Interpretable Neural Architectures for Attributing an Ad's Performance to its Writing Style

Reid Pryzant*
Stanford University
rpryzant@stanford.edu

Kazoo Sone
Google
sone@google.com

Sugato Basu
Google
sugato@google.com

Abstract

How much does "free shipping!" help an advertisement's ability to persuade? This paper presents two methods for *performance attribution*: finding the degree to which an outcome can be attributed to parts of a text while controlling for potential confounders[1]. Both algorithms are based on interpreting the behaviors and parameters of trained neural networks. One method uses a CNN to encode the text, an adversarial objective function to control for confounders, and projects its weights onto its activations to interpret the importance of each phrase towards each output class. The other method leverages residualization to control for confounds and performs interpretation by aggregating over learned word vectors. We demonstrate these algorithms' efficacy on 118,000 internet search advertisements and outcomes, finding language indicative of high and low click through rate (CTR) regardless of who the ad is by or what it is for. Our results suggest the proposed algorithms are high performance and data efficient, able to glean actionable insights from fewer than 10,000 data points. We find that quick, easy, and authoritative language is associated with success, while lackluster embellishment is related to failure. These findings agree with the advertising industry's emperical wisdom, automatically revealing insights which previously required manual A/B testing to discover.

1 Introduction

A text's style can affect our cognitive responses and attitudes, thereby influencing behavior (Spence, 1983; Van Laer et al., 2013). The predictive relationship between language and behavior has been well studied in applications of NLP to tasks like linking text to sales figures (Ho and Wu, 1999; Pryzant et al., 2017) and voter preference (Luntz, 2007; Ansolabehere and Iyengar, 1995).

In this paper, we are interested in interpreting rather than predicting the relationship between language and behavior. We focus on a specific instance: the relationship between the way a search advertisement is written and internet user behavior as measured by click through rate (CTR). In this study CTR is the ratio of clicks to impressions over a 90-day period, i.e. the probability of a click, given the person saw the ad. Our goal is to develop a method for *performance attribution* in textual advertisements: identifying lexical features (words, phrases, etc.) to which we can attribute the success (or failure) of a search ad, regardless of who created the advertisement or what it is selling.

Identifying linguistic features that are associated with various outcomes is a common activity among machine learning scientists and practitioners. Indeed, it is essential for developing transparent and interpretable machine learning NLP models (Yamamoto, 2012). However, the various forms of regression and association quantifiers like mutual information or log-odds ratio that are the de-facto standard for feature weighting and text attribution all have known drawbacks, largely related to problems of multicollinearity (Imai and Kim, 2016; Gelman and Loken, 2014; Wurm and Fisicaro, 2014; Estévez et al., 2009; Szumilas, 2010).

Furthermore, these prior methods of text attribution critically fail to disentangle the explanatory power of the text from that of confounding information which could also explain the outcome. For example, in movie reviews, the actors who star in a film are the most powerful predictors of box office success (Joshi et al., 2010). However, these are words that the film's marketers can't change.

*This work was conducted while the first author was doing internship at Google.

[1]Our code is available at github.com/rpryzant/deconfounded_lexicon_induction/tree/master/text-performance-attribution

Likewise, the name of a well-known brand in an ad for shoes might boost its effectiveness, but if we attribute the ad's success to the brand terms, we are actually crediting the power of the brand, not necessarily an actionable writing strategy (Ghose and Sundararajan, 2006).

There is an emerging line of work on text understanding for confound-controlled settings (Johansson et al., 2016; Egami et al., 2017; Pryzant et al., 2018; Li et al., 2018), but these methods are usually concerned with making causal inferences using text. They are limited to word-features and can only tell you whether a word is discriminative. Attribution involves the more fine-grained problem of identifying discriminative subsequences of the text *and* being able to explain which level of the outcome these subsequences support.

We present a pair of new algorithms for solving this problem. Based on the Adversarial and Residualizing models of (Pryzant et al., 2018), these algorithms first train a machine learning model and then analyze the trained parameters on strategically chosen inputs to infer the most important features for each output class. Our first algorithm encodes the text with a convolutional neural network (CNN) and proceeds to predict the outcome and adversarially predict the confounders. We select attributional n-grams by projecting back the weights of the output layer onto the encoder's convolutional feature maps. Our second algorithm uses a bag-of-words text representation and is trained to learn the part of the text's effect that the confounds cannot explain. We get n-grams from this method by tracing back the contribution of each feature towards each outcome class.

We demonstrate these algorithms' efficacy by conducting attribution studies on high- and low-performing search advertisements across three domains: real estate, job listings, and apparel. We find the proposed algorithms lend importance to words that are more predictive and less confound-related than a variety of strong baselines.

2 Text Attribution

We begin by proposing a methodological framework for text attribution and formalizing the activity into a concrete task.

