CEBaB: Estimating the Causal Effects of Real-World Concepts on NLP Model Behavior

Eldar David Abraham[*,1]
eldar.a@campus.technion.ac.il

Karel D'Oosterlink[*,2,3]
Karel.DOosterlinck@UGent.be

Amir Feder[*,1]
feder@campus.technion.ac.il

Yair Gat[*,1]
yairgat@campus.technion.ac.il

Atticus Geiger[*,2]
atticusg@stanford.edu

Christopher Potts[*,2]
cgpotts@stanford.edu

Roi Reichart[*,1]
roiri@technion.ac.il

Zhengxuan Wu[*,2]
wuzhengx@stanford.edu

[1]**Technion – Israel Institute of Technology** [2]**Stanford University** [3]**Ghent University**

Abstract

The increasing size and complexity of modern ML systems has improved their predictive capabilities but made their behavior harder to explain. Many techniques for model explanation have been developed in response, but we lack clear criteria for assessing these techniques. In this paper, we cast model explanation as the causal inference problem of estimating causal effects of real-world concepts on the output behavior of ML models given actual input data. We introduce CEBaB, a new benchmark dataset for assessing concept-based explanation methods in Natural Language Processing (NLP). CEBaB consists of short restaurant reviews with human-generated counterfactual reviews in which an aspect (food, noise, ambiance, service) of the dining experience was modified. Original and counterfactual reviews are annotated with multiply-validated sentiment ratings at the aspect-level and review-level. The rich structure of CEBaB allows us to go beyond input features to study the effects of abstract, real-world concepts on model behavior. We use CEBaB to compare the quality of a range of concept-based explanation methods covering different assumptions and conceptions of the problem, and we seek to establish natural metrics for comparative assessments of these methods.

1 Introduction

Explaining model behavior has emerged as a central goal within ML. In NLP, models have grown in size and complexity, and while they have become increasingly successful, they have also become more opaque [28,36], raising concerns about trust [18,23], safety [1,34], and fairness [16,19]. These concerns will persist if these models remain "black-boxes".

Seeking to open the black-box, researchers have developed methods that try to explain model behavior [2,11,13,30,41]. However, there is no consensus about how to evaluate such methods to allow robust

[*]Equal contribution. Author names alphabetical.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Table 1: Toy examples illustrating the structure of CEBaB (actual corpus examples are longer and more complex; a sample is given in Appendix B). Beginning from an OpenTable review, we give crowdworkers an actual restaurant review and they generate counterfactual restaurant reviews that would have been written if some aspect of the dining experience were changed and all else were held constant. Five different crowdworkers labeled each of the actual and counterfactual texts according to their aspect-level sentiment and overall sentiment. Aspect level sentiment labels are three way: '+' (positive sentiment), '–' (negative), and 'unk' (the aspect's value is not expressed in the text). Overall sentiment labels are 1 (worst) to 5 (best). Edited aspect labels are shown in blue.

		food	ambiance	service	noise	overall
Original text:	Excellent lobster and decor, but rude waiter.	+	+	–	unk	4
Edit Goal						
food: –	Terrible lobster, excellent decor, but rude waiter.	–	+	–	unk	2
food: unk	Excellent decor, but rude waiter.	unk	+	–	unk	3
ambiance: –	Excellent lobster, but lousy decor and rude waiter.	+	–	–	unk	3
ambiance: unk	Excellent lobster, but rude waiter.	+	unk	–	unk	3
service: +	Excellent lobster and decor, and friendly waiter.	+	+	+	unk	5
service: unk	Excellent lobster and decor.	+	+	unk	unk	5
noise: +	Excellent lobster, decor, and music, but rude waiter.	+	+	–	+	4
noise: –	Excellent lobster and decor, but rude waiter, and noisy.	+	+	–	–	3

comparisons. This is not surprising, since such evaluations require very rich empirical data. Intuitively, we would like to (1) intervene on model inputs, to modify specific concepts without changing other correlated information, (2) observe the effects this has on model predictions, and, finally, (3) assess explanation methods for their ability to accurately predict these effects.

The absence of interventional data, or even an agreed-upon non-interventional benchmark, has created an environment in which explanation methods are often evaluated individually, and without comparison to alternatives. Attempts have been made to conduct comparative evaluations [11,17,37], but only with synthetic, simplified datasets. Furthermore, these attempts do not define a unified evaluation approach, nor do they seek to contribute benchmark datasets that support such evaluations.

In this paper, we seek to overcome this obstacle by introducing **CEBaB** (**C**ausal **E**stimation-**Ba**sed **B**enchmark). Table 1 summarizes the structure of CEBaB with a toy example: beginning with a review text from the OpenTable website, we crowdsourced edits of the original text that are designed to meet a specific goal, such as changing the food rating in the original text to negative or unknown. All of the resulting edits were validated by five crowdworkers and each full text was evaluated by five crowdworkers for its overall sentiment. CEBaB is grounded in 2,299 original reviews, which were expanded via this editing procedure to a total of 15,089 texts, targeting four different aspect-level concepts (food, service, ambiance, noise) with three potential labels (positive, negative, and unknown, i.e., not expressed in the review), and each full text was labeled on a five-star scale.

We focus on using CEBaB to compare concept-based explanation methods. This allows us to go beyond the effect of individual tokens to study how more abstract concepts (in our case, aspect-level sentiment) contribute to model predictions (about the overall sentiment of the text). Our proposed metrics center around assessing concept-based explanation methods for their ability to accurately estimate *causal concept effects* [17], allowing us to isolate the effect of individual concepts.

More specifically, we use CEBaB to measure the causal effects of particular variables in a causal graph, and we cast each explanation method as a causal estimator of these measurements. For example, suppose our causal graph of the data says that all four of our aspect-level categories will affect a reviewer's overall rating. To estimate the effect of positive food quality on the predicted overall rating from a classifier, we need to compare examples with high food quality to those with low quality, holding all other aspects constant. Such pairs of examples are normally not observed, but this is precisely what CEBaB provides. With CEBaB, we can directly compare the actual change in model predictions with the change that a concept-based explanation method predicts.

In our experiments, we evaluate five leading concept-based explanation methods: CONEXP [17], TCAV [26], ConceptSHAP [57], INLP [40], CausaLM [11], and S-Learner [27]. These methods make a wide range of different assumptions about how much access we have to the model's internal structure, and they also diverge in the degree to which they account for the causal nature of the

concept effect estimation problem. Remarkably, CEBaB reveals that most methods cannot beat a simple baseline. Indeed, this negative result emphasizes the value in our primary contribution of providing the data and metrics that enables a direct comparison of explanation methods.

2 Previous Work

Benchmarks for Explanation Methods Benchmark datasets have propelled ML forward by creating shared metrics that predictive models can be evaluated on [22,25,54,55]. Unfortunately, benchmarks that are suitable for assessing the quality of model explanations are still uncommon [10, 21]. Previous work on comparing explanation methods has generally only correlated the performance of a given explainability method with others, without ground-truth comparisons [8,20,21,43].

Other works that do compare to some ground-truth either employ a non-causal evaluation scheme [26], use causal evaluation metrics which do not capture performance on individual examples [52], evaluate on synthetic counterfactuals and rule-based augmentations [11,52], or are tailored for a specific explanation method and hard to generalize [57]. To the best of our knowledge, CEBaB is the first large-scale naturalistic causal benchmark with interventional data for NLP.

Explanation Methods and Causality Probing is a relatively new technique for understanding what model internal representations encode. In probing, a small supervised [6,51] or unsupervised [5,32,44] model is used to estimate whether specific concepts are encoded at specific places in a network. While probes have helped illuminate what models (especially pretrained ones) have learned from data, Geiger et al. [15] show with simple analytic examples that probes cannot reliably provide causal explanations for model behavior.

Feature importance methods can also be seen as explanation methods [33]. Many methods in this space are restricted to input features, but gradient-based methods can often quantify the relative importance of hidden states as well [3,46,48,58]. The Integrated Gradients method of Sundararajan et al. [50] has a natural causal interpretation stemming from its exploration of baseline (counterfactual) inputs [15]. However, even where these methods can focus on internal states, it remains difficult to connect their analyses with real-world concepts that do not reduce to simple properties of inputs.

Intervention-based methods involve modifying inputs or internal representations and studying the effects that this has on model behavior [30, 41]. Recent methods perturb input or hidden representations to create counterfactual states that can then be used to estimate causal effects [9,12,47,53,15]. However, these methods are prone to generating implausible inputs or network states unless the interventions are carefully controlled [14].

Generating counterfactual texts automatically remains challenging and is still a work-in-progress [4]. To overcome this problem, another class of approaches proposes to manipulate the representation of the text with respect to some concept, rather than the text itself [9,11,40]. These methods fall into the category of concept-based explanations and we discuss two of them extensively in §3.

3 Estimating Concept Effects with CEBaB

We now define the core metrics that we use to evaluate different explanation methods. Figure 1 provides a high-level view of the causal process we are envisioning. The process begins with an exogenous variable U representing a state of the world. For CEBaB, we can imagine that the value of U is a state of affairs u of a person evaluating a restaurant in a particular way. u contributes to a review variable X, with the value x of X mediated by u and by mediating concepts $C_1, \ldots C_k$, which correspond to the four aspect-level categories in CEBaB (food, service, ambiance, and noise), each of which can have values $c \in \{\text{positive}, \text{negative}, \text{unknown}\}$. The review x is processed by a model that outputs a vector of scores over classes (sentiment labels in CEBaB).

Core Metric Our central goal is to use CEBaB to evaluate explanation methods themselves. CEBaB supports many approaches to such evaluation. In this paper, we adopt an approach based on individual-level rather than average effects. This makes very rich use of the counterfactual text and associated labels provided by CEBaB. The starting point for this metric is the Individual Causal Concept Effect:

Definition 1 (Individual Causal Concept Effect; ICaCE)**.** *For a neural network $\mathcal{N}$ and feature function ϕ, the individual causal concept effect of changing the value of concept C from c to c' for*

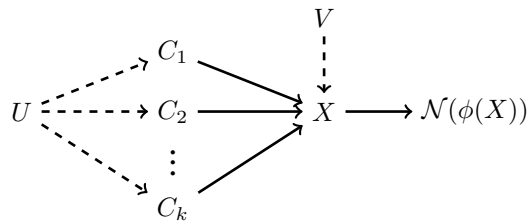


Figure 1: A causal graph describing a data generating process with an exogenous variables U and V representing the state of the world, mediating concepts $C_1, C_2 \ldots, C_k$, and data X that is featurized with ϕ. $\phi(X)$ is input to a classifier $\mathcal{N}$, which outputs a vector of scores over m output classes.

state of affairs u in an underlying data generation process $\mathcal{G}$ is

$$\textit{ICaCE}_{\mathcal{N}_\phi}(\mathcal{G}, x_u^{C=c}, c') = \mathbb{E}_{x\sim\mathcal{G}}\left[\mathcal{N}\big(\phi(x)\big) \mid do\begin{pmatrix} C = c' \\ U = u \end{pmatrix}\right] - \mathcal{N}\big(\phi(x_u^{C=c})\big). \tag{1}$$

ICaCE is a theoretical quantity. In practice, we use the Empirical Individual Causal Concept Effect.

Definition 2 (Empirical Individual Causal Concept Effect; $\widehat{\text{ICaCE}}$)**.** *For a neural network $\mathcal{N}$ and feature function ϕ, the empirical individual causal concept effect of changing the value of concept C from c to c' for state of affairs u is*

$$\widehat{\textit{ICaCE}}_{\mathcal{N}_\phi}(x_u^{C=c}, x_u^{C=c'}) = \mathcal{N}\big(\phi(x_u^{C=c'})\big) - \mathcal{N}\big(\phi(x_u^{C=c})\big), \tag{2}$$

where $(x_u^{C=c}, x_u^{C=c'})$ is a tuple of inputs originating from u with the concept C set to the values c and c', respectively.

The $\widehat{\text{ICaCE}}_{\mathcal{N}_\phi}$ for a pair of examples $(x_u^{C=c}, x_u^{C=c'})$ is simply the difference between the output score vectors for the two cases. With CEBaB, we can easily calculate these values because we have clusters of examples that are tied to the same reviewing situation u and express different concept values.

