Literature Review Milestone

CausalSent: Improving Text Understanding via Causal and Intervention-based Regularizations

Daniel Frees [*] [1] [2] **Martin Pollack** [*] [1]

1. Topic Overview

State-of-the-art natural language processing (NLP) and large language models (LLMs) have been shown to perform exceptionally well in tasks such as sentiment analysis, classification, and recommendation. However, these tend to be evaluated on test datasets with similar distributions of factors compared to the training sets. Performance can drop significantly for ”out-of-distribution” test datasets when there are systematic differences or temporal shifts. This may be due to the fact that language models are trained on massive amounts of data, enabling correspondingly significant overfitting and the learning of spurious correlations that do not generalize well. Towards better understanding the *causal* significance that words and phrases have on the meaning of a text, recent efforts have begun to focus on causal machine learning for NLP. Common techniques include employing regularization during baseline model fine-tuning to decouple spurious links (Bansal & Sharma, 2023), (Bansal et al., 2023) as well as creating and fine-tuning on synthetic counterfactual data (Zhang et al., 2021). Here we extend the analysis of (Bansal et al., 2023), which employs both techniques.

We further extend the work by asking the question: are causally robust effects necessary for downstream model performance? Other works such as (Bansal et al., 2023), (Pang et al., 2021), and (Zhang et al., 2021) all investigate heuristic approaches to estimate causal effects, each achieving moderate to significant performance gains for short or long text-matching and sequential user interactions prediction, respectively. Towards elucidating whether robust causal techniques or heuristics are more effective for model performance, we compare both causal and intervention-based regularization architectures.

The aforementioned regularization techniques require a clearly defined model output and task that needs to be accomplished. As such, we focus on two common NLP use cases: sentiment analysis and text matching.

Sentiment analysis is the task of mining emotions, attitudes, and opinions from text. In early development, researchers used lexicon-based scoring approaches to aggregate word sentiment scores from pre-defined static dictionaries. These were not particularly successful, and recently machine learning and deep learning approaches have completely dominated early methods. In particular, transformer-based models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and DistilBERT (Sanh et al., 2019) excel at sentiment analysis, as do few-shot trained and fine-tuned LLMs. While LLMs perform well on simple sentiment scoring tasks, they struggle on more complex tasks (Zhang et al., 2023) and failure points are difficult to identify given their complex black-box nature. This begs the question: do LLMs actual understand human perceptions of texts, or have they simply seen a huge number of correlations? Thus, sentiment analysis lends itself as a natural task for evaluating causal regularization methods. In particular, causal methods might be promising for increasing performance on texts where common correlations no longer hold true (e.g. irony), as well as increasing overall model interpretability.

Recommendation systems are critical to web browsing and information retrieval, powering systems ranging from Netflix TV show recommendations to modern LLM-based retrieval augmented generation (RAG). Much of the data enabling these systems takes the form of unstructured text, thus motivating the problem of *text-based matching*. Text-based matching is fundamentally a challenge in determining matching signal between sparsely related queries and items. This sparsity, combined with the fact that spurious correlations due to noise are context-dependent, makes text matching very difficult (e.g. a brand such as ‘Google‘ might be semantically relevant for matching in an item-recommendation scenario where we want to pair ‘Google Pixel 7 Pro‘ to ‘Google Phone Case‘ but spurious in the case of ‘Google Search’ and ’Google Pixel 7 Pro‘). Recent research has begun to investigate whether causal methodolo-

[*]Equal contribution [1]Department of Statistics, Stanford University [2]Google, San Francisco, CA. Correspondence to: Daniel Frees <dfrees@stanford.edu>, Martin Pollack <pollackm@stanford.edu>.

gies such as RieszNet-based counterfactual data augmentation or intervention-based regularization might be employed to better separate semantically relevant text from spurious correlations, thereby improving the results of text-based recommendation systems as data distributions change. This increased robustness is especially necessary for systems such as web searches where both query (search) and item (webpage) distributions are frequently changing.

2. Methods in the Literature

Numerous recent papers have sought to improve language models and recommendation system performance by reducing the effects of spurious correlations, often through a causal (or heuristically causal) framework.