We have access to a vocabulary $V = \{v_1, ..., v_m\}$, text $T = (w_1, ..., w_t)$ that is represented as a sequence of tokens, where each w is an element of V, outcome variable $Y \in \{1, ..., k\}$, and confounding variable(s) C. The data consists of (T^i, Y^i, C^i) triples, where the i^{th} data point includes a passage of text, an outcome, and confounding information that could also explain the outcome. Note that parts of T and C are related because language reflects circumstance (the text T is usually authored within a broader pragmatic context, for example the intent to promote a certain product at a certain price); T and Y are related because language influences behavior; C and Y are related because circumstance also influences behavior. We are interested in isolating the T-Y relationship and finding out which parts of the text act towards each possible outcome. We do so by choosing a lexicon $L_1, ..., L_k \subset V$ for each each outcome class Y_i such that the outcome x in observation (T^i, $Y^i = x$, C^i) can be credited to $T^i \cap L_x$, regardless of C. In other words, observing $Y^i = x$ can always be attributed to the tokens in L_x no matter the circumstances.

Saying that $Y^i = x$ can be attributed to L_x means (1) the words in L_x have a causal effect on Y and (2) that these words push Y towards class x, i.e., L_x is associated with class x. Based on the potential outcomes model of (Holland et al., 1985; Splawa-Neyman et al., 1990; Rubin, 1974; Pearl, 1999), Pryzant et al. (2018) developed a *causal informativeness coefficient* which measures the causal effects of a lexicon L on Y:

$$\mathcal{I}(L) = \mathbb{E}\left[\left(Y - \mathbb{E}\left[Y|C, T \cap L\right]\right)^2\right] - \mathbb{E}\left[\left(Y - \mathbb{E}\left[Y|C\right]\right)^2\right], \quad (1)$$

$\mathcal{I}(L)$ measures the ability of $T \cap L$ to explain Y's variability beyond the information already contained in the confounders. One computes $\mathcal{I}(L)$ by (1) regressing C on Y , (2) regressing $C + L \cap T$ on Y, and (3) measuring the difference in cross-entropy error between these models over a test set.

So $\mathcal{I}(L_x)$ measures the degree to which L_x influences Y, but it can't describe the degree to which L_x influences Y *towards* the specific outcome x. We propose circumventing this issue with a new *directed* informativeness coefficient $\mathcal{I}'(L, x) = \bar{lo}(L, x) \cdot \mathcal{I}(L)$, where $\bar{lo}$ is the average strength of association between the tokens in L_x and outcome x, as measured by log-odds:

$$\bar{lo}(L, x) = \frac{\sum_{v \in L} \log\ p_v^x - \log\left(1 - p_v^x\right)}{|L|} \quad (2)$$

$$p_v^x = \frac{\text{count}(Y = x \wedge v \in T)}{\text{count}(v \in T)} \quad (3)$$

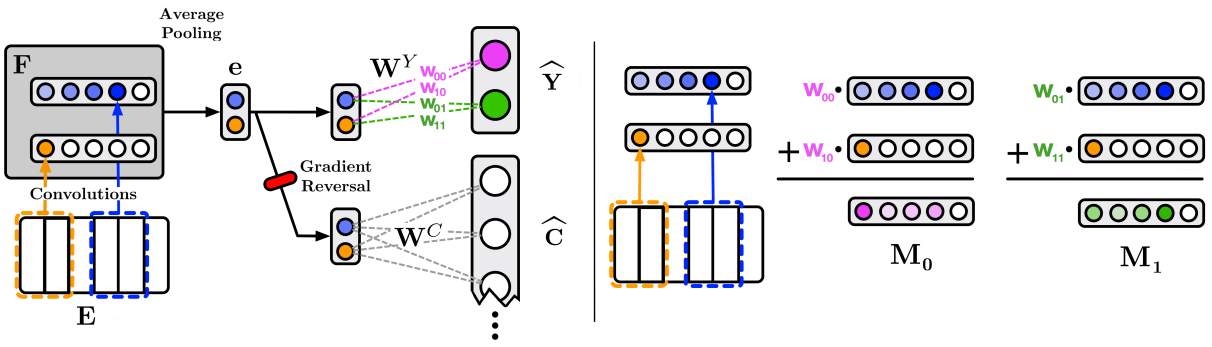

Figure 1: A Convolutional Adversarial Selector with $f = 2$ filters (both of size $n = 2$). Having filters of size 2 restricts this model to bigram attribution. Best viewed in color. **Left:** training phase. **Right:** interpretation phase.

Intuitively, if $\mathcal{I}'(L_x, x)$ is high, then L_x is both highly influential on Y and strongly associated with outcome x.

3 Proposed Algorithms

We continue by describing the pair of novel algorithms we are proposing to use for text attribution. Each algorithm consists of two phases: *training*, where we use T, Y, and C to train a machine learning model, and *interpretation*, where we analyze the learned parameters to identify attributional language.

3.1 Convolutional Adversarial Selector (CA)

Training. We begin by observing that the language we want to attribute should be able to explain the variation in Y and should also be decorrelated from the confounders C. This implies that the features we want to select should be predictive of Y, but not C (*e.g.* brand name). The Convolutional Adversarial Selector (CA) draws inspiration from this. It adversarially learns encodings of T which are useful for predicting Y but are *not* useful for predicting C. The model is depicted on the left-hand side of Figure 1.

First, we encode T into $\mathbf{e} \in \mathcal{R}^f$ with the following steps:

1. Embed the tokens of T with word vectors of dimension e. If the input text sequence has length t, the embedded input is a matrix $\mathbf{E} \in \mathcal{R}^{e \times t}$.