For assessing an explanation method $\mathcal{E}$, we compare ICaCE values with those returned by $\mathcal{E}$. Our core metric is the ICaCE-Error:

Definition 3 (ICaCE-Error)**.** *For a neural network $\mathcal{N}$, feature function ϕ and distance metric* Dist*, the ICaCE-Error of an explanation method $\mathcal{E}$ for changing the value of concept C from c to c' is:*

$$\textit{ICaCE-Error}^{\mathcal{D}}_{\mathcal{N}_\phi}(\mathcal{E}) = \frac{1}{|\mathcal{D}|}\sum_{(x_u^{C=c}, x_u^{C=c'})\in\mathcal{D}} \mathsf{Dist}\big(\widehat{\textit{ICaCE}}_{\mathcal{N}_\phi}(x_u^{C=c}, x_u^{C=c'}), \mathcal{E}_{\mathcal{N}_\phi}(x_u^{C=c}, c')\big) \tag{3}$$

We present results for three choices of Dist which vary in their ability to model the direction and magnitude of effects. These choices give subtly different but largely converging results, as detailed in Section 6 and reported more fully in Appendix D.

Aggregating Individual Causal Concept Effect It is often useful to also have a direct estimate of a model's ability to capture concept-level causal effects. For this, we employ an aggregating version of $\widehat{\text{ICaCE}}$, the Empirical Causal Concept Effect:

Definition 4 (Empirical Causal Concept Effect; $\widehat{\text{CaCE}}$)**.** *For a neural network $\mathcal{N}$ and feature function ϕ, the empirical causal concept effect of changing the value of concept C from c to c' in dataset $\mathcal{D}$ is*

$$\widehat{\textit{CaCE}}^{\mathcal{D}}_{\mathcal{N}_\phi}(C, c, c') = \frac{1}{|\mathcal{D}_C^{c\to c'}|}\sum_{(x_u^{C=c}, x_u^{C=c'})\in\mathcal{D}_C^{c\to c'}} \widehat{\textit{ICaCE}}_{\mathcal{N}_\phi}(x_u^{C=c}, x_u^{C=c'}). \tag{4}$$

This is an empirical estimator of the Causal Concept Effect (CaCE) of Goyal et al. [17]. It estimates, in general, how the classifier predictions change for a given concept and intervention direction.

Table 2: The evaluated explanation methods and their attributes. **Explainer Method** denotes the complexity of the models used by each explanation method. **Access to Explained Model** denotes the degree of access an explainer method needs to the explained model. **Concept Labels Needed** indicates whether a method estimating the effect for an input $x_u^{C=c}$ needs the actual input label c and/or the intervened value c' at test time. Models with a **Counterfactual Representation** approximate $\phi(x_u^{C=c'})$ to estimate the effect. Finally, only CausaLM and S-Learner have **Confounder Control** to minimize the impact of confounding concepts. *We predict these labels with a classifier.

Explanation method	Explainer Method	Access to Explained Model	Concept Labels Needed (test time)	Counterfactual Representation	Confounder Control
Approx	None	None	All concepts and their labels*	✗	✗
CONEXP [17]	None	None	c and c'	✗	✗
S-Learner [27]	Linear	None	All concepts and their labels*	✗	✓
TCAV [26]	Linear	Weights	None	✗	✗
ConceptSHAP [57]	Linear	Weights	None	✗	✗
INLP [40]	Linear	Weights	None	✓	✗
CausaLM [11]	Explained Model	Training Regime	None	✓	✓

Estimating Real-World Causal Effect of Aspect Sentiment on Overall Sentiment We can also estimate ground truth causal effects in CEBaB by simply using its labels directly. There are again a variety of ways that this could be done. We opt for the one that makes the richest use of the structures afforded by CEBaB. For perspicuity, in parallel to the neural network-based $\widehat{\text{ICaCE}}$ (Definition 2), we define the Empirical Individual Treatment Effect for our dataset:

Definition 5 (Empirical Individual Treatment Effects in CEBaB; $\widehat{\text{ITE}}$)**.** *The empirical individual treatment effect of changing the value of concept C from c to c' in CEBaB is*

$$\widehat{ITE}^{CEBaB}(x_u^{C=c}, x_u^{C=c'}) = f(x_u^{C=c'}) - f(x_u^{C=c}) \tag{5}$$

where f is a simple look-up procedure that retrieves the overall sentiment labels for CEBaB examples.

We aggregate over these values by taking their average, in parallel to what we do for network predictions (Definition 4). This yields the Empirical Average Treatment Effect ($\widehat{\text{ATE}}$) for CEBaB.

Alternative Metrics In Appendix A in our supplementary materials, we consider alternative formulations of the core metrics with *causal concept effects* and *absolute causal concept effects*, relating them to the different questions they engage with. We opt for the individual causal concept effect in our central metric (Definition 3), taking the central question to be what caused an ML model to produce an output for an *actual* input created from a real-world process.

4 Evaluated Explanation Methods

We compare several model explanation methods that share three main characteristics. First, they are all suitable for NLP models and have been used in the literature for generating model explanations in the form of estimated effects on model predictions. Second, they all provide concept-level explanations, for a pre-defined list of human-interpretable concepts (e.g., how sensitive a restaurant review rating classifier is to language related to food quality). This approach is also forward-looking, allowing more researchers to construct new hypotheses (i.e., concepts we have not collected labels for) and estimate their effect on the predictor. Third, all of the tested methods are model-agnostic, meaning that they separate the explanation from the model. At the same time, these methods differ in five important ways, as summarized Table 2.

We now turn to reviewing the explanation methods that we later compare on CEBaB (§6). In our mathematical formulas, we employ a unified notation for all methods, to make the definitions more accessible and easier to integrate into our experimental set-up. Assume we have a classifier $\mathcal{N}$ (which outputs a probability vector) and feature function ϕ, and we want to compute the effect on $\mathcal{N}_\phi(x_u^{C=c})$ of changing the value of concept C from c to c' using an unseen test set $(\mathcal{D}, Y)$.

Approximate Counterfactuals The gold labels of CEBaB are the difference between the logits for some original review $x_u^{C=c}$ and ground-truth counterfactual $x_u^{C=c'}$. As a baseline, we sample

an original review $x_{u'}^{C=c'}$ with the same aspect-labels as the $x_u^{C=c'}$ and use it as an approximate counterfactual:

$$\text{Approx}_{\mathcal{N}_\phi}(C, c, c'; x) = \mathcal{N}(\phi(x_{u'}^{C=c'})) - \mathcal{N}(\phi(x_u^{C=c})) \tag{6}$$

We do this sampling using predicted aspect labels from the aspect-level sentiment analysis models described in Appendix C.

Conditional Expectation (CONEXP) Goyal et al. [17] propose a baseline where the effect of a concept C is the average difference in predictions on examples with different values of C.

$$\text{CONEXP}_{\mathcal{N}_\phi}^{\mathcal{D}}(C, c, c') = \frac{1}{|\mathcal{D}^{C=c'}|} \sum_{x \in \mathcal{D}^{C=c'}} \mathcal{N}(\phi(x)) - \frac{1}{|\mathcal{D}^{C=c}|} \sum_{x \in \mathcal{D}^{C=c}} \mathcal{N}(\phi(x)) \tag{7}$$

where $\mathcal{D}^{C=c}$ and $\mathcal{D}^{C=c'}$ are subsets of $\mathcal{D}$ where C takes values c and c', respectively. To predict an effect, this method only relies on C, c, and c', resulting in an estimate that does not depend on the specific input text itself.

Conditional Expectation Learner (S-Learner) We adapt *S-Learner*, a popular method for estimating the Conditional Average Treatment Effect (CATE) [27]. To estimate causal concept effects, our *S-Learner* trains a logistic regression model $\mathcal{E}$ to predict $\mathcal{N}(\phi(x))$ using the values of all the labeled concepts of example x, denoted by x'.[2] Then, during inference, we compute an individual effect for example pair $(x_u^{C=c}, x_u^{C=c'})$ by comparing the output of the model $\mathcal{E}_x$ on this pair:

$$\textit{S-Learner}(C, c, c'; x) = \mathcal{E}({x'}_u^{C=c'}) - \mathcal{E}({x'}_u^{C=c}) \tag{8}$$

At inference time, S-Learner assumes access to all aspect-level labels x', which might not always be available. To alleviate this issue, we instead *predict* the aspect-level labels x' from the original text x using models described in Appendix C.

TCAV Kim et al. [26] use *Concept Activation Vectors* (CAVs), which are semantically meaningful directions in the embedding space of ϕ. Our adapted version of Testing with CAVs (TCAV) outputs a vector measuring the sensitivity of each output class k to changes towards the direction of a concept v_C at the point of the embedded input. It is computed as:

$$\text{TCAV}_{\mathcal{N}_\phi}(C; x) = \left(\nabla \mathcal{N}_k(\phi(x)) \cdot v_C\right)_{k=1}^{K} \tag{9}$$

where K is the number of classes and v_C is a linear separator learned to separate concept C in the embedding space of ϕ.

ConceptSHAP Yeh et al. [57] propose this expansion to SHAP [30], to generate concept-based explanation based on Shapley values [45]. Given a *complete* (i.e., such that the accuracy it achieves on a test set is higher than some threshold β) set of m concepts $\{C_1, \ldots, C_m\}$, ConceptSHAP calculates the contribution of each concept to the final prediction. Our adapted version outputs a vector for each $C \in \{C_1, \ldots, C_m\}$ and x. We justify this modification and provide implementation details in Appendix H.

CausaLM Feder et al. [11] estimate the causal effect of a binary concept C on the model's predictions by adding auxiliary adversarial tasks to the language representation model in order to learn a counterfactual representation $\phi_C^{\text{CF}}(x)$, while keeping essential information about potential confounders (control concepts). Their method outputs the text representation-based individual treatment effect (TReITE), which is computed as:

$$\text{TReITE}_{\mathcal{N}_\phi}(C; x) = \mathcal{N}'\big(\phi_C^{\text{CF}}(x)\big) - \mathcal{N}\big(\phi(x)\big) \tag{10}$$

where ϕ_C^{CF} denotes the learned counterfactual representation, where the information about concept C is not present, and $\mathcal{N}'$ is a classifier trained on this counterfactual representation. A key feature of CausaLM is its ability to control for confounding concepts (if modeled).[3] An inherent drawback of this technique is that it can only estimate interventions well for $c' = \text{Unknown}$, since the counterfactual representation is only trained to *remove* a concept C.

[2] This training approach, where an explainer model is fit to predict the output of the original model, shares the intuition of LIME, the widely used explanation method [41], but for concept-level effects.

[3] As in Feder et al. [11], we control for the most correlated potential confounder.

Table 3: Dataset statistics of CEBaB combining train/dev/test splits.

	Positive	Negative	Unknown	no maj.	Total
food	5726 (41%)	5526 (38%)	2605 (15%)	208 (31%)	14065
service	4045 (29%)	4098 (28%)	3877 (22%)	178 (27%)	12198
ambiance	2928 (21%)	2597 (18%)	5121 (29%)	203 (30%)	10849
noise	1365 (10%)	2215 (15%)	5883 (34%)	78 (12%)	9541

(a) Aspect-level labels.

1 star	1870 (12%)
2 star	3056 (20%)
3 star	3517 (23%)
4 star	2035 (13%)
5 star	2732 (18%)
no maj.	1879 (12%)

(b) Review-level ratings.

	{Neg, Pos}	{Neg, Unk}	{Pos, Unk}
food	898	1316	1291
service	851	857	938
ambiance	947	585	472
noise	1145	208	260

(c) Edit pair distribution. Edit pairs are examples that come from the same original source text and differ only in their rating for a particular aspect.

	Neg to Pos	Neg to Unk	Pos to Unk
food	1.84	1.37	−1.02
service	0.98	0.91	−0.53
ambiance	0.93	0.91	−0.50
noise	0.72	0.48	−0.47

(d) Empirical $\widehat{\text{ATE}}$ for the five-way sentiment labels in CEBaB. The reverse of a given concept change is the negative of the value given – e.g., the $\widehat{\text{ATE}}$ for 'Pos to Neg' for food is -1.84. See Appendix B for the corresponding values for binary sentiment.

Iterative Nullspace Projection (INLP) Ravfogel et al. [40] remove a concept from a representation vector by repeatedly training linear classifiers that aim to predict that attribute from the representations and projecting the learned representations on their null-space. Similar to CausaLM, INLP also estimates the TReATE (Equation 10) and can only estimate interventions for $c' =$ Unknown.

5 The CEBaB Dataset

Table 1 provides an intuitive overview of the structure of CEBaB. In the *editing* phase of dataset creation, crowdworkers modified an existing OpenTable review in an effort to achieve a specific aspect-level goal while holding all other properties of the original text constant. Our aspect-level categories are food, ambiance, service, and noise. In the *validation* phrase, crowdworkers labeled each example relative to each aspect as 'Positive', 'Negative', or 'Can't tell' (Unknown). Having five labels per example allows us to infer a majority label or reason in terms of the full label distributions. In the *rating* phase, each full text was labeled using a common five-star scale, again by five crowdworkers.

