X_1 and X_2 with binary variables, $p(Y_1|X_1) = \frac{3}{4}$, $p(X_1) = p(X_2) = p(Y_1) = \frac{1}{2}$. When X_1 and X_2 are equally informative of Y_1, the benefits of CDA decrease linearly with the error rate, and intuitively, our error rate must be less than 50% to see gains. When X_2 is more informative than X_1, we have a higher error budget to see any benefit, and when X_1 is more informative than X_2, our error budget is smaller. According to this analysis, we have a higher error budget when spurious signals offer the same or more information about the target label. When the spurious signals carry much less information, the initial selector must have a low error rate in order to benefit from the CDA approach.

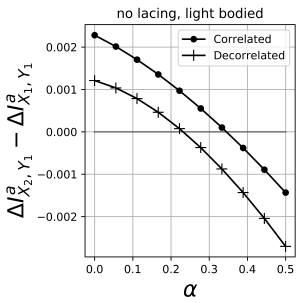

Figure 3: CDA benefits when approximating X_1 and X_2 as the occurrence of bi-grams

With the datasets used in this work, we can gain insight by approximating X_1 and X_2 with guessable bigrams. Here is another concise example: X_1 is a binary variable that indicates the occurrence of the phrase *no lacing* and X_2 is a binary variable for the phrase *light bodied*. For a beer's appearance aspect, *no lacing* is a strong indicator for a negative score, and for its palate aspect, *light bodied* is a strong negative indicator. As beer's appearance is highly correlated with its palatability, both phrases indicate a low appearance score. Taking Y_1 as the appearance score, X_1 as the occurrence of *no lacing*, and *light bodied* as X_2, we can numerically use Eqn 8 to examine how CDA helps us for varying error rates in Figure 3. CDA is beneficial whenever the curve is above zero. In the fully correlated dataset, "Correlated", described in section 4.1, the information carried by *light bodied* is closer to that of *no lacing* than it is in the decorrelated dataset. As a consequence, in the correlated dataset, we have a higher error budget and more opportunities for our CDA.

3 Architecture and Implementation

3.1 Rationale Framework

The original rationale framework, as viewed in this work, was introduced by [19]. It was not our goal to change the core rationalization algorithm, so we re-implemented the algorithm and updated some details. At a high level, our rationale framework is the same in that we use one network to select the rationale and another network to classify the text. The rationale framework can be visualized by the blue portion of Figure 4.

The original implementation [19] used RNNs for both networks, REINFORCE [30] for dealing with the discontinuity introduced by the binary rationale mask, and variable percentage rationales. In this work, we use transformers for both networks because of their effectiveness over RNNs in NLP [28], the simpler straight-through method [3] [5] instead of REINFORCE, and we use fixed percentage rationales because it eliminates sparsity related hyperparameters. Our fixed percentage rationales differ from Chen et al. [7] in that we use the top-K tokens during training and inference whereas Chen et al. [7] use an iterative re-sampling approach with Gumbel-softmax reparameterization [15].

The classifier is trained only to make quality predictions against the labels while using the rationale. This is the cross-entropy between the labels and the classifier's prediction, L_y. We follow [19] by using a coherency regularizer for the rationale selector: $L_r = \frac{\lambda_r}{T}\sum_{1\ldots T}|m_t - m_{t-1}|$ where T is the total number of tokens in a document, m is the rationale mask, and λ_r is the hyper-parameter used to tune coherency. This encourages the rationales to be contiguous. For the datasets evaluated in this work, coherency is a useful inductive bias. The loss for the rationale selector L_s is the coherency regularizer and the cross-entropy between the labels and the classifier's prediction. This is $L_s = L_r + L_y$.

3.2 Counterfactual Predictor

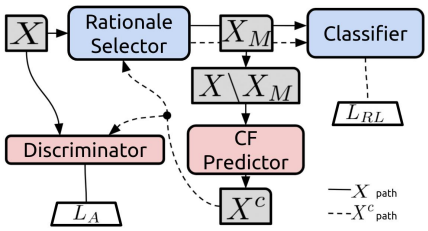

Figure 4: CF Predictor training flow.

The CDA process described earlier, Eqn 2, requires us to generate new documents with the text and label flipped for just one aspect. That is, we are sampling a new document from $p(X_1|X_2, 1 - Y_1)$. The key challenge is that *we are not provided with ground truth rationales or counterfactuals with which to learn how this data generation process works*. This can be viewed under the lens of unsupervised style transfer for which there is significant prior work in the NLP domain [20] [23]. Our method for generating counterfactual documents leverages many ideas from these works, and our main contribution here is connecting these ideas to the rationale framework. We incorporate the rationale framework in the counterfactual generation process because our goal is to lower the mutual

information between the spurious signals and the target label. An off-the-shelf style transfer method might focus on signals other than that selected by our initial rationale selector.