2. Slide convolutional filters of size $f \times n$ along the *time* axis of $\mathbf{E}$, where n are the n-gram size(s) we are interested in attributing during the interpretation stage. This process transforms text T into a set of n-gram features of various sizes, n. The input are now transformed into $\mathbf{F^n} \in \mathcal{R}^{f \times (t-n+1)}$, aka f one-dimensional feature maps of length $t-(n-1)$ for each n-gram size n.

3. Perform *global average pooling* (Lin et al., 2014) on $\mathbf{F^n}$. We now have our encoding $\mathbf{e^n} \in \mathcal{R}^f$, where each $e^n_j = \sum_i {F^n}_{j,i}$.

4. Concatenate all $\mathbf{e^n}$'s from every filter width n. This produces the final encoding, $\mathbf{e}$.

Armed with $\mathbf{e}$, we proceed to predict Y and C with a single linear transformation:

$$\hat{\mathbf{Y}} = \mathbf{e}\mathbf{W}^Y$$
$$\hat{\mathbf{C}} = \mathbf{e}\mathbf{W}^C$$

The model receives error signals from both of these "prediction heads" via a cross-entropy loss term:

$$\mathcal{L} = \sum_i -p_i \log \hat{p}_i \tag{4}$$

Where p_i and $\hat{p}_i$ correspond to the ground truth and predicted probabilities for class i, respectively.

Last, as gradients backpropagate from the C-prediction head to the encoder, we pass them through a *gradient reversal layer* in the style of (Ganin et al., 2016; Britz et al., 2017), which multiplies gradients by -1. If the loss of the Y-prediction head is $\mathcal{L}_Y$, and that of the confounders is $\mathcal{L}_C$, then the loss which is implicitly used to train the encoder is $L_e = L_Y - L_C$. This encourages the encoder to match $\mathbf{e}$'s distributions, regardless of C, thereby learning representations of the text which are invariant to the confounders (Xie et al., 2017).

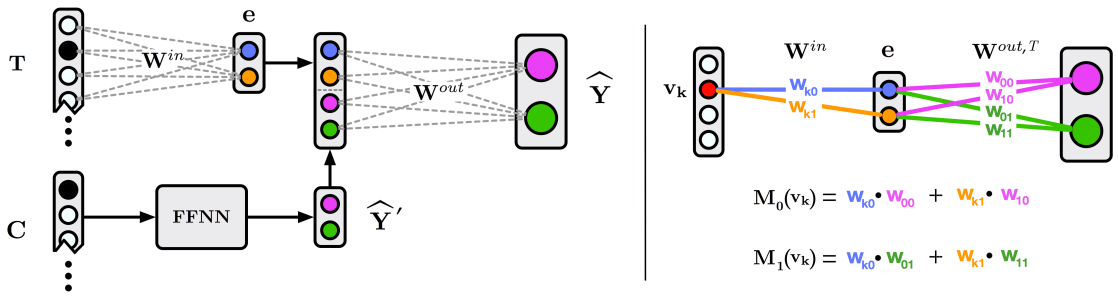

Figure 2: A Directed Residualization Selector with input embeddings of size $f = 2$. Best viewed in color. **Left:** training phase. **Right:** interpretation phase.

Interpretation. Once we've trained a CA model, we interpret its behavior in order to determine the most important n-grams for each level of the outcome. This stage is depicted in the right-hand side of Figure 1.

Inspired by the *class activation mapping* technique for computer vision (Zhou et al., 2016), we project the weights of $\mathbf{W}^Y$, the output layer, onto $\mathbf{F^n}$, the convolutional feature maps. Since $\hat{Y}_k = \sum_i e_i W^Y_{i,k}$, each $W^Y_{i,k}$ indicates the importance of e_i for class k. The elements of $\mathbf{e}$ are averages of each feature map, so $W^Y_{i,k}$ also indicates the importance of the i^{th} feature map for class k. Each feature map contains one activation per n-gram feature. This means we can quantify the importance of the j^{th} n-gram feature v^n_j towards each output class k by summing over all feature maps:

$$M_k(v^n_j) = \sum_i F^n_{i,j} \, W^Y_{i,k} \tag{5}$$

M_k is a mapping between input features and their importance towards class k.

In order to draw lexicons L_i from our vocabulary V, we perform interpretation over a dataset and map each (n-gram, outcome class) tuple to all of the importance values it was assigned. We then compute the average importance for each n-gram and select the top k for inclusion in the outgoing lexicon.

Note that this algorithm is only interpretable to the extent that there is a single linear combination relating $\mathbf{e}$ to $\hat{Y}$. With multiple layers at the "decision" stage of the network, the relationship between each dimension of $\mathbf{e}$ (and by extension, the rows of $\mathbf{F}$) and each output class becomes obfuscated.

3.2 Directed Residualization Selector (DR)

Training. Recall from Section 2 that $\mathcal{I}'(L, x)$ measures two quantities: (1) the amount by which L can further improve predictions of Y compared to the prediction only made from the confounders C, and (2) the strength of association between members of L and outcome class x. The Directed Residualization method is directly motivated by this setup. It first predicts Y directly from C as well as possible, and then seeks to fine-tune these predictions using T. This two-stage prediction process lets us control for the confounders C, because T is being used to predict the part of Y that the confounders can't explain. This model is depicted in the left-hand side of Figure 2.