We began with 2,299 original reviews from OpenTable (related to 1,084 restaurants) and expanded them, via the above editing procedure, into a total of 15,089 texts. The distribution of normalized edit distances has peaks around 0.28 and 0.77, showing that workers made non-trivial changes to the originals, and even often had to make substantial changes to achieve the editing goal. (See Appendix B for the full distribution.)

Table 3 summarizes the resulting label distributions, where an example has label y if at least 3 of the 5 labelers chose y, otherwise it is in the 'no majority' category. 99% of aspect-level edits have a majority label that corresponds to the editing goal, and 88% of the texts have a review-level majority label on the five-star scale. Overall, these percentages show that workers were extremely successful in achieving their editing goals and that edits have systematic effects on overall sentiment.

The central goal of CEBaB is to create *edit pairs*: pairs of examples that come from the same original text and differ only in their labels for a particular aspect. For example, in Table 1, the first two 'food edit' cases form an edit pair, since they come from the same original text and differ only in their food label. Original texts can also contribute to edit pairs; the original text in Table 1 forms an edit pair with each of the texts it is related to by edits. Table 3c summarizes the distribution of edit pairs, and Table 3d reports the ground-truth $\widehat{\text{ATE}}$ values (§3).

Table 4: $\widehat{\text{CaCE}}$ (Definition 4) for `bert-base-uncased` fine-tuned as a 5-way sentiment classifier. Rows are concepts, columns are real-world concept interventions, and each entry indicates the average change in classifier output when the concept is intervened on with the given direction.[4] Results are averaged over 5 distinct seeds with standard deviations. The $\widehat{\text{CaCE}}$ value of changing concept C from c to c' is the negative $\widehat{\text{CaCE}}$ value of changing concept C from c' to c.

	Negative to Positive	Negative to unknown	Positive to unknown
food	1.90 (± 0.03)	1.00 (± 0.02)	−0.82 (± 0.01)
service	1.42 (± 0.04)	0.89 (± 0.04)	−0.45 (± 0.01)
ambiance	1.27 (± 0.01)	0.79 (± 0.01)	−0.50 (± 0.03)
noise	0.75 (± 0.02)	0.44 (± 0.00)	−0.23 (± 0.02)

We release the dataset with fixed train/dev/test splits. In creating these splits, we enforce two high-level constraints. The first is our 'grouped' requirement: for each original review t, all texts that are related to t via editing occur in the same split as t. This ensures that models are not evaluated on examples that are related by editing to those they have seen in training. Second, if any text t in a group received a 'no majority' label, then the entire group containing t is put in the train set. This ensures that there is no ambiguity about how to evaluate models on dev and test examples.

Once these high-level conditions were imposed, the examples were sampled randomly to create the splits. This allows that individual workers can contribute edited texts across splits. This minor compromise was necessary to ensure that we could have large dev and test splits. Appendix C in our supplementary materials shows that worker identity has negligible predictive power.

There are two versions of the train set: *inclusive* and *exclusive*. The inclusive train set contains all original and edited non-dev/test texts (11,728 texts). The exclusive version samples exactly one train text from each set of texts that are related by editing (1,755 examples). The rationale is that models trained with an original review as well as its edited counterparts may explicitly learn causal effects trivially by aggregating learning signals across inputs. Our exclusive train split prevents this, which helps facilitate fair comparisons between explanation methods and better resembles a real-world setting.

Our dataset is released publicly in JSON format and is available in the Hugging Face `datasets` library. It includes restaurant metadata, full rating distributions, and anonymized worker ids. Appendix B in our supplementary materials provides additional details on the dataset construction, including the prompts used by the crowdworkers, the number of workers per task, worker compensation, and a sample of examples with ratings to help convey the nature of workers' edits and the overall quality of the resulting texts and labels. In addition, Appendix C reports on a wide range of classifier experiments at the aspect-level and text-level that show that models perform well on CEBaB classification tasks, which bolsters the claim that CEBaB is a reliable tool for assessing explanation methods.

6 Experiments and Results

For each experiment, we fine-tune a pretrained language model to predict the overall sentiment of all restaurant reviews from our *exclusive* OpenTable train set. Since the goal of our work is not to achieve state-of-the-art performance, but rather to compare explanation methods and demonstrate the usage of CEBaB, we test the ability of methods to explain commonly used models, trained with standard experimental configurations.

In the main text, we report results for `bert-base-uncased` fine-tuned as a five-way classifier. Appendix D includes results for `GPT-2`, `RoBERTa`, and an `LSTM`, fine-tuned on binary, 3-way and 5-way versions of the sentiment task. All results, including the ground-truth effect that depends on the specific instance of a model, are averaged across 5 seeds.

To evaluate the intrinsic capacity of a model to capture causal effects, we report the $\widehat{\text{CaCE}}$ values, as in Definition 4. The results for `bert-base-uncased` are given in Table 4. They are intuitive and

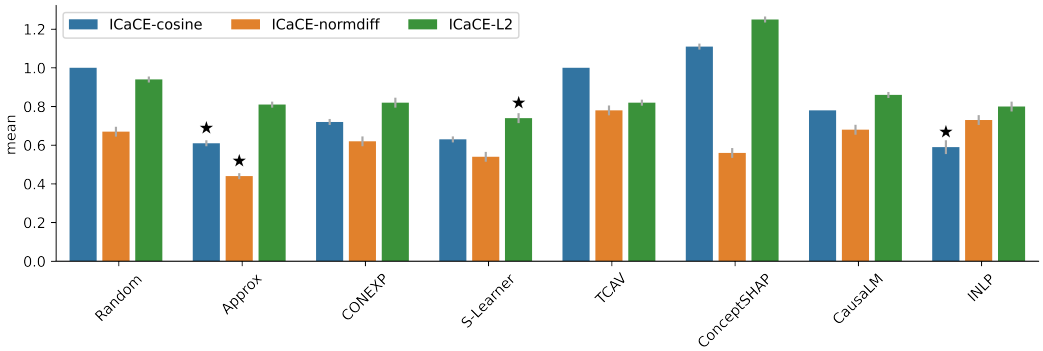


Figure 2: ICaCE-Error (Definition 3) for `bert-base-uncased` fine-tuned for five-way sentiment, averaged per aspect. We report values for cosine, L2, and normdiff. **Lower is better**. Stars mark the best result(s) per metric. Results averaged over 5 distinct seeds. [†]RandomExplainer takes the difference between two random probability vectors as the predicted effect.

well-aligned with the $\widehat{\text{ATE}}$ estimates in Table 3d, indicating that the model has captured the real-world effects.

Our primary assessment of the evaluation methods is given in Figure 2, again focusing on a five-way `bert-base-uncased` model as representative of our results. We provide values based on *cosine*, *L2*, and *normdiff* as the value of Dist in Definition 3. The *cosine*-distance metric measures if the estimated and observed effect have the same direction but does not take the magnitudes of the effects into account. The *L2*-distance measures the Euclidian norm of the difference of the observed and estimated effect. Both the direction and magnitude of the effects influence this metric. To only compare the magnitudes, we use the *normdiff*-distance, which computes the absolute difference between the Euclidean norms of the observed and estimated effects, thus completely ignoring the directions of both effects.

Remarkably, our approximate counterfactual baseline proves to be the best method at capturing both the direction and magnitude of the effects. The fact that a simple baseline method beats almost all other methods indicates that we need better explanation methods if we are going to capture even relatively simple causal effects like those given by CEBaB.

Recall from Table 2 that the compared methods require different levels of access to concept labels at inference time. Approximate counterfactuals and S-Learner have access to both the direction of the intervention and the predicted test-time aspect labels, enabling them to outperform CONEXP, which has access to only the direction of the intervention, and TCAV, ConceptSHAP, and CausaLM, which have access to neither the intervention direction nor test-time aspect labels.

The INLP method ties with the best method for the *cosine* metric, despite having access to neither intervention directions nor test-time aspect labels. Perhaps this method could be extended to make use of this additional information and decisively improve upon our approximate counterfactual baseline.

While CausaLM and INLP both estimate the effect of removing a concept from an input, INLP uses linear probes to guide interventions on the original model, while CausaLM trains an entirely new model with an auxiliary adversarial objective. The direct use of the original model is something INLP shares with the approximate counterfactual baseline; it seems that a tight connection to the original model may underlie success on CEBaB.

7 Conclusion

Our main contributions in this paper are twofold. First, we introduced CEBaB, the first benchmark dataset to support comparing different explanation methods against a single ground-truth with human-created counterfactual texts and multiply-validated concept labels for aspect-level and overall

[4]Definition 4 defines the CaCE values as vectors. In this table, we collapse the CaCE values to scalars by having $\mathcal{N}$ output the most probable predicted class, instead of the class distribution.

sentiment. Using this resource, one can isolate the true causal concept effect of aspect-level sentiment on any trained overall sentiment classifier. CEBaB provides a level playing field on which we can compare a variety of explanation methods that differ in their assumptions about their access to the model, their computational demands, their access to ground-truth concept labels at inference time, and their overall conception of the explanation problem. Furthermore, the evaluated methods make absolutely no use of CEBaB's counterfactual train set. In turn, we hope that CEBaB will facilitate the development of explanation methods that can take advantage of the very rich counterfactual structure CEBaB provides across all its splits.

Second, we have provided an in-depth experimental analysis of how well multiple model explanation methods are able to capture the true concept effect. A naive baseline that approximates counterfactuals through sampling achieves the best performance, with INLP and S-Learner being the only other methods that achieves state-of-the art on any metric. While CEBaB is only grounded in one task, sentiment analysis alone is enough to produce starkly negative results that should serve as a call to action for NLP researchers aiming to explain their models.

Acknowledgments and Disclosure of Funding

This research is supported in part by a grant from Meta AI. Karel D'Oosterlinck was supported through a doctoral fellowship from the Special Research Fund (BOF) of Ghent University. We thank our crowdworkers for their invaluable contributions to CEBaB.

References

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete Problems in AI Safety. *arXiv:1606.06565 [cs]*, 2016. URL `http://arxiv.org/abs/1606.06565`.

[2] Jasmijn Bastings, Yonatan Belinkov, Emmanuel Dupoux, Mario Giulianelli, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad. Proceedings of the fourth BlackboxNLP workshop on analyzing and interpreting neural networks for NLP. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2021.

[3] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71. Springer, 2016.

[4] Nitay Calderon, Eyal Ben-David, Amir Feder, and Roi Reichart. Docogen: Domain counterfactual generation for low resource domain adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7727–7746, 2022.

[5] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August 2019. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W19-4828`.

[6] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 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 (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL `https://www.aclweb.org/anthology/P18-1198`.

[7] 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.

[8] 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.

[9] Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.

[10] Amir Feder, Katherine A Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E Roberts, et al. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *arXiv preprint arXiv:2109.00725*, 2021.

[11] Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. CausaLM: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386, June 2021. doi: 10.1162/coli_a_00404. URL https://aclanthology.org/2021.cl-2.13.

[12] Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.144. URL https://aclanthology.org/2021.acl-long.144.

[13] Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. Causal Mediation Analysis for Interpreting Neural NLP: The Case of Gender Bias. *arXiv:2004.12265 [cs]*, November 2020. URL http://arxiv.org/abs/2004.12265. arXiv: 2004.12265.

[14] Atticus Geiger, Kyle Richardson, and Christopher Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.16. URL https://aclanthology.org/2020.blackboxnlp-1.16.

[15] Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[16] Bryce Goodman and Seth Flaxman. European Union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, October 2017. ISSN 2371-9621, 0738-4602. doi: 10.1609/aimag.v38i3.2741. URL http://arxiv.org/abs/1606.08813. arXiv: 1606.08813.

[17] Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining Classifiers with Causal Concept Effect (CaCE). *arXiv:1907.07165 [cs, stat]*, February 2020. URL http://arxiv.org/abs/1907.07165. arXiv: 1907.07165.

[18] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[19] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of Opportunity in Supervised Learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf.

[20] Peter Hase and Mohit Bansal. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? *arXiv preprint arXiv:2005.01831*, 2020.

[21] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.

[22] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR, 2020.

[23] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL `https://aclanthology.org/2020.acl-main.386`.

[24] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[25] Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online, June 2021. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2021.naacl-main.324`.

[26] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *International Conference on Machine Learning*, pages 2668–2677. PMLR, July 2018. URL `http://proceedings.mlr.press/v80/kim18d.html`. ISSN: 2640-3498.

[27] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

[28] Zachary C. Lipton. The mythos of model interpretability. *Communications of the ACM*, 61(10):36–43, September 2018. ISSN 0001-0782. doi: 10.1145/3233231. URL `https://doi.org/10.1145/3233231`.

[29] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*, July 2019. URL `http://arxiv.org/abs/1907.11692`. arXiv: 1907.11692.