Figure 4 shows the CF Predictor's training flow. At its core, the CF Predictor leverages class-dependent Masked-language models (MLM) [9]. We replace the original document's rationale with an inference from a MLM and leave the rest of the document unchanged. The MLMs are trained to produce documents with the desired class through reinforcement learning (RL) [12], and they are trained to produce realistic documents through adversarial learning [11]. We use straight-through [3] to propagate gradients through token selection during training. For a dataset with binary labels, we train two separate MLMs: one for generating class-0 documents and another for generating class-1 documents [31]. The loss for the counterfactual predictor, L_{CFP} is defined as:

$$L_{CFP} = \lambda_{RL} L_{RL} - \lambda_A L_A \tag{9}$$

where L_{RL} is the classifier loss after passing the counterfactual through the rationale selector and the classifier. It is the cross-entropy between the desired label, all ones or zeros, and the predicted label. L_A is the loss component from adversarial training. Following [11] and [32], our discriminator, $\mathcal{D}$, seeks to distinguish generated documents from the originals, so the discriminator's loss, $L_{\mathcal{D}}$, is $\frac{1}{\lambda_A} L_A$ where L_A is the cross-entropy between real-fake labels and the prediction given by the discriminator. λ_A and λ_{RL} balance the adversarial and classification losses. We include $\frac{1}{\lambda_A}$ when training the discriminator to hamstring it relative to the predictor. We found that this generally helped us avoid mode collapse commonly seen when adversarially training generative models. For L_A, we mask the contribution of original documents without the desired label. When training the class-1 counterfactual predictor, we show the discriminator real documents X with $Y_1 = 1$ and counterfactual documents, X^c, where the original documents' labels were $Y_1 = 0$.

During training, the counterfactual is produced in one step. All words not included in the selected rationale from the original document, X_M, are kept in the counterfactual document. The kept tokens are $X \backslash X_M$. All words in the selected rationale are replaced by the CF Predictor using one prediction from the MLMs in a greedy fashion. After training, when producing the counterfactual dataset, the counterfactual documents still keep all non-rationale tokens. We now replace the rationale tokens from left to right using the output of the CF Predictor MLMs. A counterfactual token at position t is decoded according to the following process.

$$x^c_{1,t} = \arg\max_{x_{1,t}} p(x_{1,t} | x_2, 1 - y_1, x^c_{1,0\ldots t}) \tag{10}$$

We found this to be a good trade-off between greedy decoding and a more expensive beam search. Greedy decoding could generate frequent, repeated tokens while beam search could be an unnecessary expense for generating documents that reflect the target distribution, but do not necessarily need to pass the bar of human readers.

Figure 5 illustrates an original document and its counterfactual generated by our CF predictor. Notice here the initial selector was successful in identifying the smell aspect. The inclusion of the original and its counterfactual document in the augmented dataset successfully decreases the mutual information between the non-smell text and the smell label.

4 Experiments

4.1 Datasets

We conduct experiments using datasets from two sources. This first source contains reviews compiled by Wang et al. [29] from *TripAdvisor.com*. We use the training, dev, and test sets curated by Bao et al. [2] and used for rationalization by Chang et al. [5]. The label is binary, and we focus on the *location* aspect. There are 198 test samples with human-annotated rationales.

The second source consists of reviews collected by McAuley et al. [24] from *RateBeer*. Each review is a paragraph of text with five numerical scores in the range of 1 to 5 for the appearance, smell, palate, taste, and overall aspects of the beer. We focus on the appearance, smell, and palate aspects. We created training and dev datasets for each aspect that follow the source distribution. Following [5], we binarize this data so that all reviews with a score ≥ 3 are class 1 and all reviews with a score ≤ 2 are class 0. We then balance the dataset between classes. Additionally, we use procedurally decorrelated datasets created by Lei et al. [19], and binarize these datasets as well. We now have two

Label	Document
negative	presentation : 12 oz clear bottle , label is metallic with an emblem of a buck . inkjetted on the neck i 'm not sure if this is out of date , or was produced then . appearance : pours clear , somewhere between an amber and a brown color . average head , but it fades pretty quickly to leave just a hint of lace **. smell : a bit of grain , fairly mild . this is sort of a** generic beer aroma , without much to pick out specifically . taste : a little bit of sweetness and malt this tastes a bit like a light marzen . overall impression : it is n't bad it is sort of a generic beer with a bit more flavor than your average macro sort of like a yuengling . nothing to write home about , but it 's drinkable enough .
positive	presentation : 12 oz clear bottle , label is metallic with an emblem of a buck . inkjetted on the neck i 'm not sure if this is out of date , or was produced then . appearance : pours clear , somewhere between an amber and a brown color . average head , but it fades pretty quickly to leave just a hint of lace **. smell : caramel malt and toffee dominate the nose with a bit of raisin and** generic beer aroma , without much to pick out specifically . taste : a little bit of sweetness and malt this tastes a bit like a light marzen . overall impression : it is n't bad it is sort of a generic beer with a bit more flavor than your average macro sort of like a yuengling . nothing to write home about , but it 's drinkable enough .