First, the confounders C are converted into one-hot feature vectors that are passed through a feed-forward neural network (FFNN) to obtain a vector of preliminary predictions $\hat{\mathbf{Y}}'$. We then re-predict the outcome with the following steps:

$$\mathbf{e} = \mathbf{t}\ \mathbf{W}^{in} \tag{6}$$

$$\hat{\mathbf{Y}} = \left[\ \mathbf{e}\ \middle|\ \hat{\mathbf{Y}}'\ \right]\ \mathbf{W}^{out} \tag{7}$$

Where $\mathbf{t} = \{0,1\}^{|V|}$ is a bag-of-words representation of T, $\mathbf{W}^{in} \in \mathcal{R}^{|V| \times f}$, $\mathbf{e} \in \mathcal{R}^f$, $\mathbf{W}^{out} \in \mathcal{R}^{(f+k) \times k}$, and k is the number of classes in Y. The model receives supervision from both $\hat{\mathbf{Y}}'$ and $\hat{\mathbf{Y}}$. We use the same cross-entropy loss function as the Convolutional Adversarial Selector of Section 3.1.

Note the similarities between this approach and the popular residualizing regression (RR) attribution technique (Jaeger et al., 2009; Baayen et al., 2010, inter alia). Both use the text to improve an estimate generated from the confounds. RR treats this as two separate regression tasks (using C to predict Y, then T to predict the first model's

residuals). We introduce the capacity for nonlinear interactions by backpropagating between RR's steps.

Interpretation. This stage is depicted in the right-hand side of Figure 2. Once we've trained a DR model, we determine the importance of each feature v_j for each class Y_k by tracing all possible paths between v_j and Y_k, multiplying the weights along those paths, then summing across paths. The resulting importance value, $M_k(v_j)$, is how much Y_k's log-likelihood increases if v_j is added to a text according to the trained model (and thus irrespective of the confounders).

We can derive this procedure by considering the models' parameters. In equation 7, we produce log-likelihood estimates for $\mathbf{Y}$ by concatenating $\mathbf{e}$ and $\hat{\mathbf{Y}}'$ and multiplying the result with $\mathbf{W}^{out}$. This means the first $|\mathbf{e}| = f$ rows of $\mathbf{W}^{out}$ (written as $\mathbf{W}^{out,T}$) are an *output projection* transforming $\mathbf{e}$ into $\hat{\mathbf{Y}}^T$, the text's contribution towards $\hat{\mathbf{Y}}$. So $W^{out}_{i,k}$ indicates the importance of e_i for output class k. As per equation 6, $\mathbf{e}$ is the sum of all of the rows of $\mathbf{W}^{in}$ that correspond to features in the text. So we can decompose $\hat{\mathbf{Y}}^T$ into a sum of contributions from each text feature v_j:

$$\hat{\mathbf{Y}} = \left[\ \mathbf{e}\ \middle|\ \hat{\mathbf{Y}}'\ \right] \left[\frac{\mathbf{W}^{out,T}}{\mathbf{W}^{out,C}}\right]$$

$$\hat{\mathbf{Y}}^T = \mathbf{t}\,\mathbf{W}^{in} \cdot \mathbf{W}^{out,T}$$

$$\hat{Y}^T_k = \sum_j^{|V|} \sum_i^f \mathbb{1}_T(v_j) W^{in}_{j,i}\, W^{out,T}_{i,k}$$

And the estimated log-likelihood contribution of of any v_j towards class k is

$$M_k(v_j) = \sum_i^f W^{in}_{j,i}\, W^{out,T}_{i,k} \tag{8}$$

For this algorithm, there is no need to run the model over any data in order to retrieve importance values – we can directly obtain these values from the trained parameters. This procedure is depicted in the right-hand side of Figure 2.

Last, like the CA algorithm, DR is only interpretable to the extent that there is a single linear combination between $\mathbf{e}$ and $\hat{Y}$.

4 Experiments

We demonstrate the efficacy of the proposed algorithms on a dataset of internet advertisements.

4.1 Experimental Set-Up

Data. In this setting our (T, Y, C) data triples consist of

- T: the header text of sponsored search results in an internet search engine.
- Y: a binary categorical variable which indicates whether the corresponding advertisement was high-performing or low-performing.
- C: a categorical variable which indicates the *brand* of the ad. We use customer id and the hostname of the landing page the ad points to as a proxy for this.

We collect advertisements across three domains: apparel (16,000 advertisements), job listings (70,000), and real estate (32,000). See section A for more details on these data. We selected pairs of ads where both had the same landing page and targeting, but where one ad was in the 97.5^{th} CTR percentile (high-performing) and its counterpart was in the 2.5^{th} percentile (low-performing). This implies that any performance differences may be attributed to differences in their text.

We tokenized these data with Moses (Koehn et al., 2007) and joined word-tokens into n-grams of size 1, 2, 3, and 4 for the n-gram portion of the study.