[30] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 4768–4777, Red Hook, NY, USA, December 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4.

[31] Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

[32] Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1907367117. URL `https://www.pnas.org/content/117/48/30046`.

[33] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[34] Clemens Otte. Safe and Interpretable Machine Learning: A Methodological Review. In Christian Moewes and Andreas Nürnberger, editors, *Computational Intelligence in Intelligent Data Analysis*, Studies in Computational Intelligence, pages 111–122, Berlin, Heidelberg, 2013. Springer. ISBN 978-3-642-32378-2. doi: 10.1007/978-3-642-32378-2_8.

[35] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

[36] Judea Pearl. The limitations of opaque learning machines. *Possible minds: twenty-five ways of looking at AI*, pages 13–19, 2019.

[37] Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C Lipton, Graham Neubig, and William W Cohen. Evaluating explanations: How much do explanations from the teacher aid students? *Transactions of the Association for Computational Linguistics*, 10:359–375, 2022.

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

[39] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[40] Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection. *arXiv:2004.07667 [cs]*, April 2020. URL `http://arxiv.org/abs/2004.07667`. arXiv: 2004.07667.

[41] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "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 California USA, August 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939778. URL `https://dl.acm.org/doi/10.1145/2939672.2939778`.

[42] D. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66:688–701, 1974.

[43] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021.

[44] Naomi Saphra and Adam Lopez. Understanding learning dynamics of language models with SVCCA. 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 3257–3267, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1329. URL `https://www.aclweb.org/anthology/N19-1329`.

[45] Lloyd Shapley. A value for n-person games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, 1953.

[46] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

[47] Paul Soulos, R. Thomas McCoy, Tal Linzen, and Paul Smolensky. Discovering the compositional structure of vector representations with role learning networks. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 238–254, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.23. URL `https://aclanthology.org/2020.blackboxnlp-1.23`.

[48] Jost Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, 12 2014.

[49] Chi Sun, Luyao Huang, and Xipeng Qiu. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*, 2019.

[50] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org, 2017.

[51] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P19-1452`.

[52] Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 107–118, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.15. URL `https://aclanthology.org/2020.emnlp-demos.15`.

[53] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M. Shieber. Investigating gender bias in language models using causal mediation analysis. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/92650b2e92217715fe312e6fa7b90d82-Abstract.html`.

[54] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL `http://aclweb.org/anthology/W18-5446`.

[55] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[56] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[57] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33:20554–20565, 2020.

[58] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.

Checklist

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] Limitations that we found most significant are discussed in the supplementary material.

(c) Did you discuss any potential negative societal impacts of your work? [No] The work does not treat a specific task where we see an immediate risk for negative societal impact. Though the methods described in this paper might be used by practitioners, and we suggest doing so with care since the suggested methods need to be studied further in controlled settings before being used in such domains.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] The paper conforms with the provided guidelines.

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [N/A]

   (b) Did you include complete proofs of all theoretical results? [N/A]

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] Code and Data for reproducing all of our results are submitted as part of the supplementary material.

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5 and supplementary material.

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All results are averaged across 5 runs

   (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] See supplementary 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] See Section 3

   (b) Did you mention the license of the assets? [Yes] See supplementary material.

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] All assets we use are allowed for public use for the purposes of scientific research.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The data we use is taken from publicly available datasets, where all personally identifiable information have been removed.

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? [Yes] See Section 5 for a high-level description of the crowdsourced data and the supplementary material for the exact set of instructions given to crowd workers.

   (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? [Yes] See supplementary material for worker pay and costs breakdown.

Supplementary Materials

A Causal Concept Effects and Metrics for Explanation Methods

Data do not materialize out of thin air. Rather, data are generated from real-world processes with complex causal structures we do not observe directly. Causal inference is the task of estimating theoretical causal effect quantities.

When estimating causal effects, researchers commonly measure the *average treatment effect*, which is the difference in mean outcomes between the treatment and control groups [42]. Formally, we define the average treatment effect of binary treatment T on an outcome Y under a data generation process $\mathcal{G}$ that represents the unknown details of the real-world.

Definition 6 (Average Treatment Effect; ATE [42,35])**.**

$$ATE_T(Y, \mathcal{G}) = \mathbb{E}_{\mathcal{G}}\big[Y \,\big|\, do(T=1)\big] - \mathbb{E}_{\mathcal{G}}\big[Y \,\big|\, do(T=0)\big]. \tag{11}$$

The ATE is a theoretical quantity we cannot compute in practice, since we do not have access to $\mathcal{G}$ nor can we observe both interventions for the same subject.

However, we are concerned with estimating the causal effect of variables representing *non-binary concepts* in real-world systems, on data in an appropriate format for processing by a modern AI model that predicts *vector encoding probability distributions* over outputs.

Let $\mathcal{N}$ be a neural network outputting a probability vector, where its k-th entry represents the probability to predict the k-th class, and let ϕ be a feature representation (e.g., BERT embedding). In the context of model explanations, we will define the tools needed to answer three questions:

1. Given a real-world circumstance u that led to input data $x_u^{C=c}$, what is expected effect of a concept C changing from value c to value c' on the model output of $\mathcal{N}_\phi$ provided input data $x_u^{C=c}$?
2. What is the expected effect of a concept C changing from value c to value c' on the output of the model $\mathcal{N}_\phi$ provided input data X across real-world circumstances U?
3. What is the magnitude of the expected effect of a changing the concept C on the output of the model $\mathcal{N}_\phi$ provided input data X across real-world settings U?

For example, in the context of CEBaB, we might ask

1. Given a real-world dining experience u with good food quality ($C_{\text{food}} = +$) that led to a restaurant review $x_u^{C_{\text{food}}=+}$, what is the effect of changing the food quality C_{food} from $C_{\text{food}} = +$ to $C_{\text{food}} = -$ on the output of an overall-sentiment text classifier $\mathcal{N}_\phi$ provided a review of the dining experience?
2. What is the expected effect of changing the food quality C_{food} from positive $+$ to negative $-$ on the output of the model $\mathcal{N}_\phi$ across real-world dining experiences that lead to restaurant reviews?
3. What is the magnitude of the expected effect of a changing food quality C_{food} on the output of the model $\mathcal{N}_\phi$ across real-world dining experiences that lead to restaurant reviews?

Each of the above questions requires the estimation of a different theoretical quantity. In respect to the order of the questions, these quantities are the *individual causal concept effect*, the *causal concept effect*, and the *absolute causal concept effect*.

We believe the most practical question in explainable AI is: why does this model have this output behavior for an *actual* input. For this reason, our focus in the main text is *individual causal concept effects*. We define our central metric that captures the performance of an explainer on CEBaB as the average error on individual causal effect predictions (Definition 3).

We do not evaluate the ability of explainers to evaluate the causal concept effect or the absolute causal concept effect.

A.1 Theoretical Quantities

Definition 7 (Causal Concept Effects; [17]). *For an exogenous setting u that led to concept C taking on value c and the creation of input data $x_u^{C=c}$, the* individual causal concept effect *of a concept C changing from value c to c' in a data generation process $\mathcal{G}$ on a neural network $\mathcal{N}$ with feature representation ϕ is*

$$ICaCE_{\mathcal{N}_\phi}(\mathcal{G}, x_u^{C=c}, c') = \mathbb{E}_{x\sim\mathcal{G}}\left[\mathcal{N}\big(\phi(x)\big) \mid do\begin{pmatrix} C = c' \\ U = u \end{pmatrix}\right] - \mathcal{N}\big(\phi(x_u^{C=c})\big) \tag{12}$$

The causal concept effect *is the effect in general, meaning there is no input data generated from a fixed exogenous real-world setting:*

$$CaCE_{\mathcal{N}_\phi}(\mathcal{G}, C, c, c') = \mathbb{E}_{x\sim\mathcal{G}}\big[\mathcal{N}\big(\phi(x)\big) \mid do(C = c')\big] - \mathbb{E}_{x\sim\mathcal{G}}\big[\mathcal{N}\big(\phi(x)\big) \mid do(C = c)\big] \tag{13}$$

The absolute causal concept effect *estimate of the magnitude of the effect a concept has on a classifier output, regardless the concept values. We aggregate over all possible intervention values in the following way*

$$ACaCE_{\mathcal{N}_\phi}(\mathcal{G}, C) = \frac{1}{|\{\{c, c'\} \subseteq C\}|} \sum_{\{c,c'\}\subseteq C} \left|CaCE_{\mathcal{N}_\phi}(\mathcal{G}, C, c, c')\right|, \tag{14}$$

where C is the set of all possible values for concept in addition to denoting the concept itself.[5]

A.2 Empirical Estimates

Similar to the ATE, causal concept effects are theoretical quantities we can only estimate in reality. To perform such estimates, we need a dataset consisting of pairs $(x_u^c, x_u^{c'}) \in \mathcal{D}$ that are drawn from a data generation process $\mathcal{G}$. A major contribution of this work is crowdsourcing such a dataset, CEBaB. These pairs allow us to compute empirical estimations of (individual) causal concept effects.

Definition 8 (Empirical Causal Concept Effects). *For an exogenous setting u, the empirical individual causal concept effect of a concept C changed from value c to c', for $\mathcal{D}$ sampled from $\mathcal{G}$, on a neural network $\mathcal{N}$ trained on a feature representation ϕ is*

$$\widehat{ICaCE}_{\mathcal{N}_\phi}(x_u^{C=c'}, x_u^{C=c}) = \mathcal{N}\big(\phi(x_u^{C=c'})\big) - \mathcal{N}\big(\phi(x_u^{C=c})\big) \tag{15}$$

Given a full dataset $\mathcal{D}$ of such pairs, we can estimate the causal concept effect

$$\widehat{CaCE}_{\mathcal{N}_\phi}(\mathcal{D}, C, c, c') = \frac{1}{|\mathcal{D}_C^{c\to c'}|} \sum_{(x_u^c, x_u^{c'})\in\mathcal{D}} \widehat{ICaCE}_{\mathcal{N}_\phi}(x_u^{C=c}, x_u^{C=c'}) \tag{16}$$

And also the absolute causal concept effect

$$\widehat{ACaCE}_{\mathcal{N}_\phi}(\mathcal{D}) = \frac{1}{|\{\{c, c'\} \subseteq C\}|} \sum_{(c,c')\in C} |\widehat{CaCE}_{\mathcal{N}_\phi}(\mathcal{D}, C, c, c')| \tag{17}$$

Notice that the only difference between causal concept effects (Definition 7) and empirical causal concept effects (Definition 8) is that we change the expectation taken over $\mathcal{G}$ to be the average over a dataset $\mathcal{D} \sim \mathcal{G}$.

A.3 Explainer Errors

Given a dataset $\mathcal{D}$ and an explainer $\mathcal{E}_{\mathcal{N}_\phi}(x_u^c, c')$ that predicts individual causal concept effects $ICACE_{\mathcal{N}_\phi}(x_u^c, c')$, we define metrics capturing the ability of $\mathcal{E}$ to estimate causal effects by simple computing the averaged distance between our explainer and the empirical causal effect

[5] We take the absolute value since $\text{CaCE}_{\mathcal{N}_\phi}(\mathcal{G}, C, c, c') = -\text{CaCE}_{\mathcal{N}_\phi}(\mathcal{G}, C, c', c)$, and these cancel each other in the summation.

Definition 9 (Explainer Distances)**.** *The average distance between the explainer and the empirical individual causal concept effects.*

$$\textit{ICaCE-Error}^{\mathcal{D}}_{\mathcal{N}_\phi}(\mathcal{E}, C, c, c') = \frac{1}{\left|\mathcal{D}_C^{c\to c'}\right|} \sum_{(x_u^{C=c}, x_u^{C=c'}) \in \mathcal{D}_C^{c\to c'}} \mathsf{Dist}\big(\widehat{\textit{ICaCE}}_{\mathcal{N}_\phi}(x_u^{C=c}, x_u^{C=c'}), \mathcal{E}_{\mathcal{N}_\phi}(x_u^{C=c}, x_u^{C=c'})\big) \quad (18)$$

The distance between the average of explainer outputs and the empirical causal concept effect

$$\textit{CaCE-Error}^{\mathcal{D}}_{\mathcal{N}_\phi}(\mathcal{E}, C, c, c') = \|\widehat{\textit{CaCE}}_{\mathcal{N}_\phi}(\mathcal{D}, C, c, c'), \frac{1}{\left|\mathcal{D}_C^{c\to c'}\right|} \sum_{x_u^c, x_u^{c'} \in \mathcal{D}_C^{c\to c'}} \mathcal{E}_{\mathcal{N}_\phi}(x_u^c, c'))\| \quad (19)$$

The distance between the average magnitude of explainer outputs and the empirical absolute causal effect

$$\textit{ACaCE-Error}^{\mathcal{D}}_{\mathcal{N}_\phi}(\mathcal{E}, C) = \|\widehat{\textit{ACaCE}}_{\mathcal{N}_\phi}(\mathcal{D}, C), \frac{1}{|\{\{c, c'\} \subseteq C\}|} \sum_{(c,c') \in C} \frac{1}{\left|\mathcal{D}_C^{c\to c'}\right|} \sum_{x_u^c, x_u^{c'} \in \mathcal{D}_C^{c\to c'}} |\mathcal{E}_{\mathcal{N}_\phi}(x_u^c, c'))|\| \quad (20)$$

where $\|\cdot\|$ is some distance metric and $\mathcal{D}_C$ is the subset of data where C is the concept changed and $\mathcal{D}_C^{c\to c'}$ is the subset of data where C is the concept changed from value c to value c'.