2.1. Primary Approach

The primary work we seek to replicate and extend is (Bansal & Sharma, 2023). In this work, Bansal and Sharma investigate a novel methodology for regularizing spurious correlations using estimated text feature causal effects. Spurious features often arise because of helpful correlations in the majority group (samples where the correlations hold, constituting a majority of samples), but can confound predictions in the minority group (samples where the correlations break down, constituting a minority of samples). Most prior methods for reducing spurious correlations aimed to completely remove the spurious features; however (Bansal & Sharma, 2023) argue that this can detriment performance for certain samples.

Instead, the authors suggest their primary method *Feature Effect Augmentation (FEAG)*, which is comprised of three main steps. First, causal effects (”feature effects”) of text features are estimated using a doubly-robust ML estimator, RieszNet as suggested in (Chernozhukov et al., 2021). These feature effects can be used to generate synthetic counterfactual observations in which we flip the binary treatment variable and re-calculate the accompanying sentiment according to the estimated feature effect. These counterfactuals are appended into the training dataset to ground/regularize the model towards producing ’true’ treatment effects. Finally, we fine-tune a base sentiment model, such as BERT, by training against a weighted sample of the original and synthetic data. By following this training regime, the goal is to improve minority group performance without significantly reducing majority group performance.

2.1.1. Measuring Text Feature Effects

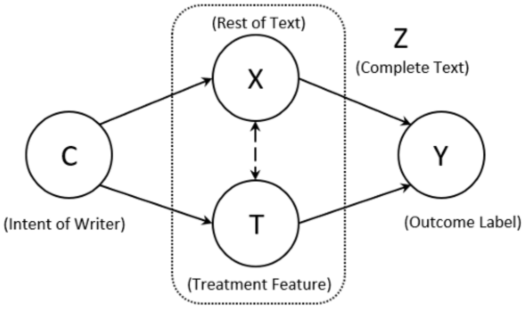

Figure 1. Caption

To describe the causal process generating a text with a particular sentiment/meaning, we propose the same causal graph as (Bansal & Sharma, 2023), pictured in Figure 1. A writer of a text has some intent, C, which through the words they write, Z, produces an outcome Y, such as the sentiment of the text (or meaning of the text). In the hopes of regularizing against the ’true’ feature effects, we must carefully consider how we compute feature effects. In the context of sentiment analysis, we consider feature effects to be the average treatment effect (ATE) of a text feature on sentiment. For example, if the phrase ”very fun” on average improves sentiment by $+0.26$, agnostic of other text features, then this would be the average treatment effect of ”very fun”. A naive approach to learning this treatment effect would be to train a model, g, to approximate the true/oracle data distributing function and learn the average treatment effect:

$$\widehat{\text{ATE}}_{\text{Direct}} = \frac{1}{n}\sum_i \left(g(X_i, 1) - g(X_i, 0)\right) \tag{1}$$

However, treatment (text containing the phrase ”very fun”) and control (text not containing the phrase ”very fun”) groups are not balanced in terms of the other text features or covariates, creating bias in the direct model ATE. It is more likely that the phrase ”very fun” would co-occur with a description of paintball than a description of getting knee surgery. As such, we must control for covariates, X, in attempting to estimate each text feature T’s treatment effect. It is reasonable to assume that other co-occuring words also impact the sentiment of the text, and that the treatment words and co-occuring words also affect each other, but only stemming form the intent of the writer (an unobserved latent variable C), as visualized in the graph. Given that T and Y are therefore d-separated by X, we can condition on X to estimate the causal effect of T on Y. It also suffices to condition on the propensity score $e(x) = Pr[T|X]$. This motivates a different, now unbiased ATE where we weight each outcome by the propensity of treatment (such that treated outcomes are upweighted when propensity was low, and vice versa).