Figure 5: A subset of the augmented dataset for beer smell. The rationale is bold in the original document (top). The replaced words are bold in the counterfactual document (bottom). x_1 and y_1 have changed from original to counterfactual. Ground truth rationales and counterfactuals are not provided in any training data.

datasets for each RateBeer aspect: a "correlated" set and a "decorrelated" set. Appendix Section A.2 shows correlation matrices before and after decorrelating the data as well as additional dataset details. In the correlated datasets, the correlations and presumably the mutual information between aspects are much higher than in the decorrelated datasets. This makes the rationalization task more difficult. Additionally, McAuley et al. [24] provides about 1,000 holdout samples where the aspect specific text is annotated by human experts.

4.2 Baselines

Three MMI-based baselines are used. First, "MMI" is the original Rationalization scheme [19] that has been updated and re-implemented to use the most recent NLP models as described in section 3.1. This model is trained on the original, unaugmented dataset.

Second, Factual Data Augmentation (FDA) is used to verify that our gains are not only due to data augmentation but due to the CDA procedure. FDA augments the original dataset with new samples generated by the CF predictor models, but instead of flipping the label and passing it to the $1 - y_1$ component, we pass the sample to the component of the CF predictor with the *same* label as the original document during inference. The new samples are produced using the following process:

$$y_1^c \leftarrow y_1; \qquad x_1^c \leftarrow \arg\max_{x_1} p(X_1|y_1, x_2) \tag{11}$$

The third baseline, simple substitution using antonyms (ANT), does not use neural models to augment counterfactual data. The counterfactual is generated by replacing words in the rationale with antonyms from WordNet [25] [4]. Note that we only accept antonyms that are in the vocabulary used by the models and antonyms that have the same part of speech as labeled by NLTK [4].

4.3 Experiment Settings and Assumptions

We train the rationale selector and the classifier together, early stop based on the selector cost, freeze the selector, and finally fine-tune the classifier on the original dataset. For all of the data sets and models, we use the dev set for early stopping (more details in Appendix Section A.3). Our MLM transformers [28] were pretrained on *unlabeled* data from the TripAdvisor and RateBeer datasets *separately*. For the TripAdvisor dataset, we pretrain on all data that does not appear in the location aspect's train, dev, or annotated dataset from [5]. For the *RateBeer* datasets, we pretrain on all data that does not appear in any train, dev, or annotated dataset from [19] and the correlated datasets. We used a masking rate of 10% and masked tokens were treated as described in [9]. For the rationale models, the transformers are initialized from models pre-trained with random masking. We found

Hotel - Location

we only stayed one night . the location was great and the premises was beautiful . the room was a typical hotel room but was decent , clean and spacious enough for our needs . we were given a first floor room next to the pool . it was relatively quiet . the staff were informative and friendly . the daily parking was a total rip - off at $ 20/day but i would think that the rates are similar anywhere else you stay on the island . all in all , it was a good enough stay .

Correlated - Smell

pours two thick fingers of tan froth on a dark brown body . good nose , with caramel and brown sugar sweetness arising to greet it . the taste blends the sweetness with a dash of hopped bitterness well . well carbonated , this is a sturdy , very drinkable beer that has joined me for much leisure time . i wo n't apologize for that ; nor will you after a couple brown ales .

Figure 6: Examples from the annotated sets. Hotel-Location (top) and Correlated-Smell (bottom). ***Human annotations are underlined***. CDA rationales are in blue. MMI rationales are in red. Overlaps between CDA and MMI are in magenta.

that contiguous masking was better for pre-training the CF predictor. This contiguous masking is similar to that from Joshi et al. [16], but masked tokens are treated the same as in Devlin et al. [9] and there is only one contiguous span. The rationale selector and classifier are initialized from the random-masking transformers. For the rationale selectors, following [5], we set the rationale percentage to 10% for all datasets. The CF predictor components and the GAN discriminator are initialized from the contiguous-masking transformers. These models all have a vocabulary with 2^{15} tokens, 8 layers, 8 attention heads, and a hidden dimension of 256. Appendix Section A.5 shows our server configurations and more details on our experiment setup.

Models are selected and reported based on the best performance on the dev set across a grid search. All methods are evaluated using the same grid search when training the rationale model. The model is selected before fine-tuning the classifier with a frozen rationale selector. The initial selector used to train the counterfactual predictors was the selected MMI model with a fine-tuned classifier. The parameters and checkpoints for the CF Predictor models are tuned and chosen to maximize the accuracy of the training documents' predicted label as compared to the target label (measured by the original rationale model) and to maximize the entropy in the inserted counterfactual tokens. The CF Predictor model is chosen from a grid search, using only the training dataset, across λ_A and λ_{RL}.

For the TripAdvisor dataset, we repeat the experiment three times: training an initial rationale selector, training a counterfactual predictor, generating a counterfactual dataset, and training a new rationale model. Additionally, we train two additional rationale models with different random seeds and the selected hyperparameters. Consequently, there are nine rationale models for each of the four methods, and we report the mean and standard deviation across those nine models. In the RateBeer datasets, we repeated the same experiments two times and then trained two additional models with different random seeds and the selected hyperparameters for each rationale model. As a result, we have a total of six runs for each method for the RateBeer datasets. All models are in Tensorflow [1]. Our software is publicly released [1].