Implementation. We implemented nonlinear models with the Tensorflow framework (Abadi et al., 2016) and optimized using Adam (Kingma and Ba, 2014) with a learning rate of 0.001. We implemented linear models with the scikit learn package (Pedregosa et al., 2011). We evaluate each algorithm by selecting lexicons of size $|L_i| = 50$. We optimized the hyperparameters of all algorithms for each dataset. Complete hyperparameter specifications are provided in the online supplementary materials; for the proposed **DR** and **CA** algorithms we set $|\mathbf{e}|$ to 8, 32, and 32 for the apparel, job listing, and real estate data, respectively.

Baselines. Along with the **C**onvolutional **A**dversarial Selector (**CA**) and **D**irected **R**esidualization Selector (**DR**) of Section 3, we compare the following methods: **R**egression (**R**), **R**esidualized **R**egressions (**RR**), **R**egression with **C**onfound features (**RC**), and the **A**dversarial **S**election (**AS**) algorithm of (Pryzant et al., 2018), which selects words that are important for a confound-controlled prediction task

by considering the attentional scores of an adversarially-trained RNN.

4.2 Experimental results

We begin by investigating whether the proposed methods successfully discovered features that are simultaneously indicative of each CTR status and untangled from the confounding effects of brand (Tables 1, 2, 3).

	High CTR					
	Unigrams			N$-grams$		
	$\bar{lo}$	$\mathcal{I}$	$\mathcal{I}'$	$\bar{lo}$	$\mathcal{I}$	$\mathcal{I}'$
DR	0.84	1.19	1.01	2.09	0.81	**1.68**
CA	1.28	1.19	**1.53**	1.99	0.78	1.55
AS	0.59	0.35	0.21	0.58	0.61	0.36
R	0.91	0.83	0.76	0.68	0.63	0.43
RC	0.92	0.99	0.90	0.55	0.78	0.43
RR	0.23	0.36	0.08	0.01	0.21	0.00
	Low CTR					
	Unigrams			N$-grams$		
	$\bar{lo}$	$\mathcal{I}$	$\mathcal{I}'$	$\bar{lo}$	$\mathcal{I}$	$\mathcal{I}'$
DR	0.73	0.78	0.58	1.12	0.88	0.99
CA	1.17	0.81	**0.96**	1.42	0.88	**1.26**
AS	0.58	0.20	0.11	0.56	0.42	0.24
R	0.79	0.46	0.37	0.83	0.52	0.43
RC	1.05	0.29	0.31	1.42	0.49	0.70
RR	0.24	0.34	0.08	0.20	0.14	0.03

Table 1: Comparative performance over apparel advertisements. $\mathcal{I}$ and $\mathcal{I}'$ are inflated by an order of magnitude for readability.

On the apparel data (Table 1), we find that the proposed algorithms select words that are often both the most influential on CTR (highest $\mathcal{I}$) and are also the most strongly associated with their target outcome classes (highest $\bar{lo}$). It is not surprising that the Adversarial Selector of (Pryzant et al., 2018) (**AS**) had low $\bar{lo}$ because the method is only capable of identifying discriminative features while controlling for confounds. **AS** was also inconsistent in its ability to select words that are predictive of CTR while being unrelated to brand. This may be due to the instability of adversarial learning (Shrivastava et al., 2017) or the complex nonlinear relationship between the model's attention scores and final predictions.

On the job advertisements (Table 2), the proposed **DR** algorithm performed the best, selecting words that were both more influential on CTR and more strongly associated with its target than

	High CTR					
	Unigrams			N$-grams$		
	$\bar{lo}$	$\mathcal{I}$	$\mathcal{I}'$	$\bar{lo}$	$\mathcal{I}$	$\mathcal{I}'$
DR	0.67	0.61	**0.41**	3.63	0.25	**0.91**
CA	1.33	0.17	0.22	3.35	0.17	0.57
AS	0.43	0.33	0.14	2.42	0.25	0.60
R	0.65	0.13	0.08	2.98	0.17	0.51
RC	0.35	0.71	0.24	3.04	0.16	0.51
RR	0.26	0.40	0.10	1.81	0.18	0.33
	Low CTR					
	Unigrams			N$-grams$		
	lo	$\mathcal{I}$	$\mathcal{I}'$	lo	$\mathcal{I}$	$\mathcal{I}'$
DR	0.89	1.04	0.93	3.43	0.20	**0.69**
CA	1.20	0.86	**1.02**	4.62	0.13	0.62
AS	0.12	0.54	0.07	3.12	0.18	0.56
R	0.76	0.85	0.65	1.95	0.13	0.26
RC	0.48	0.97	0.47	1.90	0.13	0.24
RR	0.36	0.82	0.03	0.90	0.12	0.11

Table 2: Comparative performance over job postings. $\mathcal{I}$ and $\mathcal{I}'$ are inflated by an order of magnitude for the *unigram results only.*

any other algorithm. In general, $\mathcal{I}$ values were an order of magnitude larger for n-grams than unigrams, indicating that for job postings on the internet, *phrases* are more important than the individual words they are composed of. This suggests job seekers may read advertisements more closely than internet shoppers, who are known to "skim" content and are thus more attuned to individual keywords (Campbell and Maglio, 2013; Seda, 2004).