$$\widehat{\text{ATE}}_{\text{propensity}} = \frac{1}{n}\sum_i \alpha_e(Z_i)Y_i. \tag{2}$$

where

$$\alpha_e(Z_i) = \left(\frac{T_i}{e(X_i)} - \frac{1 - T_i}{1 - e(X_i)} \right). \tag{3}$$

defining propensity as

$$e(X_i) = Pr[T|X_i] \tag{4}$$

An impactful earlier work on learning causal effects using neural networks, DragonNet (Shi et al., 2019) learns a propensity-based estimator for the average treatment effect (ATE) of each text feature, similar to the above method. However, propensity-based estimators suffer from instability when we have low overlap, meaning that $e(X_i)$ is approaching 1 or 0. In text data, it is highly likely for this to be the case. If we choose our treatment word to be 'murder', for example, it is very unlikely for that word to coexist with a set of words $\{$kitten, happy, playing$\}$. As a result, the propensity weight α_e becomes highly unstable. A newer method, RieszNet, is a much better option as it learns analogous weights α_{RR} directly rather than by estimating propensities as an intermediate step and plugging into a potentially unstable weight equation (Chernozhukov et al., 2021).

2.1.2. Regularizing Model Learning via Synthetic Counterfactuals

By automatically estimating causal effects via RieszNet, the approach of (Bansal & Sharma, 2023) enables not only counterfactual generation but also label estimation for the new counterfactuals. Other previous methods such as POLYUICE (Wu et al., 2021) introduce robust frameworks for producing a variety of counterfactuals, but ultimately still require manual crowd-sourced labeling efforts to produce training data.

These counterfactuals then can be used to regularize the downstream model to produce outputs closer to the 'true' feature effects. One way to achieve this is direct regularization: we add a term to our loss function controlling how far away the model's learned feature effect is from the estimated average treatment we found using RieszNet. Consider, for example, the sentiment analysis training. If we let $\mathcal{L}$ be the loss on the sentiment task (usually cross-entropy), $f(X, T)$ be the sentiment model head at the current step, and $\hat{\tau}$ be our estimated treatment effect from the RieszNet model head, we get that our full, regularized loss equation is

$$\mathcal{L} + \lambda \sum_i (f(X_i, 1) - f(X_i, 0) - \hat{\tau}_i)^2 \tag{5}$$

To compute $f(X_i, 1)$ and $f(X_i, 0)$ we create counterfactual inputs $(X, 1 - T)$ from the original inputs X, T. With $\hat{\tau}$ as our estimated treatment effect, we can compute a simple synthetic sentiment

$$Y^c = \begin{cases} Y + \hat{\tau} & \text{if } T = 0 \\ Y - \hat{\tau} & \text{if } T = 1 \end{cases}$$

However, this data augmentation technique is insufficient when within our text input Z we have that X and T are correlated through the confounding writer intention C. Simply applying average treatment effects will yield biased counterfactual sentiment values. A more robust synthetic counterfactual can be produced: instead of intervening on the treatment T, we use our observed X values to calculate expected values for C and T. In particular, we learn a function $\hat{C} = h(X)$ to estimate the latent C, which we then use to simulate new values of T based on $Pr[T|\hat{C}]$ and some noise. These new values of T as well as our estimated treatment effects can then be used to calculate new outcomes Y as previously, and generate synthetic counterfactual examples. This method ensures that we are faithful to our causal data distribution, thus giving us accurate fake examples that mirror reality.

In the end, we get a counterfactual dataset $\mathcal{D}^c$ containing one counterfactual observation for each sample in our original dataset $\mathcal{D}$. Then, we fit our model so that it minimizes our loss on the actual data as well as on the counterfactual data, with the amount of weight given to each dependent on the regularization hyperparameter λ. Specifically, we fine-tine our language model with the following objective:

$$\arg\min_f \left[\mathbb{E}_{\mathcal{D}}[\mathcal{L}(Y, f(Z)] + \lambda \mathbb{E}_{\mathcal{D}^\rfloor}[\mathcal{L}(Y, f(Z))] \right]$$

Thus, as $\lambda \to 0$, we revert back to normal fine-tuning using our dataset. But as $\lambda \to \infty$, we tend to only focus on the counterfactual data and not the original. This formulation is more efficient than the loss in Equation 5, as we don't iterate over the counterfactual dataset on every step of stochastic gradient descent.