4.4 Results

We evaluate the rationale models by the precision (Rat. Prec.) as compared to human annotations. This token-level metric is taken as the mean across samples in the annotated set. We also report the accuracy of the classifier on the development set (Dev. Acc.).

As shown in Tables 1, 2, and 3, our CDA approach outperforms all baselines on $\frac{6}{7}$ experiments. As expected, CDA's performance over other methods is most pronounced on the correlated RateBeer datasets compared to on the decorrelated ones. Only on the decorrelated appearance dataset, CDA performs second to MMI. This may be because on the decorrelated appearance dataset the spurious aspects have very low mutual information with the target label, appearance, and therefore the error margin for the initial selector is very tight and CDA is expected to yield low gains as discussed in Section 2.4.

[1] github.com/mlplyler/CFs_for_Rationales

The baseline augmentation schemes (FDA, ANT) produced mixed results. The FDA scheme should not have changed the mutual information between the spurious aspects and the target label. Whenever it performed worse than MMI alone, it most likely introduced noise into the dataset. While the ANT scheme might be effective in lowering the mutual information between the spurious aspects and the target label, the generated counterfactual does not likely model $p(x_1|1 - y_1, x_2)$, so we expect it to also reduce $I(X_1, Y_1)$.

Empirically, we have demonstrated that models trained on the CDA augmented data tend to outperform models trained on the original datasets. This lends credence to the idea that our scheme is indeed lowering the mutual information between the spurious aspects and the target label. The dataset and method pairs with very high variance are dragged down by degenerated runs where the rationale selector has little or no skill. These runs were seen when varying the random seed with the selected hyperparameters. Removing the degenerate runs can bring the mean rationale precision of the high variance experiments closer to CDA, but we believe these degenerate models seen primarily in the baselines align well with our theory that CDA allows the rationale selector to more easily identify X_1.

Table 1: TripAdvisor - Location

	Dev. Acc.	Rat. Prec.
MMI	78.16 ± 5.83	26.14 ± 13.25
FDA	80.61 ± 4.38	31.36 ± 10.38
ANT	69.79 ± 2.90	12.25 ± 3.64
CDA	78.11 ± 6.77	**39.76** ± 10.48

Case Study: Figure 6 presents a case study comparing the rationales selected by MMI and our proposed CDA approach in the TripAdvisor and Correlated Beer datasets respectively. As shown in Figure 6, our CDA models can select the text that aligns better with human annotations and they successfully avoid selecting sentiment-carrying text that is not relevant to the aspect of interest.

Table 2: Correlated RateBeer Results

	Appearance		Smell		Palate	
	Dev. Acc.	Rat. Prec.	Dev. Acc.	Rat. Prec.	Dev. Acc.	Rat. Prec.
MMI	76.37 ± 3.05	56.28 ± 17.22	79.49 ± 4.53	45.39 ± 16.45	82.67 ± 0.99	34.98 ± 8.79
FDA	75.73 ± 3.64	41.02 ± 5.09	76.43 ± 5.08	41.67 ± 23.73	77.60 ± 6.06	32.35 ± 14.41
ANT	62.09 ± 8.76	43.25 ± 16.51	57.70 ± 9.87	11.72 ± 7.16	61.93 ± 7.59	8.62 ± 12.31
CDA	73.30 ± 2.93	**67.79** ± 10.63	81.82 ± 0.69	**61.84** ± 6.76	80.65 ± 0.64	**41.24** ± 1.72

Table 3: Decorrelated RateBeer Results

	Appearance		Smell		Palate	
	Dev. Acc.	Rat. Prec.	Dev. Acc.	Rat. Prec.	Dev. Acc.	Rat. Prec.
MMI	81.33 ± 0.93	**86.92** ± 7.35	77.99 ± 5.46	79.71 ± 10.56	68.64 ± 10.09	46.39 ± 29.21
FDA	80.67 ± 1.32	81.97 ± 6.00	79.66 ± 1.23	83.62 ± 3.52	77.26 ± 1.90	66.03 ± 3.76
ANT	71.41 ± 9.97	67.67 ± 27.68	58.81 ± 6.31	9.32 ± 10.76	57.75 ± 7.98	10.83 ± 11.15
CDA	80.24 ± 1.07	82.82 ± 8.60	79.18 ± 1.43	**86.66** ± 2.92	76.81 ± 0.85	**67.79** ± 3.20

5 Related Work

The rationale framework used in this work was introduced in [19] and connected to the MMI criteria by Chen et al. [7]. Much of the follow-up work introduced modifications to the learning algorithm to overcome spurious signals picked up by MMI. Yu et al. [33] sought to limit the signal left in the complement of the rationales. The concept of class-wise rationalization was introduced in Chang et al. [5], and Chang et al. [6] introduced a rationale algorithm that seeks invariant rationales across environments. These works modify the rationale framework and the MMI criteria to help the rationale selector find desirable signals. In our work, we keep the MMI criteria unchanged, but instead seek to diminish the undesirable signals through counterfactual data augmentation.