In the main text, we use the ICaCE-Error as our primary evaluation metric.

B CEBaB

Our supplementary materials contain a full Datasheet for CEBaB as a separate markdown document.

B.1 Restaurant-level metadata from OpenTable

Table 5 gives an overview of the metadata associated with the original review texts in CEBaB.

Table 5: CEBaB metadata from OpenTable, tabulated at the level of individual original reviews. A total of 1,084 restaurants are represented in the data.

italian	1076
american	654
french	254
seafood	202
mediterranean	113

(a) Cuisine.

northeast	863
west	634
south	470
midwest	332

(b) U.S. regions.

1 star	244
2 star	1207
3 star	123
4 star	330
5 star	395

(c) Star ratings.

B.2 Crowdworkers

A total of 254 workers participated in our experiments. All of them come from a pool of workers whom we prequalified to participate in our tasks based on the work they did for us on previous crowdsourcing projects. Thus, we expected that they would do high quality work, and they more than lived up to our expectations, as indicated by the high degree of success they achieved when editing and the high degree of consensus they reached about how to label examples.

There are a total of 642 instances of 15,0006 for which, despite our best efforts, a worker validated an example that they themselves created during the editing phase. Removing the contributions of these workers affects the majority in only 24 cases, with no clear pattern to the changes, so we kept all the validation labels in order to ensure that every example has give responses.

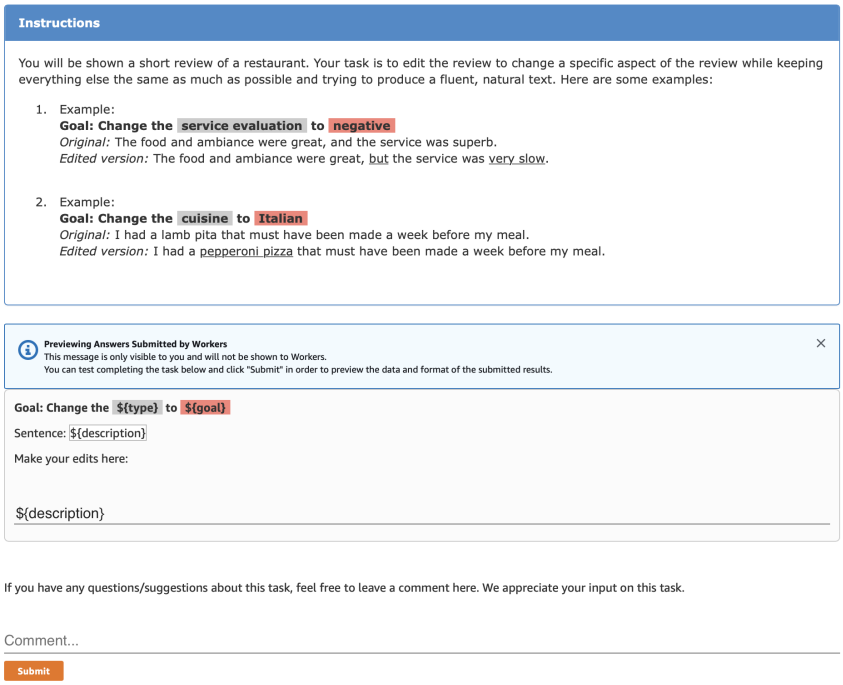


Figure 3: Edit phase annotation interface where the task was to convey 'Positive' or 'Negative' for the target aspect.

B.3 Editing Phase

A total of 183 workers participated in this phase. Workers were paid US$0.25 per example. Figure 3 shows the annotation interface that workers used when changing the target aspect's sentiment to either 'Positive' or 'Negative', and Figure 4 shows the interface where the task was to hide the target aspect's sentiment.

Figure 5 summarizes the distribution of edit distances between original and edited texts. These distances are calculated at the character-level and normalized by the length of the original or review, whichever is longer.

B.4 Validation Phase

A total of 174 workers participated in this phase. Workers were paid US$0.35 per batch of 10 examples. Figure 6 shows the annotation interface that workers used.

B.5 Review-level Rating Phase

A total of 155 workers participated in this phase. Workers were paid US$0.35 per batch of 10 examples. Figure 7 shows the annotation interface that workers used.

B.6 Randomly Selected Examples

Table 6 provides a random sample of edit pairs from CEBaB's dev set.

B.7 Five-way Empirical ATE for CEBaB

Table 7 provides the binary $\widehat{\text{ATE}}$ values for CEBaB. These can be compared with the corresponding five-way values in Table 3d in the main text.

Table 6: Randomly sampled edit pairs from CEBaB.

description	original?	aspect	edit goal	aspect labels	aspect maj.	review labels	review maj.
Food was disgusting and very unreasonable!!!!!! Every request was honored and very friendly staff.\nHomemade bread which was foul.....	False	food	–	–, –, –, –, –	–	2, 2, 2, 2, 2	2
Every request was honored and very friendly staff.	False	food	unk.	unk, unk, unk, +, +	unk.	5, 5, 5, 4, 4	5
The food was average, but the service was terrible.	True	food	None	–, –, –, unk, +	–	2, 2, 2, 3, 3	2
The food was above average, but the service was terrible.	False	food	+	+, +, +, +, +	+	3, 3, 3, 3, 2	3
We hated our afternoon at Shorebreak!	False	ambiance	–	–, –, –, unk, unk	–	1, 1, 1, 1, 1	1
We loved our afternoon at Shorebreak!	False	ambiance	unk.	unk, unk, unk, –, +	unk.	5, 5, 5, 4, 4	5
The Sunday Jazz Brunch is great - Good music and fine, creative food. The service was great, my server answered all of my questions. The ambiance is quiet, but not so quiet as to inhibit conversation. A wonderful way to spend an early Sunday afternoon.	False	service	+	+, +, +, +, +	+	5, 5, 5, 4, 4	5
The Sunday Jazz Brunch is great - Good music and fine, creative food. The ambiance is quite, but not so quite as to inhibit conversation. A wonderful way to spend an early Sunday afternoon. The only bad spot was the horrid service.	False	service	–	–, –, –, –, –	–	4, 4, 4, 4, 3	4
My pasta dish was flavorless and rubbery and my husband's was cold. At least it 45 minutes to get it. Very poor, indeed.	True	food	None	–, –, –, –, –	–	1, 1, 1, 2, 2	1
My pasta dish was amazing and cooked great. At least it 45 minutes to get it. Very poor, indeed.	False	food	+	+, +, +, +, –	+	3, 3, 3, 3, 1	3
liked the restaurant a lot and loved the meal. Found the chicken great!	False	food	+	+, +, +, +, +	+	5, 5, 5, 4, 3	5
I liked the restaurant a lot,	False	food	unk.	unk, unk, unk, unk, +	unk.	5, 5, 5, 4, 4	5
At the heart of it, this is a HOTEL restaurant.	True	noise	None	unk, unk, unk, unk, unk	unk.	3, 3, 3, 3, 2	3
At the heart of it, this is an extremely loud restaurant.	False	noise	–	–, –, –, –, –	–	1, 1, 1, 3, 2	1
I was expecting some dishes from the Northern Italian Cuisine. The menu was not distinguishable from any other chain. The food was good but no differentiation. It was noisy, but I believe by design.	True	food	None	+, +, +, +, +	+	3, 3, 3, 4, 2	3
I was expecting some dishes from the Northern Italian Cuisine. The menu was not distinguishable from any other chain. The food was even worse than that. It was also noisy, but I believe by design.	False	food	–	–, –, –, –, +	–	1, 1, 1, 2, 2	1

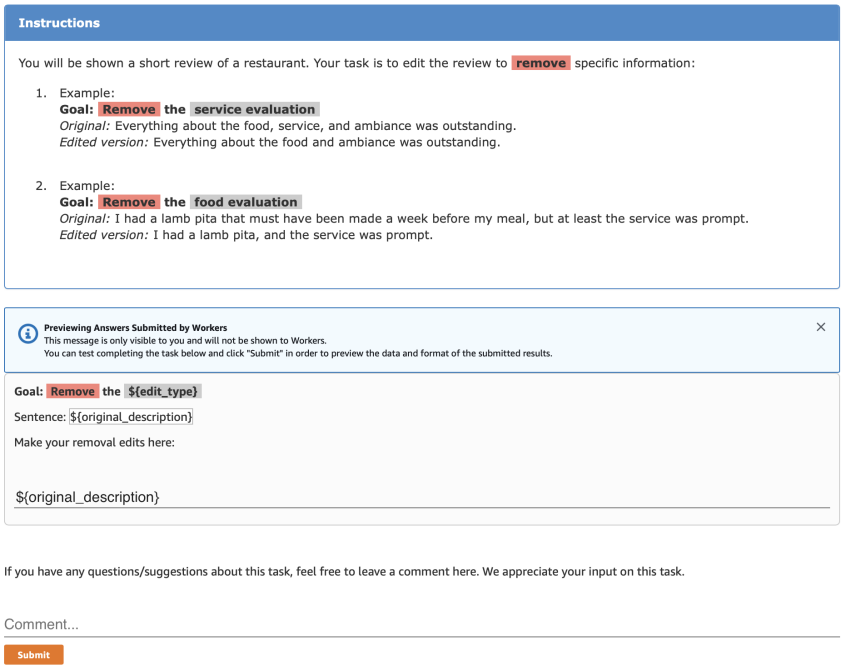


Figure 4: Edit phase annotation interface where the task was to hide the sentiment of the target aspect.

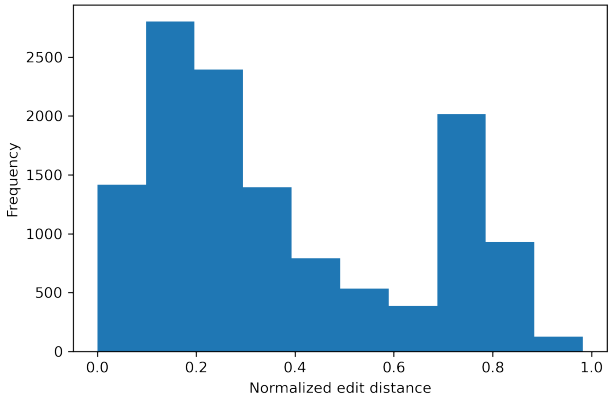


Figure 5: Normalized edit distances between original texts and those created during the editing phase for CEBaB.

B.8 Edit variability

In the editing phase we ask human annotators to produce edits of an original review with regard to some concept. This is inherently a noisy process, which may impact the quality of our final benchmark. The CEBaB dataset features a modest set of paired edits (176 pairs in total). Each of these pairs contains two edits, starting from the same original sentence and edit goal, which results in two different edited sentences. Like all sentences in CEBaB, these edits were labeled for their review score by human annotators.