For real estate, Table 3 indicates that except for the case of weak unigrams, the proposed **DR** and **CA** algorithms can perform best. In many cases, the regression-based approaches successfully selected words that are strongly related to each target outcome class ($\bar{lo}$ was relatively high), but failed to choose words whose explanatory power exceeds that of the confounds ($\mathcal{I}$ was relatively low). For a plain regression (**R**) this makes sense; there is no mechanism to control for confounders. For the other regression-based approaches (**RC** & **RR**), this may be due to the multicolinearity of confounders and text which is described in (Gelman and Loken, 2014; Wurm and Fisicaro, 2014) as a fundamental weakness of these attribution algorithms. Again, n-grams performed drastically better than unigrams, implying that phraseology may matter more than vocabulary to prospective home-

	High CTR					
	Unigrams			N−*grams*		
	lo	$\mathcal{I}$	$\mathcal{I}'$	lo	$\mathcal{I}$	$\mathcal{I}'$
DR	0.75	0.32	**0.25**	2.16	0.05	**0.12**
CA	1.00	0.24	0.24	2.63	0.04	0.11
AS	0.33	0.13	0.04	1.20	0.03	0.03
R	0.56	0.06	0.03	2.32	0.05	0.11
RC	0.68	0.05	0.03	1.76	0.04	0.08
RR	0.21	0.20	0.04	0.74	0.03	0.02
	Low CTR					
	Unigrams			N−*grams*		
	lo	$\mathcal{I}$	$\mathcal{I}'$	lo	$\mathcal{I}$	$\mathcal{I}'$
DR	0.60	0.12	0.07	1.80	0.18	0.32
CA	0.80	0.09	0.08	2.05	0.16	**0.33**
AS	0.12	0.14	0.01	0.18	0.25	0.04
R	0.63	0.07	0.05	0.49	0.33	0.16
RC	1.39	0.07	**0.10**	0.57	0.17	0.10
RR	0.22	0.05	0.01	0.14	0.08	0.01

Table 3: Comparative performance over real estate advertisements. $\mathcal{I}$ and $\mathcal{I}'$ are inflated by an order of magnitude for the *unigram results only.*

owners.

4.3 Algorithmic Analysis

Ablation Study. We proceed to ablate the mechanism by which each proposed algorithm controls for the confounds. First we toggled the gradient reversal layer of the Convolutional Adversarial Selector (**CA**). Doing so reduced the algorithm's performance by an average of 0.03 $\bar{lo}$ and 0.24 $\mathcal{I}$. For the Directed Residualization Selector (**DR**), we removed the part of the model that uses the confounds to generate preliminary predictions. Doing so resulted in an average increase of 0.02 $\bar{lo}$ and a decrease of 0.21 $\mathcal{I}$. For both algorithms, only the average difference in $\mathcal{I}$ was significant ($p < 0.05$). From these results, we conclude that these confound-controlling mechanisms bear little impact on the degree to which the selected words are associated with their corresponding outcome classes. However, the mechanisms are important for getting the models to avoid confound-related features.

Visualization. We visualize $M_{\text{high}-\text{CTR}}$ and $M_{\text{low}-\text{CTR}}$ as computed by a proposed and baseline method (Figure 3). We see that the regression lends high-CTR importance to the name of a popular real estate company, and low-CTR importance to an unpopular location (which that company happens to specialize in). The Adversarial Selector gives confound-related features less importance. By disabling the reversal layer, we recover some of the regression's confound-relatedness.

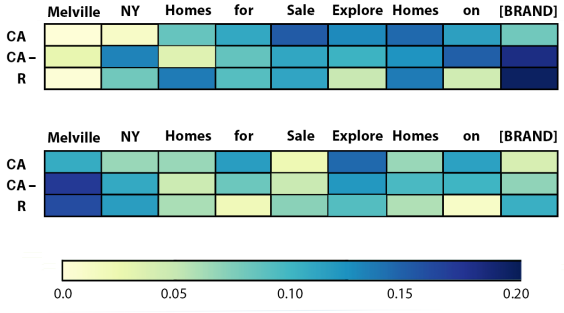

Figure 3: Feature importance maps for a real estate ad. high-CTR (top) and low-CTR (bottom) are the outcome classes. These maps are computed by the Convolutional Adversarial Selector with and without gradient flipping (CA, CA-) and a regression (R). Note that the Convolutional Adversarial Selector without gradient flipping (CA-) has similar weights to a regression model (R) while CA moves weight away from the brand-related words.

4.4 Language Analysis

We continue by studying high-scoring words and phrases from the models we experimented with in order to glean useful insights about internet advertising. Please note that this is an illustration of the present algorithm and this study is limited in scope. These are experimental results, not suggestions for real online advertising campaigns.

When comparing the words selected by the proposed and baseline methods, we observe that many of the regression-based methods selected brand names or words that are closely associated with brands, like locations (areas where real estate and staffing agencies specialize) or proper nouns (fashion designers, real estate agents, and so on). Indeed, for apparel, the percent of selected words and phrases which contained the name of a fashion retailer was less for DR and CA (6.5% and 8.5%) than AS (9%), R (23%) RC (19%) and RR (13%).

After clustering words and phrases based on the cosine similarity of their GloVe embeddings (Pennington et al., 2014), the authors found semantic classes that include industry best practices (e.g., Schwab, 2013). For example:

- **Involvement.** This includes language which

creates a dialogue with the reader ("your", "you", "we") and portrays a personal experience ("personalized") at the reader's discretion ("compare", "view"). This aligns with growing demand for personalized internet services (Meeker, 2018).