Data augmentation generally and counterfactual data augmentation (CDA) specifically has been a popular technique in recent natural language processing (NLP) work. Lu et al. [22] introduced counterfactual data augmentation as a general methodology and showed their rule-based scheme

could mitigate gender biases commonly seen in word embeddings and NLP models. Other works used CDA to train better named entity recognition (NER) models in the medical text domain [34]. More work [17] continued the ideas of CDA by working with sentiment analysis and natural language inference tasks. They constructed their CDA datasets using crowd workers and showed models trained on the augmented datasets generalized better out-of-domain. Follow-up work [18] showed how their crowd worker interventions reduce spurious signals when viewed through a simplified causal modeling lens. The causal lens and generative counterfactual data augmentation have also shown recent, positive results in the computer vision domain [27].

6 Conclusions

This work presents a counterfactual data augmentation method for lowering the mutual information between spurious signals and a target label in a dataset. We derive theoretical conditions that indicate when our approach will be beneficial to the dataset. Empirically, we show rationale models trained with MMI on our counterfactually augmented datasets result in rationale selectors with improved behaviours. Furthermore, our analysis shows that the benefits of our scheme are proportional to the error rate of the original rationale selector. This suggests we could perform counterfactual data augmentation iteratively where we achieve better rationale selectors after each iteration that could be used as the initial selector in the next round of data augmentation.

Our method does not make any changes to the core rationale algorithm. There has been more work in the literature on rationale algorithms since Lei et al. [19] and Chen et al. [7], but they were not included in this study because they do not necessarily follow the maximum mutual information criteria on which our theory is based. From a practical standpoint, it is likely our CDA method could connect with other rationalization strategies because the CF Predictor is trained using a frozen rationale model. This could be an important area for future study.

7 Broader Impact

This work focuses on methods for training interpretable NLP models. While interpretability is typically a desired trait, defining exactly what is interpretable is not always clear [21]. Rationale models have been shown to not always be faithful [14] [13], and in the worst case, they can degenerate in such a way where the rationale selector makes the decision instead of the classifier [33]. The degenerated case is likely to produce rationales that are nonsensical to human interpretation.

Section 2.4 presents some theoretical proof on how CDA affects the mutual information between text and the target label with different initial selector error rates. Some conclusions drawn in Section 2.4 rely on a simplified model with binary variables. Furthermore, the simplified model makes the assumption that the generated text follows the same distribution as the original dataset. While this is the goal of the adversarial learning strategy that we implemented, we found that the generated text had lower entropy than the rationales they replaced. This is likely due to both the $\arg\max$ in Eqn 2 and L_{RL}.

We have demonstrated positive results for improving rationales for sentiment analysis sentiment classifiers. Generalizing the approach to other NLP tasks is less straightforward. First, the naive approach to generalizing to K classes in the classification setting suggests having K CF Predictor models and generating $K - 1$ counterfactual documents per data sample. This would be a significant increase in computation and resources. Second, for the task of natural language inference, CDA has been successful with human annotators [17]. It might be possible to train a CF Predictor for each type of hypothesis. It is less clear how to train a CF Predictor to modify a premise.

Our method uses data augmentation to lower the mutual information between some signals in documents and a label of interest. When the initial rationale selector aligns with human judgment, models trained on the augmented data tend to better align with human judgement. The method can be thought of as amplifying signals relative to others. We are changing the properties of the dataset. For sensitive tasks, one should check for undesired biases and fairness concerns before and after CDA.

Acknowledgments: We sincerely thank Jascha Swisher, Stephen Shauger, and Ben Caston for their insights and help. We also thank the Laboratory for Analytic Sciences for supporting the work.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[2] Yujia Bao, Shiyu Chang, Mo Yu, and Regina Barzilay. Deriving Machine Attention from Human Rationales. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1903–1913, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1216. URL `https://aclanthology.org/D18-1216`.

[3] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *arXiv:1308.3432 [cs]*, August 2013. URL `http://arxiv.org/abs/1308.3432`. arXiv: 1308.3432.

[4] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[5] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. A Game Theoretic Approach to Class-wise Selective Rationalization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/hash/5ad742cd15633b26fdce1b80f7b39f7c-Abstract.html`.

[6] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant Rationalization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1448–1458. PMLR, November 2020. URL `https://proceedings.mlr.press/v119/chang20c.html`. ISSN: 2640-3498.

[7] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In *Proceedings of the 35th International Conference on Machine Learning*, pages 883–892. PMLR, July 2018. URL `https://proceedings.mlr.press/v80/chen18j.html`. ISSN: 2640-3498.

[8] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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 Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

[10] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A Benchmark to Evaluate Rationalized NLP Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.408. URL `https://aclanthology.org/2020.acl-main.408`.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL `https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html`.

[12] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Toward Controlled Generation of Text. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1587–1596. PMLR, July 2017. URL `https://proceedings.mlr.press/v70/hu17e.html`. ISSN: 2640-3498.