Figure 8a shows the distribution of the difference in final review majorities produces by these paired edits. Most paired edits differ at most by one star in their final majority rating, indicating that in general there is some noise associated with the editing procedure, but this does not have a major impact on the final review score. Figure 8b shows the same distribution when we consider the average

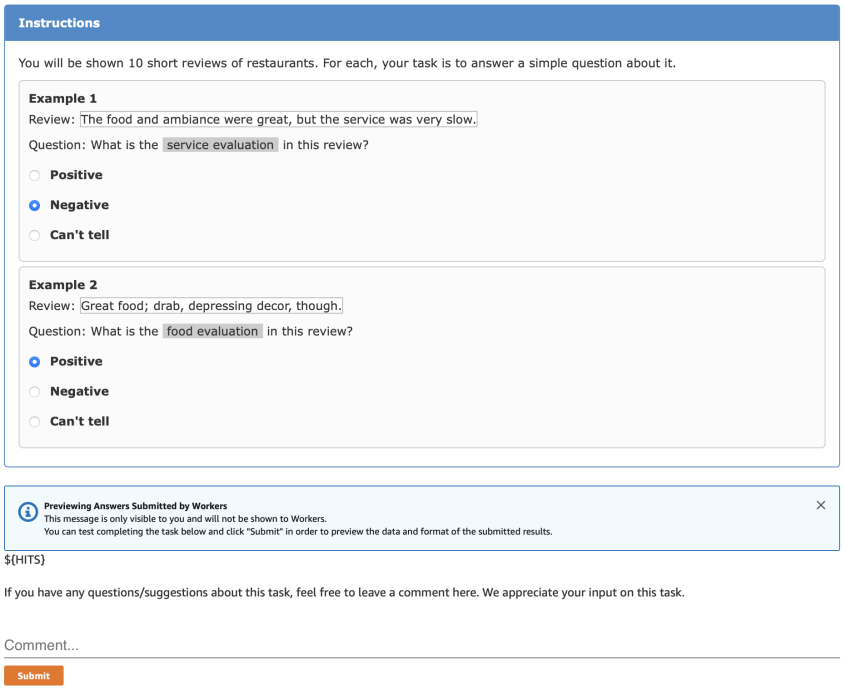


Figure 6: Validation phase annotation interface.

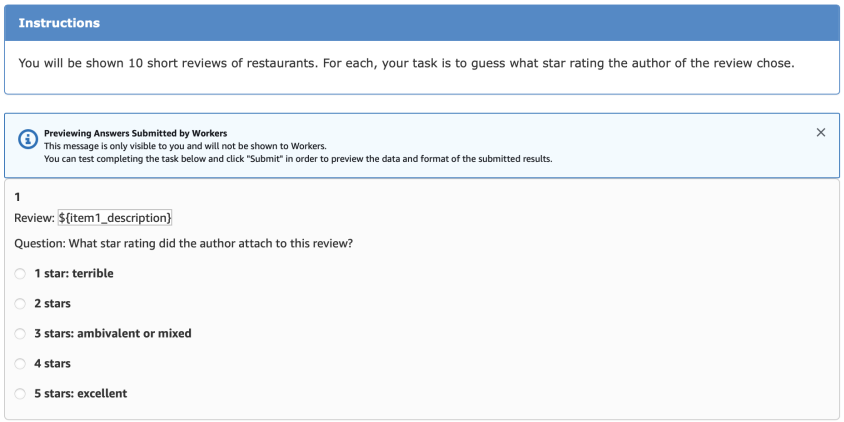
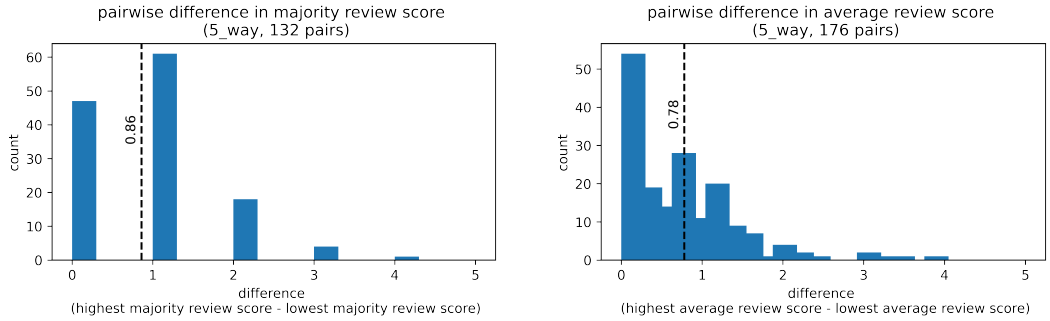


Figure 7: Review-level annotation interface.

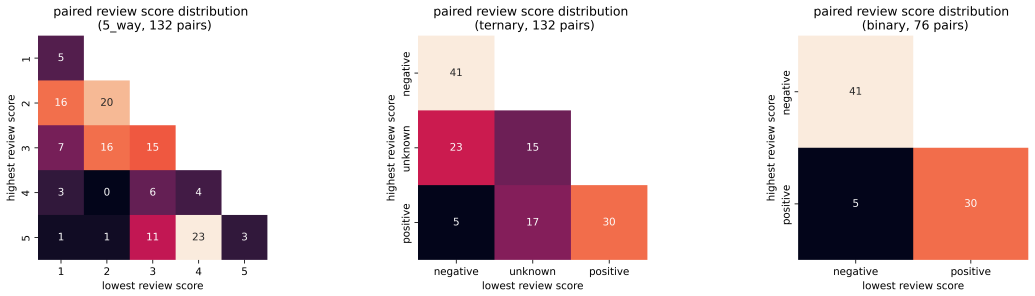
Table 7: Empirical $\widehat{\text{ATE}}$ for the binary sentiment labels in CEBaB. Reversing concept order results in the negation of the value given.

	Neg to Pos	Neg to Unk	Pos to Unk
food	0.77	0.49	−0.41
service	0.25	0.20	−0.16
ambiance	0.14	0.18	−0.14
noise	0.08	0.04	−0.14



(a) (b)

Figure 8: Pairwise absolute difference in majority (a) and average (b) review score for all double edits. Figure (a) only considers the 132 pairs where both edits have an actual review majority. Figure (b) considers all 176 pairs. Averages of the distributions are shown with a dotted vertical line.



(a) (b) (c)

Figure 9: Pairwise review majority distribution for all double edits in 5-way (a), ternary (b), and binary (c) classification settings. Figures (a) and (b) consider only the 132 pairs where both edits have an actual review majority. Figure (c) considers the 76 pairs that have both a review majority and non-neutral labels.

review score an edit received, as opposed to the majority score. If we consider these average scores, most of the paired edits differ only slightly in their resulting review score.

Figures 9a-c shows the distribution of this pairwise review score in more detail. In an idealized setting without variability, the distribution would be centered around the diagonal of the heatmap. When going from 5-way classification to ternary and binary classification, the variability introduced by the edits becomes less relevant with regard to the final review majority label.

C CEBaB Modeling Experiments

This section reports on standard classifier-based experiments with CEBaB, aimed at providing a sense for the dataset when it is used as a standard supervised sentiment dataset. We report experiments on the aspect-level and review-level ratings. In addition, we present evidence that author identity does not have predictive value.

C.1 Experiments Set-up

We rely on the Hugging Face `transformers` library.[6] [56] We train our models with 4 Nvidia 2080 Ti RTX 11GB GPUs on a single node machine. We use a maximum sequence length of 128 with a fix batch size of 32 with a initial learning rate of $2e^{-5}$. We run each experiment 5 times with distinct random seeds. We train our models with a minimum epoch number of 5 with our largest training set. We linearly scale our training epoch number by the size of the training set. We skip hyperparameter

[6] `https://github.com/huggingface/transformers`

Table 8: Model performance results for sequence classification as well as aspect-based sentiment analysis (ABSA) under 3 training conditions. Mean Macro-F1 scores across 5 runs with distinct random seeds are reported.

	Exclusive				Inclusive			
Model	Binary	Ternary	5-way	ABSA	Binary	Ternary	5-way	ABSA
				dev split				
BERT	0.97	0.82	0.68	0.88	0.98	0.85	0.72	0.90
GPT-2	0.97	0.80	0.67	0.88	0.98	0.84	0.70	0.89
LSTM	0.94	0.75	0.59	0.83	0.96	0.82	0.68	0.87
RoBERTa	0.99	0.83	0.71	0.89	0.99	0.86	0.76	0.90
				test split				
BERT	0.97	0.82	0.70	0.87	0.98	0.84	0.73	0.89
GPT-2	0.97	0.80	0.65	0.87	0.97	0.83	0.68	0.89
LSTM	0.94	0.75	0.60	0.82	0.96	0.81	0.68	0.87
RoBERTa	0.98	0.83	0.70	0.88	0.99	0.86	0.75	0.90

tuning for optimized task performance as our goal for this paper is to evaluate explanation methods. We release all of our models on Huggingface Dataset Hub.

C.2 Models

We include 4 different types of models, including BERT (`bert-base-uncased`) [7], RoBERTa (`roberta-base`) [29], GPT-2 (`gpt2`) [38], as well as LSTM with dot-attention [31]. Our LSTM model uses `bert-base-uncased` tokenizer for simplicity. We initialize the embeddings of tokens for our LSTM using fastText [24]. We reconfigure the classification head all other models the same classification head as in RoBERTa as a non-linear multilayer perceptron (MLP).[7]

C.3 Multi-class Sentiment Analysis Benchmark

We report model performance results under 3 training conditions: **Binary Classification**, where we label reviews with 1 star and 2 star ratings as negative, reviews with 4 star and 5 star as positive, and 3-star reviews are dropped; **Ternary Classification**, where we add another neutral class for reviews with 3 star ratings; and **5-way Classification**, where each star rating by itself is considered as a class. We leave out reviews in the train set in the 'no majority' category. (Dev and Test do not contain any such examples.) Table 8 shows the performance results for our models under different conditions. Our results suggest that RoBERTa has the edge over others across all evaluated tasks.

C.4 Aspect-based Sentiment Analysis Benchmark

Our dataset can be naturally used as an aspect-based sentiment analysis (ABSA) benchmark. For each sentence, it may contain up to 4 aspects with respect to the reviewing restaurant. As ABSA benchmarks are usually small and sparse with missing labels, our dataset provides validated aspect-based labels, and is one of the largest human validated ABSA benchmark.

To evaluate model performance, we adapt standard finetuning approach for ABSA benchmarks as proposed by [49]. Instead of single sentence classification, we add another auxiliary sentence representing the aspect. For instance, to predict the label for the 'food' aspect for "the food here is good but not the service", we append a single aspect token with a separator, and construct our input sentence as "the food here is good but not the service [SEP] food". Table 8 shows the performance results for our models under different conditions.

[7] We implemented T5 (`t5-base`; [39]) as a text-to-text model with the goal of treating predicted tokens as class labels. However, this raised unanticipated implementation questions concerning how to post-process multi-token class labels (e.g., "very positive") for use in our explainer methods. As a result, we have elected to leave the T5 results out of the current draft, but we intend to include them in the next version once they have been more thoroughly vetted.

Table 9: Model performance on top-k author identity prediction with number of train and dev examples.

Model	Accuracy	Macro-F1	# train	# dev
Random (k=5)	0.16	0.15	1105	227
Random (k=10)	0.10	0.10	2072	519
Random (k=15)	0.07	0.07	2963	741
RoBERTa (k=5)	0.27	0.16	1105	227
RoBERTa (k=10)	0.14	0.05	2072	519
RoBERTa (k=15)	0.11	0.04	2963	741

C.5 Author Identity Prediction

One potential artifact of our benchmark is edited sentence may expose author identity, which may result in artifact in interpreting model performance. To quantify this potential artifact, we train models to predict author identities based on the sentences. We create author identity prediction dataset by aggregating our dataset by anonymized worker ids. We then split the dataset into train/dev with a 4-to-1 ratio. For model training, we finetune RoBERTa for 5 epochs with a batch size of 32, a learning rate of $2e^{-5}$, and a maximum sequence length of 128. Note that we only consider top-k annotators ranked by their contributions (i.e., number of examples in our dataset). Table 9 shows the performance results of our finetuned models with a random classifier. Our results suggest that potential artifacts may exist but only for a limited extend.

D Additional Results

In this section, we report additional results for bert-base-uncased, roberta-base, gpt-2, and an LSTM, fine-tuned on binary, ternary and 5-way versions of the sentiment task. These models are described in Appendix C. Table 10 summarizes all the results.

We refer to the results section in the main text for an explanation of the different metrics considered. Which metric is best depends on the final use-case and whether it is more important to estimate the direction or the magnitude of the effect.

ICaCE-cosine Figure 10 shows the results for the ICaCE-Error with the *cosine* distance metric. The explanation methods that take the direction of the intervention into account (Approx, CONEXP, S-Learner) are the clear winners across all different models considered. S-Learner marginally wins across the most settings, but the conceptually simple Approx baseline is a close second. The strong performance of this simple baseline across the board suggests that most methods perform subpar, and that there is potential value in developing better concept-based model explanation methods.

Both TCAV and ConceptSHAP struggle to achieve better-than-random performance across all settings. Further analysis is needed to exactly understand why these methods are struggling.

Some additional trends emerge that require more analysis to fully understand. For example, Approx generally increases in performance when evaluated on more fine-grained classification settings, while CONEXP is typically worse here.

ICaCE-normdiff Figure 11 shows the results for the ICaCE-Error with the *normdiff* distance metric. In general, it is more difficult for explanation methods to estimate the magnitude of the intervention effect when the task increases in complexity. For a given explanation method and model, best results are often achieved for the binary classification problem.