2.1.3. Model Training Pipeline

As demonstrated in both (Shi et al., 2019) and (Chernozhukov et al., 2021), an effective architecture for causally regularized NLP models consists of a single transformer-based backbone such as BERT, concatenated with two different final layer linear heads, one to learn the Riesz Representer and another to learn the downstream task (e.g. sentiment prediction). This eases learning by promoting shared hidden state information between the two model heads. The Riesz Representer head can be trained by minimizing the Riez Representation Theorem-derived loss suggested by (Chernozhukov et al., 2021). This can then be used to estimate true feature effects and produce counterfactual data to augment the training dataset. Finally, the sentiment (or other downstream task) head can be trained by minimizing the cross-entropy between predicted and true sentiments both in the original dataset and in the augmented dataset.

2.1.4. Results

(Bansal & Sharma, 2023) evaluate their RieszNet counterfactual data augmentation pipeline for two sentiment analysis datasets: CivilComments toxicity detection (Borkan et al., 2019) and IMDB movie review sentiment (Maas et al., 2011). In both settings, the novel causal regularization method performs on par with standard training baselines overall. However, when considering specific groups of examples, for example a minority groups, the novel method improves individual group accuracies by up to 10%.

2.2. Related Works

Other works have investigated causal effect regularization through interventional and heuristic methods, achieving promising results as well.

In an earlier work, (Bansal et al., 2023) investigate an intervention-based (masking) method for regularizing fine-tuned encoder models to restrain word-level causal effects to be similar to the base encoder model. Bansal et al. find that fine-tuned recommendation encoders perform worse than base models on out-of-distribution (OOD) data such as new item categories or queries, and so they introduce an intervention-based regularizer to restrain the causal effect of individual words in a fine-tuned model to be similar to that of the BERT (or similar) base encoder. Their results show that causal-based regularization can reduce the deleterious effects of fine-tuning and are recommendable for mild to moderate distribution shifts in the deployment of a recommendation system based on short text matching.

Similarly, (Pang et al., 2021) aim to minimize spurious noise in text inputs towards developing a superior semantic encoder for text matching. In contrast to (Bansal et al., 2023), their approach is algorithmic, using PageRank to filter noisy chunks of words in the attention blocks of a Transformer-based encoder model. Each pair of sentences is chopped into chunks, and those chunks are fed into a graph where each node is a text chunk and each edge is weighted based on the semantic similarity between chunks. Low scoring chunks are considered heuristically to be likely spurious content, as suggested by (Pang et al., 2017). The work of (Pang et al., 2021) further differs from both (Bansal & Sharma, 2023) and (Bansal et al., 2023) in that it focuses on long texts, which is a generally harder problem both pedagogically and computationally. Pang et al.'s results demonstrate that isolating semantically relevant inputs can improve not only out-of-distribution performance, but overall text-matching performance.

Whereas both (Bansal & Sharma, 2023) and (Bansal et al., 2023) regularize language models during fine-tuning by adding a loss term proportional to the difference in causal effects between the pre-trained and fine-tuned model, (Pang et al., 2021) uses an approach that filters out unimportant chunks of text for each sentence pair. This represents a fundamental difference in the regularization approach and suggests that both data filtering and loss function modifications can be useful causal regularization strategies. However, it seems likely that data filtering might be more effective for tasks involving long texts. Further, based on the results of (Bansal & Sharma, 2023), data filtering approaches such as this are likely to harm performance on certain examples where the extra data might have provided useful signal for the downstream prediction task.

Most importantly, (Pang et al., 2021)'s results suggest that causal (or heuristically causal) regularization methods are likely to be effective in the long-form text matching domain as well. Given the current research interest in mining and modeling of huge texts, this result is very promising. A better understanding of causal regularization methods might prove hugely impactful not only for short-form text maching and sentiment analysis, but also for larger scale LLM training and large-scale text data mining.