[13] Alon Jacovi and Yoav Goldberg. Aligning faithful interpretations with their social attribution. *Transactions of the Association for Computational Linguistics*, 9:294–310, 2021.

[14] Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. Learning to Faithfully Rationalize by Construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4459–4473, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.409. URL https://aclanthology.org/2020.acl-main.409.

[15] Eric Jang, Shixiang Gu, and Ben Poole. CATEGORICAL REPARAMETERIZATION WITH GUMBEL-SOFTMAX. page 12, 2017.

[16] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. doi: 10.1162/tacl_a_00300. URL https://aclanthology.org/2020.tacl-1.5.

[17] Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. LEARNING THE DIFFERENCE THAT MAKES A DIFFER- ENCE WITH COUNTERFACTUALLY-AUGMENTED DATA. page 17, 2020.

[18] Divyansh Kaushik, Amrith Setlur, Eduard Hovy, and Zachary C Lipton. EXPLAINING THE EFFICACY OF COUNTERFACTUALLY AUGMENTED DATA. page 33, 2021.

[19] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas, 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1011. URL http://aclweb.org/anthology/D16-1011.

[20] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, Retrieve, Generate: a Simple Approach to Sentiment and Style Transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1169. URL https://aclanthology.org/N18-1169.

[21] Zachary C. Lipton. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, June 2018. ISSN 1542-7730. doi: 10.1145/3236386.3241340. URL https://doi.org/10.1145/3236386.3241340.

[22] Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. Gender Bias in Neural Natural Language Processing. In Vivek Nigam, Tajana Ban Kirigin, Carolyn Talcott, Joshua Guttman, Stepan Kuznetsov, Boon Thau Loo, and Mitsuhiro Okada, editors, *Logic, Language, and Security: Essays Dedicated to Andre Scedrov on the Occasion of His 65th Birthday*, Lecture Notes in Computer Science, pages 189–202. Springer International Publishing, Cham, 2020. ISBN 978-3-030-62077-6. doi: 10.1007/978-3-030-62077-6_14. URL https://doi.org/10.1007/978-3-030-62077-6_14.

[23] Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. Politeness Transfer: A Tag and Generate Approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.169. URL https://aclanthology.org/2020.acl-main.169.

[24] Julian J. McAuley, Jure Leskovec, and Dan Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In Mohammed Javeed Zaki, Arno Siebes, Jeffrey Xu Yu, Bart Goethals, Geoffrey I. Webb, and Xindong Wu, editors, *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pages 1020–1025. IEEE Computer Society, 2012. doi: 10.1109/ICDM.2012.110. URL https://doi.org/10.1109/ICDM.2012.110.

[25] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL https://doi.org/10.1145/219717.219748.

[26] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 2662–2670, Melbourne, Australia, August 2017. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-0-3. doi: 10.24963/ijcai.2017/371. URL https://www.ijcai.org/proceedings/2017/371.

[27] Axel Sauer and Andreas Geiger. COUNTERFACTUAL GENERATIVE NETWORKS. page 25, 2021.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[29] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *In Proceedings of KDD '10*, pages 783–792. ACM, 2010.

[30] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 2004.

[31] Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. Mask and Infill: Applying Masked Language Model for Sentiment Transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 5271–5277, Macao, China, August 2019. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-4-1. doi: 10.24963/ijcai.2019/732. URL `https://www.ijcai.org/proceedings/2019/732`.

[32] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 2852–2858. AAAI Press, 2017.

[33] Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. Rethinking Cooperative Rationalization: Introspective Extraction and Complement Control. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4092–4101, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1420. URL `https://www.aclweb.org/anthology/D19-1420`.

[34] Xiangji Zeng, Yunliang Li, Yuchen Zhai, and Yin Zhang. Counterfactual Generator: A Weakly-Supervised Method for Named Entity Recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7270–7280, Online, November 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-main.590`.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes]
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Original version in the original supplemental material. Final version to be released in public repository.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] in supplemental material
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We complete each experiment two or three times and then vary the random seed two additional times
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] in supplemental material
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] To our knowledge all assets are open sourced
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] in supplemental material. New, highly-correlated split on RateBeer data.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

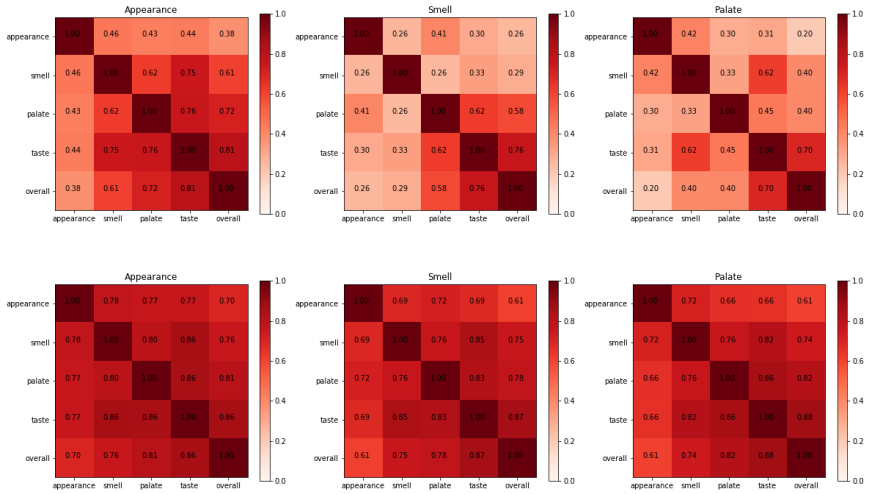

Figure 7: RateBeer Datasets Correlations. Decorrelated (top) and Correlated (bottom).