The conceptually simple Approx baseline wins across the board. S-Learner is only able to match its performance a few times. While previous results already showed that most of the methods fall behind the Approx baseline, the results are particularly striking for this metric.

While S-learner and CONEXP were somewhat comparable on the *cosine* metric, their differences become clear on the normdiff metric: S-Learner is better at estimating the magnitude of the intervention.

An interesting trend can be observed for TCAV, which has good performance on the binary task but becomes worse than random when evaluated on the ternary and 5-way settings. ConceptSHAP is the only method that consistently breaks the upward trend when going from ternary to the 5-way setting. More analysis is needed to understand both these phenomena.

ICaCE-L2 Figure 12 shows the results for the ICaCE-Error with the *L2* distance metric. Because this metric takes both the scale and direction of the effect into account, it is slightly harder to interpret. In general, the performance drops when evaluated on more fine-grained classification settings.

Again, the Approx baseline is a strong contestant, but on this metric the results are more varied. S-Learner is consistently the best at producing the closest explanation in Euclidian distance to the real effect for the 5-way setting.

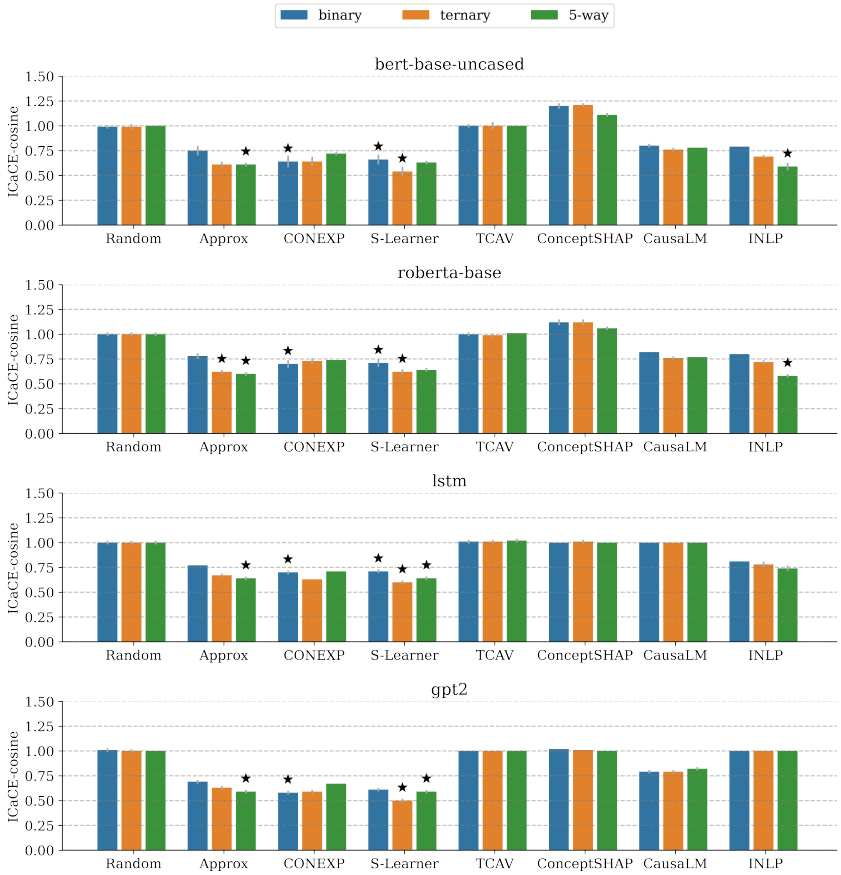


Figure 10: ICaCE-Error for all experiments using the *cosine* distance metric. **Lower is better**. Results averaged over 5 distinct seeds. Error bars (in gray) display the standard deviation. Stars denote the best results for a given classification setting.

E CausaLM

E.1 Our adaptation

The CausaLM algorithm was originally designed to estimate the average treatment effect of a high-level concept on pre-trained language models. Its output estimator is the textual representation averaged treatment effect (TReATE), which is computed as:

$$\text{TReATE}_{\mathcal{N}_\phi}(C;\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}} \mathcal{N}'\big(\phi_C^{\text{CF}}(x)\big) - \mathcal{N}\big(\phi(x)\big), \tag{21}$$

Table 10: ICaCE scores on the test set for the binary, ternary and 5-way classification settings. **Lower is better**. Results averaged over 5 distinct seeds; standard deviations in parentheses.

(a) ICaCE scores for 5-way sentiment classification setting.

Model	Metric	Random	Approx	CONEXP	S-Learner	TCAV	ConceptSHAP	CausaLM	INLP
BERT	$L2_{ICaCE}$	0.94 (.01)	0.81 (.01)	0.82 (.02)	0.74 (.02)	0.82 (.01)	1.25 (.01)	0.86 (.01)	0.80 (.02)
	COS_{ICaCE}	1.00 (.00)	0.61 (.01)	0.72 (.01)	0.63 (.01)	1.00 (.00)	1.11 (.01)	0.78 (.00)	0.59 (.03)
	$NormDiff_{ICaCE}$	0.67 (.02)	0.44 (.01)	0.62 (.02)	0.54 (.02)	0.78 (.02)	0.56 (.02)	0.68 (.02)	0.73 (.02)
RoBERTa	$L2_{ICaCE}$	0.97 (.01)	0.83 (.01)	0.86 (.01)	0.78 (.01)	0.85 (.01)	1.24 (.01)	0.90 (.01)	0.84 (.01)
	COS_{ICaCE}	1.00 (.01)	0.60 (.01)	0.74 (.00)	0.64 (.01)	1.01 (.00)	1.06 (.01)	0.77 (.00)	0.58 (.01)
	$NormDiff_{ICaCE}$	0.72 (.01)	0.45 (.00)	0.67 (.01)	0.59 (.01)	0.83 (.01)	0.61 (.00)	0.74 (.00)	0.81 (.01)
GPT-2	$L2_{ICaCE}$	0.81 (.02)	0.72 (.02)	0.68 (.02)	0.60 (.02)	0.68 (.02)	1.03 (.02)	0.76 (.02)	0.72 (.01)
	COS_{ICaCE}	1.00 (.00)	0.59 (.01)	0.67 (.00)	0.59 (.01)	1.00 (.00)	1.00 (.00)	0.82 (.01)	1.00 (.00)
	$NormDiff_{ICaCE}$	0.52 (.02)	0.41 (.01)	0.47 (.02)	0.40 (.01)	0.65 (.02)	0.46 (.01)	0.52 (.02)	0.58 (.03)
LSTM	$L2_{ICaCE}$	0.89 (.01)	0.86 (.01)	0.79 (.01)	0.73 (.01)	0.78 (.02)	1.27 (.04)	0.76 (.01)	0.79 (.01)
	COS_{ICaCE}	1.00 (.01)	0.64 (.01)	0.71 (.00)	0.64 (.01)	1.02 (.01)	1.00 (.00)	1.00 (.00)	0.74 (.02)
	$NormDiff_{ICaCE}$	0.62 (.01)	0.50 (.01)	0.59 (.01)	0.53 (.01)	0.70 (.01)	0.54 (.00)	0.76 (.01)	0.60 (.01)

(b) ICaCE scores for ternary sentiment classification setting.

Model	Metric	Random	Approx	CONEXP	S-Learner	TCAV	ConceptSHAP	CausaLM	INLP
BERT	$L2_{ICaCE}$	0.79 (.01)	0.54 (.01)	0.65 (.00)	0.56 (.00)	0.56 (.00)	0.94 (.01)	0.72 (.00)	0.58 (.01)
	COS_{ICaCE}	0.99 (.02)	0.61 (.02)	0.64 (.04)	0.54 (.04)	1.00 (.03)	1.21 (.01)	0.76 (.01)	0.69 (.01)
	$NormDiff_{ICaCE}$	0.60 (.00)	0.42 (.01)	0.54 (.00)	0.48 (.00)	0.55 (.00)	0.62 (.01)	0.62 (.00)	0.55 (.01)
RoBERTa	$L2_{ICaCE}$	0.79 (.01)	0.56 (.00)	0.65 (.01)	0.57 (.01)	0.55 (.01)	0.88 (.02)	0.74 (.01)	0.55 (.01)
	COS_{ICaCE}	1.00 (.01)	0.62 (.01)	0.73 (.02)	0.62 (.02)	0.99 (.01)	1.12 (.02)	0.76 (.01)	0.72 (.01)
	$NormDiff_{ICaCE}$	0.61 (.01)	0.43 (.00)	0.54 (.00)	0.48 (.00)	0.54 (.00)	0.61 (.01)	0.66 (.01)	0.54 (.01)
GPT-2	$L2_{ICaCE}$	0.75 (.01)	0.57 (.01)	0.60 (.01)	0.52 (.01)	0.52 (.01)	0.69 (.01)	0.68 (.01)	0.61 (.03)
	COS_{ICaCE}	1.00 (.01)	0.63 (.01)	0.59 (.01)	0.50 (.01)	1.00 (.00)	1.01 (.00)	0.79 (.01)	1.00 (.00)
	$NormDiff_{ICaCE}$	0.54 (.01)	0.42 (.01)	0.47 (.01)	0.42 (.01)	0.51 (.01)	0.52 (.01)	0.55 (.01)	0.51 (.01)
LSTM	$L2_{ICaCE}$	0.76 (.00)	0.58 (.01)	0.63 (.01)	0.55 (.01)	0.55 (.01)	1.03 (.04)	0.53 (.01)	0.68 (.01)
	COS_{ICaCE}	1.00 (.01)	0.67 (.01)	0.63 (.00)	0.60 (.01)	1.01 (.01)	1.01 (.01)	1.00 (.00)	0.78 (.02)
	$NormDiff_{ICaCE}$	0.56 (.01)	0.45 (.01)	0.51 (.00)	0.46 (.01)	0.51 (.01)	0.65 (.01)	0.52 (.01)	0.56 (.01)

(c) ICaCE scores for binary sentiment classification setting.

Model	Metric	Random	Approx	CONEXP	S-Learner	TCAV	ConceptSHAP	CausaLM	INLP
BERT	$L2_{ICaCE}$	0.60 (.01)	0.19 (.01)	0.51 (.00)	0.31 (.00)	0.31 (.01)	0.76 (.06)	0.57 (.01)	0.51 (.05)
	COS_{ICaCE}	0.99 (.01)	0.75 (.04)	0.64 (.05)	0.66 (.04)	1.00 (.01)	1.20 (.02)	0.80 (.01)	0.79 (.00)
	$NormDiff_{ICaCE}$	0.52 (.01)	0.19 (.01)	0.50 (.00)	0.30 (.00)	0.30 (.01)	0.55 (.05)	0.56 (.01)	0.50 (.04)
RoBERTa	$L2_{ICaCE}$	0.59 (.01)	0.18 (.01)	0.51 (.00)	0.31 (.00)	0.29 (.01)	0.68 (.06)	0.61 (.00)	0.31 (.01)
	COS_{ICaCE}	1.00 (.01)	0.78 (.02)	0.70 (.03)	0.71 (.03)	1.00 (.01)	1.12 (.02)	0.82 (.00)	0.80 (.00)
	$NormDiff_{ICaCE}$	0.52 (.00)	0.18 (.00)	0.51 (.00)	0.31 (.00)	0.29 (.01)	0.54 (.04)	0.60 (.00)	0.31 (.01)
GPT-2	$L2_{ICaCE}$	0.59 (.00)	0.19 (.01)	0.50 (.00)	0.31 (.00)	0.29 (.00)	0.39 (.01)	0.55 (.01)	0.45 (.01)
	COS_{ICaCE}	1.01 (.01)	0.69 (.01)	0.58 (.01)	0.61 (.01)	1.00 (.00)	1.02 (.00)	0.79 (.01)	1.00 (.00)
	$NormDiff_{ICaCE}$	0.51 (.01)	0.19 (.01)	0.50 (.00)	0.31 (.00)	0.29 (.00)	0.35 (.01)	0.53 (.01)	0.41 (.01)
LSTM	$L2_{ICaCE}$	0.58 (.01)	0.20 (.01)	0.51 (.00)	0.32 (.01)	0.31 (.00)	0.78 (.05)	0.28 (.00)	0.47 (.01)
	COS_{ICaCE}	1.00 (.01)	0.77 (.00)	0.70 (.01)	0.71 (.01)	1.01 (.01)	1.00 (.00)	1.00 (.00)	0.81 (.00)
	$NormDiff_{ICaCE}$	0.50 (.01)	0.20 (.01)	0.50 (.00)	0.32 (.01)	0.29 (.00)	0.64 (.04)	0.28 (.00)	0.46 (.01)