- **Authority.** This includes appeals to the rhetorical device of ethos, in the form of authoritative framing, such as "official site" and "®".

- **Logos.** These expressions appeal to the sensibilities of the reader, framing the product as easy ("simple", "any budget"), cheap ("outlet", "xx% off", "plus free shipping"), or available ("available", "shop them at").

We also find some semantic classes among weakly performing words and phrases. One notable class includes "filler words" consisting of lackluster embellishment. This aligns with prior psychological research suggesting that words that don't contribute to a topic can have a slightly negative effect on attitude (Fazio et al., 1986; Grush, 1976).

Finally, we note that popular items or categories of items were frequently high-scoring. This comes as no surprise and reflects an important aspect of the proposed methodology: it only controls for the confounders it is given, and we controlled for the *brand* of an ad, not its *content*. There are innumerable factors which influence clicking behavior (position, demographics, etc.) that we did not model explicitly in this study; we leave this to future work.

5 Related Work

Neural Network Interpretability. A variety of work has been done on understanding the relationship between input features and the network's behavior. Attention mechanisms (Bahdanau et al., 2015; Luong et al., 2015) are a popular method for highlighting parts of the input, but the nonlinear relationship between attention scores and ouputs makes it a poor tool for attribution on a per-class basis (as our Adversarial Selector (AS) baseline demonstrates). Dosovitskiy and Brox (2015) and Mahendran and Vedaldi (2015) invert the layers of a neural network to show which input features are being used. Zhou et al. (2016) extends this work to show exactly which parts of the input are being used. Parts of our Convolutional Adversarial Selector draw on this, and as far as these authors know, we are the first to adapt class activation maps to language data. Sundararajan et al. (2017) also highlight important parts of the input with a method that is similar to our Directed Residualization Selector. Their method uses gradients to trace influence. Because our models' gradients are a composite of signals, only some of which we want to consider while attributing, the method can't be applied directly to our setting. Ribeiro et al. (2016), Biran and McKeown (2017), and Lei et al. (2016) also use "importance scores" to explain the predictions of neural network-based classifiers.

Causal Inference. Our methods have connections to recent advances in the causal inference literature. Johansson et al. (2016) and Shalit et al. (2016) propose an algorithm for causal inference which bears similarity to our Convolutional Adversarial Selector (CA). Imai et al. (2013) advocate a lasso-based method similar to our Directed Residualization (DR), and Egami et al. (2018) explore how to make causal inferences from texts through careful data splitting. Unlike the present study, these papaers are largely unconcerned with the underlying interpretability. Pryzant et al. (2018) makes a foray into causal interpretability, developing the *informativeness coefficient* metric we use in our evaluations. This work also proposed two algorithms for deconfounded lexicon induction which inspired our proposed CA and DR algorithms.

6 Conclusion

In this paper, we presented two new algorithms for the analysis of persuasive text. These algorithms are based on interpreting the behaviors and parameters of trained machine learning models. They perform *performance attribution*, the practice of finding words that are indicative of particular outcomes and are unrelated to confounding information. We used these algorithms to conduct the first public investigation into successful writing styles for internet search advertisements. We find that the proposed method can automatically identify successful (and unsuccessful) writing styles of advertising. These findings are inline with industry practices built on manual A/B testing and also previous psychological studies. This is an exciting new direction for NLP research. There are many directions for future work, including core algorith-

mic innovation and applying the proposed algorithms to new and rich social questions.

7 Acknowledgments

We are grateful to Emanuel Schorsch, Kristen LeFevre and Jason Baldridge for their helpful comments and suggestions.

A Corpus Statistics

Table 4 shows general statistics of the corpus used in the present study.

Category	N	$\|V\|$	$\bar{t}$	$\bar{l}$
Apparel	16,242	4,635	9.3	53.9
Job Postings	70,016	7,312	10.1	54.4
Real Estate	32,398	6,952	9.1	54.2

Table 4: Corpus statistics of advertising text used in this study. N is the number of documents (advertising headlines) used in the study. $|V|$ is the vocabulary size (number of unique tokens in the category corpus). $\bar{t}$ and $\bar{l}$ are average number of tokens and average length per ad respectively.

References

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

Stephen Ansolabehere and Shanto Iyengar. 1995. *Going Negative: How Attack Ads Shrinks and Polarize the Electorate*. New York: Free Press.

R. Harald Baayen, Victor Kuperman, and Raymond Bertram. 2010. Frequency effects in compound processing. In *Compounding*, pages 257–270. Benjamins.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations (ICLR)*.

Or Biran and Kathleen R McKeown. 2017. Human-centric justification of machine learning predictions. In *IJCAI*, pages 1461–1467.

Denny Britz, Reid Pryzant, and Quoc V. Le. 2017. Effective domain mixing for neural machine translation. In *Second Conference on Machine Translation (WMT)*.

Christopher S Campbell and Paul Philip Maglio. 2013. Method of rewarding the viewing of advertisements based on eye-gaze patterns. US Patent 8,538,816.