A Appendix

A.1 Funding Transparency Statement

1) The Laboratory for Analytic Sciences supported this work. 2) No competing interests.

A.2 Dataset Descriptions

Table 4 describes the size of the datasets used in this work. For the *RateBeer* datasets, the correlated and decorrelated datasets are the same size for each of the three aspects. After binarizing the data, the data is balanced by taking the samples in the order that they appear in the source files. We do not include annotated samples with empty rationales. For the TripAdvisor dataset, the training dataset is modified from its source such that no dev or test samples are in the training set. All documents are a maximum of 256 tokens.

Figure 7 shows correlation matrices for the decorrelated (top) and correlated (bottom) datasets for *RateBeer*. This is for non-binarized data but after balancing the classes. The correlated datasets more closely follow the distribution of the full *RateBeer* datasets. Because there is much more correlation between the aspects, the rationale selector has a more difficult time identifying the correct aspect. To our knowledge, these datasets are open for academic use.

Table 4: Datasets Descriptions. Formatted as class-0 / class-1.

Source	Dataset	Train	Dev	Annotated
TripAdvisor	Location	6,715 / 6,715	895 / 895	95 / 103
RateBeer (Decorr)	Appearance	16,891 / 16,891	2,103 / 2,103	13 / 923
	Smell	15,169 / 15,169	2,218 / 2,218	29 / 848
	Palate	13,652 / 13,652	2,000 / 2,000	20 / 785
RateBeer (Corr)	Appearance	16,891 / 16,891	2,000 / 2,000	13 / 923
	Smell	15,169 / 15,169	2,000 / 2,000	29 / 848
	Palate	13,652 / 13,652	2,000 / 2,000	20 / 785

Dataset/Experiment	MMI	FDA	ANT	CDA
Location 1	8	15	10	12
Location 2	20	18	20	5
Location 3	17	17	12	16
Appearance-Decorr 1	17	8	5	21
Appearance-Decorr 2	14	15	5	21
Smell-Decorr 1	18	22	16	15
Smell-Decorr 2	20	20	20	22
Palate-Decorr 1	20	20	5	22
Palate-Decorr 2	15	20	17	22
Appearance-Corr 1	12	16	9	9
Appearance-Corr 2	18	17	15	18
Smell-Corr 1	20	15	10	18
Smell-Corr 2	21	17	13	15
Palate-Corr 1	4	15	19	15
Palate-Corr 2	14	16	17	18

Table 5: Coherency lambda, L_c, selected for rationale models. The dataset number denotes the experiment number.

A.3 Training Procedure

A.3.1 Masked-Language-Models

We trained MLM models for the *RateBeer* and *TripAdvisor* data separately. These MLMs used a dropout rate of 10%. The *RateBeer* random-mask and contiguous-mask MLMs used a batch size of 128 across 4 GPUs and trained for one week. The *TripAdvisor* random-mask MLM used a batch size 256 across 8 GPUs while the contiguous-mask MLM used a batch size of 128 across 4 GPUs. Both trained for one week.

A.3.2 Rationale Models

We train the rationale models in two phases. For the first, we train the rationale selector and the classifier together. We train for 21 epochs and early stop based on the selector cost. The second phase is for training the classifier only on the original dataset. This is done to recover any loss in accuracy due to noise introduced by data augmentation, but we do this for the models trained on the original data only as well. This second stage is only done for the selected models.

Across a grid search for a method and dataset pair, we select the model with the lowest L_s. Here, this is defined as $L_s = \lambda_c L_c + \lambda_y L_y$. λ_c and λ_y are taken as the inverse means of L_s and L_c across the grid search.

We initialized a grid search for all datasets and methods over the coherent regularizer L_c, and we perform this grid search in two stages. The first stage L_c values considered for *RateBeer* datasets were $[5, 10, 15, 20]$ and the learning rate was 1e-4 for all models. The second stage L_c values used were $[L_{c1} - 2, L_{c1} - 1, L_{c1} + 1, L_{c1} + 2]$ where L_{c1} is the best L_c value from the previouus stage. Each run across all searches used a uniquely drawn random seed, a batch size of 50, a dropout rate of 5% for the selector, and a dropout rate of 10% for the classifier. The rationale classifier used max-out across the time dimension. The rationale selector and the classifier were trained using the Adam optimizer. Table 5 shows the final selected hyper-parameters for all reported rationale models.