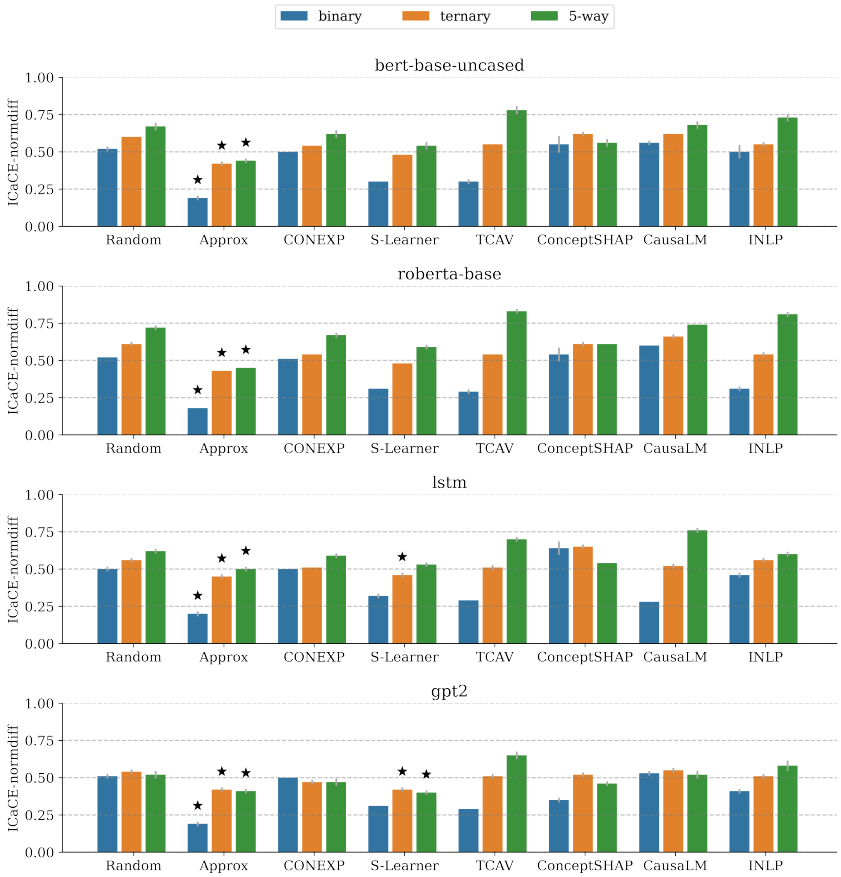


Figure 11: ICaCE-Error for all experiments using the *normdiff* distance metric. **Lower is better**. Results averaged over 5 distinct seeds. Error bars (in gray) display the standard deviation. Stars denote the best results for a given classification setting.

where ϕ_C^{CF} denotes the learned counterfactual representation that information about concept C is not present, $\mathcal{N}'$ is a classifier trained on this counterfactual representation, and $\mathcal{D}$ is a dataset.

However, for comparison on the CEBaB data, we require the estimation of individual causal concept effects (ICaCE). To allow a fair comparison, we swap the TReATE output estimator with TReITE (Equation 10). The only difference between these estimators is that in TReITE we remove the average across $\mathcal{D}$, and output the estimated effect of individual examples.

E.2 Implementation details

For all counterfactual models, we optimize using the Adam optimizer with `lr=2e-5`, `epochs=3`, `batch_size=48`, and the relative weight of the adversarial task, λ, is set to 0.1.

For both the factual models and fine-tuning phase, we optimize using the Adam optimizer with `lr=1e-3`, `epochs=50`, and `batch_size=256`. The differences in hyperparameter values is due to the different architectures we employ; for the counterfactual models we train the entire language model (ϕ), and for the factual models and the fine-tuning phase we freeze the embedding weights (ϕ) and train only the classification head ($\mathcal{N}$).

All CausaLM models were trained using 2 Nvidia GTX 1080 Ti 12GB GPUs.

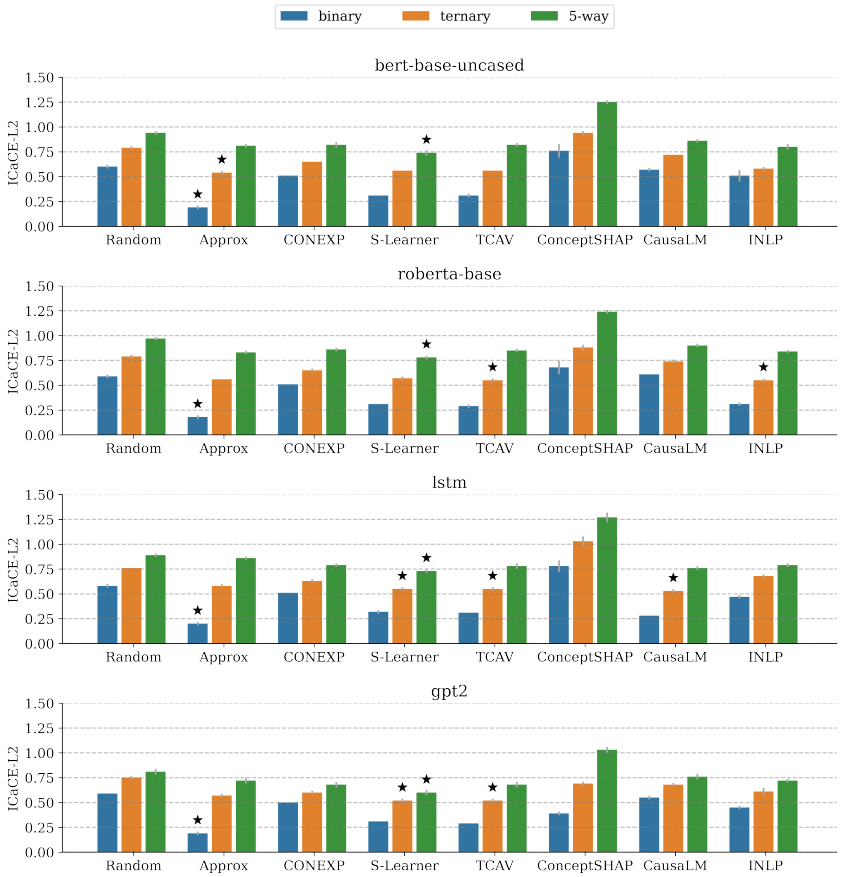


Figure 12: ICaCE-Error for all experiments using the *L2* distance metric. **Lower is better**. Results averaged over 5 distinct seeds. Error bars (in gray) display the standard deviation. Stars denote the best results for a given classification setting.

F INLP

F.1 Our adaptation

The INLP algorithm was originally designed to debias word embeddings by iteratively projecting them onto the null-space of some protected attribute (concept). However, INLP may serve as an estimation method similar to CausaLM, with the two following crucial differences. First, its lack of ability to control for potential confounders. Second, it operates on the representation rather than on the actual model weights. Since CausaLM and INLP share common characteristics, their output estimators are computed in the same way. See §E for extended details.

F.2 Implementation details

In order to guard for a "protected attribute" (concept), INLP determines whether this concept is present in an embedding or not by learning a linear separator in the embedding space. Following the practice suggested in the original paper, we choose our linear separator to be an SVM learned using SGD with $\alpha = 0.01$, $\varepsilon = 0.001$, and `max_iter=1000`. Logistic regression showed similar behavior. We project the representation to the null-space with respect to the concept 10 times. In fact, and similarly to the original paper, we converge to random accuracy of predicting the concept from the counterfactual representation after 4-5 iterations.

For all concepts, the classification head on top of the language model that trained to predict the overall sentiment labels trains for 5 epochs using the Adam optimizer with `lr=2e-5`.

G TCAV

G.1 Our adaptation

The Testing with Concept Activation Vectors (TCAV) explanation method was originally designed to count the percentage of test inputs from dataset $\mathcal{D}$ that are positively influenced by some high-level concept. It outputs a count over the number of examples that are change towards the direction of concept C, and computed as:

$$\text{TCAV}_{\mathcal{N}_\phi}(k, C; \mathcal{D}) = \frac{|\{x \in \mathcal{D} : \nabla \mathcal{N}_k(\phi(x)) \cdot v_C > 0\}|}{|\mathcal{D}|}, \tag{22}$$

where k is some class index and v_C is a linear direction in the activation space, given by the coefficients of a linear separator trained to distinguish between examples that include or exclude the concept C.

While TCAV's output is a count over examples, we use the raw sensitivity (directional derivative). This approach is supported by the authors of the original paper: "one could also use a different metric that considers the magnitude of the conceptual sensitivities" [26]. Also, since TCAV operates on the gradients of a model's logits but the ICaCEs are the difference of two probability vectors, we normalize its outputs by taking `Tanh`.

G.2 Implementation details

To learn the Concept Activation Vector (CAV, i.e., a linear direction in the activation space of ϕ), we train a linear separator to distinguish between examples that include the concept (labeled positive or negative) and examples that do not include it (labeled unknown). When learning CAVs, we drop all CEBaB train examples that are not labeled for aspect (concept) or do not have a majority with respect to the aspect.

Identically to the original paper, our CAV linear separator is an SVM learned using SGD with $\alpha = 0.01$, $\varepsilon = 0.001$ and `max_iter` $= 1000$.

H ConceptSHAP

H.1 Our adaptation

The original ConceptSHAP algorithm takes a complete set of concepts $C \in \{C_1, ..., C_m\}$ (such that its completeness score in Equation 25 is higher than some threshold) and outputs the relative contribution to the test accuracy of each C_i. It outputs an estimator given by the following formula

$$\text{Shapley}_{\{C_1,\ldots,C_m\}}(C) = \sum_{S \subseteq \{C_1,\ldots,C_m\} \setminus C} \frac{(m - |S| - 1)!\,|S|!}{m!} \left[\eta(S \cup \{C\}) - \eta(S)\right], \tag{23}$$

where η is a scoring function operating on sets of concepts that output accuracy ratios.

Similarly to the other methods, if η outputs accuracy ratios, then the output of ConceptSHAP is not a suitable estimator for ICaCE. Our straightforward adaptation for ConceptSHAP is to make η output class probabilities for classes instead of accuracy ratios.

Our adapted version outputs a vector for each $C \in \{C_1, \ldots, C_m\}$ and x according to the following equation:

$$\text{ConceptSHAP}_{\mathcal{N}_\phi}(C; x) = \sum_{S \subseteq \{C_1,\ldots,C_m\} \setminus C} \frac{(m - |S| - 1)!\,|S|!}{m!} \left[\eta(S \cup \{C\}) - \eta(S)\right], \tag{24}$$

where η is a function defined as $\eta_{\mathcal{N}_\phi}(S) = \sup_g \mathcal{N}\big(g\big(V_S\, \phi(x)\big)\big)$, and V_S is a matrix with the learned concept directions as its rows $V_S = \big(v_C^T\big)_{C \in S} \in \mathbb{R}^{|S| \times h}$.

Yeh et al. [57] calculate concept directions v_{C_j} automatically by learning a neural network classifier. To allow for a fair comparison between ConceptSHAP and the other evaluated methods, we use the

concept activation vectors $v_{C_1}, \ldots, v_{C_m}$ as the input concepts (similarly to those used in Kim et al. [26]).

In addition, in the original paper the authors learn the concepts v_C automatically, by using a carefully constructed loss function. To allow a fair comparison, we learn the concept vector by exploiting our labeled aspects (concepts), in a way similar to TCAV. See Section G.2 for more details.

H.2 Completeness Scores of Treatment Concepts

Given a feature representation ϕ and a classification head $\mathcal{N}$, the completeness score is defined by:

$$\text{completeness}_{\mathcal{N}_\phi}(S; D, Y) = \frac{\sup_g \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D},Y} \mathbb{1}\left[y = \arg\max_{y'} \mathcal{N}_{y'}\left(g\left(V_S\, \phi(x)\right)\right)\right] - a_r}{\frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D},Y} \mathbb{1}\left[y = \arg\max_{y'} \mathcal{N}_{y'}\left(\phi(x)\right)\right] - a_r}, \tag{25}$$

where a_r is is the accuracy of a classifier that outputs random predictions, $S \subseteq \{C_1, ..., C_m\}$ and V_S is a matrix with the learned concept directions as its rows $V_S = \left(v_C^T\right)_{C\in S} \in \mathbb{R}^{|S|\times h}$.

For all models, the completeness we get for the set of concepts $S = \{\text{ambiance}, \text{food}, \text{service}, \text{noise}\}$ is larger than 0.9.

H.3 Hyperparameters

The hyperparameters for CAV are identical to those of TCAV (Section G.2). To calculate η and the completeness score, we follow the original paper and set g to be a two-layer perceptron with 500 hidden units, learned using Adam optimizer for 50 epochs, employing `lr=1e-2` and `batch_size=128`.