Alexey Dosovitskiy and Thomas Brox. 2015. Inverting convolutional networks with convolutional networks. *CoRR abs/1506.02753*.

Naoki Egami, Christian J. Fong, Justin Grimmer, Margaret E. Roberts, and Brandon M. Stewart. 2017. How to make causal inferences using texts.

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

Pablo A. Estévez, Michel Tesmer, Claudio A. Perez, and Jacek M. Zurada. 2009. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201.

Russell H Fazio, David M Sanbonmatsu, Martha C Powell, and Frank R Kardes. 1986. On the automatic activation of attitudes. *Journal of personality and social psychology*, 50(2):229.

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

Andrew Gelman and Eric Loken. 2014. The statistical crisis in science data-dependent analysisa garden of forking pathsexplains why many statistically significant comparisons don't hold up. *American Scientist*, 102(6):460.

Anindya Ghose and Arun Sundararajan. 2006. Evaluating pricing strategy using e-commerce data: Evidence and estimation challenges. *Statistical Science*, pages 131–142.

Joseph E Grush. 1976. Attitude formation and mere exposure phenomena: A nonartifactual explanation of empirical findings. *Journal of Personality and Social Psychology*, 33(3):281.

Chin-Fu Ho and Wen-Hsiung Wu. 1999. Antecedents of customer satisfaction on the internet: An empirical study of online shopping. In *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, pages 9–pp. IEEE.

Paul W Holland, Clark Glymour, and Clive Granger. 1985. Statistics and causal inference. *ETS Research Report Series*, 1985(2).

Kosuke Imai and In Song Kim. 2016. *When Should We Use Linear Fixed Effects Regression Models for Causal Inference with Longitudinal Data?* Ph.D. thesis, Working paper, Princeton University, Princeton, NJ.

Kosuke Imai, Marc Ratkovic, et al. 2013. Estimating treatment effect heterogeneity in randomized program evaluation. *The Annals of Applied Statistics*, 7(1):443–470.

T. Florian Jaeger, Victor Kuperman, and Austin Frank. 2009. Issues and solutions in fitting, evaluating, and interpreting regression models. In *Talk given at WOMM presession to the 22nd CUNY Conference on Sentence Processing*.

Fredrik Johansson, Uri Shalit, and David Sontag. 2016. Learning representations for counterfactual inference. In *International Conference on Machine Learning*, pages 3020–3029.

Mahesh Joshi, Dipanjan Das, Kevin Gimpel, and Noah A. Smith. 2010. Movie reviews and revenues: An experiment in text regression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 293–296, Stroudsburg, PA, USA. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations (ICLR)*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *EMNLP*.

Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. Towards robust and privacy-preserving text representations. *arXiv preprint arXiv:1805.06093*.

Min Lin, Qiang Chen, and Shuicheng Yan. 2014. Network in network. *International Conference on Learning Representations*.

Frank Luntz. 2007. *Words that work: It's not what you say, it's what people hear*. Hachette Books.

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

Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. *Computer Vision and Pattern Recognition (CVPR)*.

Mary Meeker. 2018. Internet trends 2018. page 192.

Judea Pearl. 1999. Probabilities of causation: three counterfactual interpretations and their identification. *Synthese*, 121(1-2):93–149.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Reid Pryzant, Young-joo Chung, and Dan Jurafsky. 2017. Predicting sales from the language of product descriptions. In *Special Interest Group on Information Retrieval (SIGIR) eCommerce Workshop*.

Reid Pryzant, Kelly Shen, Dan Jurafsky, and Stefan Wagner. 2018. Deconfounded lexicon induction for interpretable social science. In *16th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM.

Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688.

Victor O. Schwab. 2013. *How to Write a Good Advertisement: A Short Course in Copywriting*. Echo Point Books Media.

Catherine Seda. 2004. *Search Engine Advertising: Buying your way to the top to increase sales*. New Riders.

Uri Shalit, Fredrik Johansson, and David Sontag. 2016. Estimating individual treatment effect: generalization bounds and algorithms. *arXiv preprint arXiv:1606.03976*.

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. 2017. Learning from simulated and unsupervised images through adversarial training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 6.

Donald P. Spence. 1983. Narrative persuasion. *Psychoanalysis & Contemporary Thought*.

Jerzy Splawa-Neyman, Dorota M Dabrowska, and TP Speed. 1990. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, pages 465–472.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*.

Magdalena Szumilas. 2010. Explaining odds ratios. *Journal of the Canadian Academy of Child and Adolescent Psychiatry*, 19(3):227.

Tom Van Laer, Ko De Ruyter, Luca M. Visconti, and Martin Wetzels. 2013. The extended transportation-imagery model: A meta-analysis of the antecedents and consequences of consumers' narrative transportation. *Journal of Consumer research*, 40(5):797–817.

Lee H. Wurm and Sebastiano A. Fisicaro. 2014. What residualizing predictors in regression analyses does (and what it does not do). *Journal of Memory and Language*, 72:37–48.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. 2017. Controllable invariance through adversarial feature learning. In *Conference on Neural Information Processing Systems (NIPS)*, Long Beach, California, USA.

Teppei Yamamoto. 2012. Understanding the past: Statistical analysis of causal attribution. *American Journal of Political Science*, 56(1):237–256.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2921–2929. IEEE.