A.3.3 CF Predictor Models

The CF predictor models were trained using only the original training dataset. Hyper-parameters L_A and L_{RL} were tuned and checkpoints were chosen based on the percentage of labels that were successfully flipped as measured by the rationale model and based on the entropy of the infilled words. Higher entropy is better as low entropy suggests the model is frequently infilling repeated

Dataset/Experiment	$+\lambda_{RL}$	$+\lambda_A$	$-\lambda_{RL}$	$-\lambda_A$
Location 1	5	20	5	15
Location 2	1	5	1	1
Location 3	15	20	1	10
Appearance-Decorr 1	1	5	5	20
Appearance-Decorr 2	5	15	5	15
Smell-Decorr 1	10	20	5	15
Smell-Decorr 2	5	5	5	20
Palate-Decorr 1	1	5	1	10
Palate-Decorr 2	5	15	1	10
Appearance-Corr 1	1	5	5	15
Appearance-Corr 2	1	5	5	20
Smell-Corr 1	10	20	1	10
Smell-Corr 2	1	5	10	20
Palate-Corr 1	1	20	1	5
Palate-Corr 2	1	10	1	15

Table 6: CF Predictor Selected Hyperparameters.

words. CF Predictors models and checkpoints were chosen based on the metric $4.5a + t$ where a is the percentage of documents whose labeled was flipped (as measured by the rationale model) and t is the entropy of the infilled words. This metric was estimated using the first 500 samples in the training dataset and using one step decoding. This is not the iterative decoding method used to produce the final rationale model. We check this metric every 50 training steps. CF predictors were trained with the Adam optimizer with a linearly increasing and decreasing rate. The peak learning rate was reached in 100 steps. These models used a dropout rate of 10%.

The positive and negative CF Predictors were selected from grid searches using $[1, 5, 10, 15, 20, 25]$ for both L_A and L_{RL} with the condition that $L_{RL} \leq L_A$. All models were trained with the Adam optimizer with a a linearly increasing and decreasing learning rate that peaked at 5e-6 after 100 steps, 10% dropout, and for seven epochs or until the compute node failed. Table 6 shows the final selected hyperparameters.

A.4 Additional Rationale Results

Here we show some additional metrics for the rationale models. The recall and F1 scores shown in Tables 7, 8, and 9 taken as the mean score across documents.

Table 7: TripAdvisor - Location

	Rat. F1	Rat. Recall
MMI	24.89 ± 13.16	30.96 ± 16.81
FDA	30.07 ± 10.45	37.44 ± 13.29
ANT	11.80 ± 3.59	15.61 ± 4.84
CDA	**38.29** ± 10.40	**47.38** ± 13.07

Table 8: Additional Decorrelated RateBeer Results

	Appearance		Smell		Palate	
	Rat. F1	Rat. Recall	Rat. F1	Rat. Recall	Rat. F1	Rat. Recall
MMI	**62.36** ± 5.41	**51.82** ± 4.58	63.84 ± 8.73	57.25 ± 7.96	41.97 ± 26.85	42.80 ± 27.69
FDA	58.37 ± 4.72	48.26 ± 4.20	67.06 ± 2.94	60.24 ± 2.69	59.96 ± 3.41	61.26 ± 3.47
ANT	47.81 ± 20.36	39.31 ± 17.11	7.17 ± 8.02	6.28 ± 6.78	9.27 ± 9.90	9.06 ± 9.85
CDA	59.13 ± 6.55	48.96 ± 5.69	**69.56** ± 2.44	**62.45** ± 2.26	**61.59** ± 2.82	**62.96** ± 2.77

Table 9: Additional Correlated RateBeer Results

	Appearance		Smell		Palate	
	Rat. F1	Rat. Recall	Rat. F1	Rat. Recall	Rat. F1	Rat. Recall
MMI	39.44 ± 12.75	32.32 ± 10.76	35.66 ± 13.05	31.62 ± 11.61	31.34 ± 8.03	31.85 ± 8.18
FDA	27.70 ± 3.81	22.22 ± 3.23	32.72 ± 18.93	28.96 ± 16.91	29.31 ± 13.12	30.03 ± 13.46
ANT	30.03 ± 11.96	24.49 ± 9.99	9.14 ± 5.51	8.05 ± 4.76	7.57 ± 11.03	7.56 ± 11.10
CDA	**48.02** ± 7.64	**39.56** ± 6.28	**48.98** ± 5.56	**43.68** ± 5.12	**37.33** ± 1.64	**38.33** ± 1.87

A.5 Server Configuration

This work was computed on a SLURM cluster with a variety of compute nodes. TripAdvisor rationale models were trained on AMD Epyc Rome CPUs and rtx2060super GPUs. The RateBeer rationale models were trained on Intel Skylake CPUs and p4000 GPUs. All CF Predictor models were trained on rtx2060super GPUs. All rationale models' classifier were finetuned on the p4000 nodes with Skylake CPUs. rtx2060, rtx2060super, and p4000 nodes were used to pre-train the MLMs.