(Zhang et al., 2021) also investigate the separation of causal and spurious components in encoder systems, but for the task of sequential user interactions prediction. Zhang et al. learn user encoder f_θ and item encoder g_θ to predict a user u's interaction $y_{u,t}$ at time t based on their historical interaction sequence $x_{u,t} = y_{u,[:t]}$. Zhang et al. identify 'indispensable' item-level concepts by scoring the semantic item-encoding (g_θ) similarity between items throughout a user's interaction history, filtering out the bottom half of 'dispensable' items (heuristically identified as noise due to low mutual similarity), and then using contrastive learning with synthetic counterfactuals to ensure that users are pushed further apart by differences in 'indispensable' item history as compared with differences in 'dispensable' item history. Their results outperform state-of-the-art (SOTA) sequential recommenders, demonstrating that methods for isolation of causal components to denoise encodings can provide benefits for other kinds of recommender systems, not just text-matching systems. Further, their work suggests future directions for text-matching; it may be interesting to investigate whether similar contrastive learning on 'indispensible' text components can regularize text query embeddings. Alternatively, a similar method might be used to inform the heuristic selection of treatment words T upon which we train the RieszNet feature effect model.

3. Literature Summary

Here we have summarized in detail a method proposed by (Bansal & Sharma, 2023) to regularize learned text feature effects to match causal feature effects estimated via RieszNet (Chernozhukov et al., 2021), as well as several earlier methods using heuristically causal methods to improve the performance of sentiment, encoder, and recommendation models utilizing signal from text data. Causal regularization in the field of NLP is still incredibly new, and as such we hope that our replication and comparison of these methods will yield helpful insights to guide future research.

References

Bansal, P. and Sharma, A. Controlling learned effects to reduce spurious correlations in text classifiers. *arXiv preprint arXiv:2305.16863*, Jun 2023. URL https://doi.org/10.48550/arXiv.2305.16863. Accepted to ACL 2023, version 2.

Bansal, P., Prabhu, Y., Kiciman, E., and Sharma, A. Using interventions to improve out-of-distribution generalization of text-matching recommendation systems, 2023. URL https://arxiv.org/abs/2210.10636.

Borkan, D., Dixon, J., Sorensen, J., Thain, N., and Vasserman, L. Civil comments dataset, 2019. URL https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/data. A dataset of public comments with toxicity labels, originally collected from a commenting platform for online news and articles.

Chernozhukov, V., Newey, W. K., Quintas-Martinez, V., and Syrgkanis, V. Riesznet and forestriesz: Automatic debiased machine learning with neural nets and random forests. *arXiv preprint arXiv:2110.03031*, Jun 2021. URL https://arxiv.org/abs/2110.03031. Accepted for a long presentation at ICML, version 3.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019. URL https://arxiv.org/abs/1810.04805.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. URL https://arxiv.org/abs/1907.11692.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Imdb movie reviews dataset, 2011. URL https://ai.stanford.edu/~amaas/data/sentiment/. A dataset for binary sentiment classification, consisting of 50,000 highly polar movie reviews.

Pang, L., Lan, Y., Guo, J., Xu, J., Xu, J., and Cheng, X. Deeprank: A new deep architecture for relevance ranking in information retrieval. *arXiv preprint arXiv:1710.05649*, Jul 2017. doi: 10.1145/3132847.3132914. URL https://doi.org/10.48550/arXiv.1710.05649. Published as a conference paper at CIKM 2017, version 2.

Pang, L., Lan, Y., and Cheng, X. Match-ignition: Plugging pagerank into transformer for long-form text matching.

arXiv preprint arXiv:2101.06423, 2021. URL https://arxiv.org/abs/2101.06423. Last revised 17 Aug 2021 (v2).

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. URL https://arxiv.org/abs/1910.01108.

Shi, C., Blei, D. M., and Veitch, V. Adapting neural networks for the estimation of treatment effects. *arXiv preprint arXiv:1906.02120*, Oct 2019. URL https://doi.org/10.48550/arXiv.1906.02120. Version 2.

Wu, T., Ribeiro, M. T., Heer, J., and Weld, D. S. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. *arXiv preprint arXiv:2101.00288*, Jun 2021. URL https://doi.org/10.48550/arXiv.2101.00288. ACL 2021, main conference, long paper.

Zhang, S., Yao, D., Zhao, Z., Chua, T.-s., and Wu, F. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*, pp. 11, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 978-1-4503-8037-9. doi: 10.1145/3404835.3462908. URL https://doi.org/10.1145/3404835.3462908.

Zhang, W., Deng, Y., Liu, B., Pan, S. J., and Bing, L. Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*, May 2023. URL https://doi.org/10.48550/arXiv.2305.15005. Version 1.