CaM-Gen: Causally-aware Guided Text Generation

Anonymous ACL submission

Abstract

Content is created for a well-defined purpose, often described by a metric or signal represented in the form of structured information. The relationship between the goal (metrics) of target content and the content itself is non-trivial. While large-scale language models show promising text generation capabilities, guiding the generated text with external metrics is challenging. These metrics and content tend to have inherent relationships and not all of them may be of consequence. We introduce CaM-Gen: Causally-aware Generative Networks guided by user-defined target metrics incorporating the causal relationships between the metric and content features. We leverage causal inference techniques to identify causally significant aspects of a text that lead to the target metric and then explicitly guide generative models towards these by a feedback mechanism. We propose this mechanism for variational autoencoder and Transformer-based generative models. The proposed models beat baselines in terms of the target metric control while maintaining fluency and language quality of the generated text. To the best of our knowledge, this is one of the early attempts at controlled generation incorporating a metric guide using causal inference.

1 Introduction

Most content is created for a well defined goal. For example, a blog writer often publishes articles to gain popularity and trigger conversations, and a columnist may write an opinionated piece to gather feedback. In marketing applications, these goals are business objectives that need to be optimised using the content shared with the customers. The validation of whether the goal was met or not is done by tracking metrics that capture the reader behavior. In social media, metrics include number of comments, likes, or shares whereas for a publishing house they are the number of views and readers. These engagement metrics (hereafter, metrics) are proxy for target goals. Based on historical content, textual content characteristics that successfully achieve the desired metrics can be assessed (Tan et al., 2019; Verma et al., 2020). Guiding text generation models by these signals is important for meeting the required goals.

While recent neural language models have shown tremendous success towards fluent text generation (Radford et al., 2018; Devlin et al., 2019), achieving controlled, goal-specific generation is challenging. There have been work on text generation controlling for style, topic, or size (Keskar et al., 2019). These methods are able to leverage content characteristics that are common between the definition of goal (i.e. control) and the text. However, for metrics that are not explicit in the text, controlled generation is non-trivial to codify. The challenge is introduced due to the fact that for external metrics, there is a need to first identify the relationship between the content characteristics and the metric and then to explicitly introduce a guide/constraint enabling the generator to learn the desired content properties. Contrary to style, these choices might be difficult for a layman to manually identify and input to the generative models.

Textual content is an amalgam of various linguistic features (Verma and Srinivasan, 2019) – lexical, pertaining to word choices; semantics, concerned with the meaning; syntactic, relating to parts of speech tags; and surface-level features, comprising punctuation, word count, sentence count, etc. As is expounded in causal literature, a correlation analysis between these features and the target outcome is insufficient (Aldrich, 1995). For a finer control, we need to identify features that have direct and significant impact on the outcome metric and guide the generation along those features. A causally significant relationship helps encode the ‘if this, then that’ logic; adding such a guide for the generator can help ensure on-metric generation.

We discuss two modeling frameworks for metric-guided generation – conditional variational autoencoders (Sohn et al., 2015) and Transformer-based language models (Vaswani et al., 2017). We propose a modified graph for causal guidance in the conditional variational autoencoders (CVAE). We also introduce a causal guidance framework in Transformer-based language models using causal losses for explicit feedback on causal features.

Our key contributions are introducing causal guidance frameworks for metric-guided, controlled text generation in CVAE and Transformer-based generative models. We experiment with a new dataset of news articles related to COVID-19 along with the NYT-comments dataset,[1] showing improved performance against baseline methods. To the best of our knowledge, this is one of the first attempts towards controlled generation on engagement metrics and inclusion of causal guidance for controlled generation in generative models.

2 Related Work

The literature on text generation spans various generative models, including variational autoencdoer (VAEs), generative adversarial networks (GANs), and sequential models. VAEs have been used for unconditional (Bowman et al., 2016), as well as constrained text generation (Zhang et al., 2016; Pagnoni et al., 2018). Pagnoni et al. (2018) generate a sentence sequence y conditioned on the input sentence for machine translation, thus mimicing a sequence-to-sequence model. Hu et al. (2017) control sentiment and tense in text generation using discriminators with VAEs. Zhao et al. (2017) introduce an additional reconstruction network in CVAEs for controlling linguistic features in dialog generation. As we show in our experiments, this does not adapt well to controlled generation where the relationship with target goal is not as explicit in text. We identify these nuanced relationships between text and underlying goal and enable explicit control over text features influencing the target outcome by modifying the VAE graph.

While VAEs enable controlled generation, they do not generate fluent language with limited data. Large Transformer-based language models (Radford et al., 2018; Devlin et al., 2019) have shown efficacy in generating fluent language, allowing for fine-tuning for specific tasks on a smaller dataset while maintaining good language quality. Keskar et al. (2019) introduce style control, such as domain (books, wikipedia, etc.), by conditioning the generated distribution on the style token y, i.e. $p(x|y) = \prod_{i=1}^{n} p(x_i|x_{<i}, y)$. The language model learns the conditional probability $p(x_i|x_{<i}, y)$ by training on sequences of raw text prepended with the style control. This approach provides only weak control, especially if the variation in textual features for the same target metric is large. Zeng et al. (2020) enable finer control over generation space by introducing the control y in various internal layers of Transformer network. Singh et al. (2020) control for a combination of lexical styles to reproduce author's styles using a RL framework for Transformer-based language models. While style is well reflected in the choice of vocabulary and language distribution, the difference in the language distribution is not as apparent for an external metric as control. We observe that the external metric is more influenced by various syntactic and surface-level text features, as opposed to the underlying vocabulary. We achieve finer control over these by a causally-aware generative language model.

Causal Inference. Causal analysis entails dissecting the effect of specific treatments on outcome variables, while controlling for other confounding factors. These methods are widely used in fields such as marketing, advertising, healthcare and more recently textual analysis. Paul (2017) employ a propensity matching algorithm to identify causal association between the text with its sentiment classification. Veitch et al. (2020) study the effect of presence of elements such as theorems on the acceptance rate of papers or the effect of gender on the popularity of social media posts. Tan et al. (2019), Verma et al. (2020) use a doubly robust method on propensity-based matching to analyze the effect of specific content features on the user response by accounting for the confounding effect of the content and other textual features. They use adaptive and flexible multi-layer neural networks to model potential outcomes. We adapt their technique to uncover causal effect of various syntactic and surface-level features in textual data and then use these for guiding causally-aware generation.

3 Causal Features Identification

To incorporate finer control over generation of text to achieve specific target metric, we first identify features that contribute to the respective outcome.

[1] https://www.kaggle.com/aashita/nyt-comments

Here, the outcome metric is the target value we wish to control. We consider various syntactic (e.g. *noun/adjective count*) and surface-level textual features (e.g. *word/sentence/paragraph count*) and measure their effect on the metric. Consider two text choices – S1: "*The dog sprinted ahead so fast, the girl had much hard time keeping up with it.*", S2: "*The dog sprinted fast ahead. The girl panted trying to keep up.*"; both meaningful and reasonable generations. Say, content with less words per sentence and more sentences is better liked. In this case, *word count* would have negative effect on outcome metric and *sentence count* would have a positive effect. Thus, the model should generate shorter sentences, resulting in S2.[2] To this end, we perform a causal analysis to identify how changing a certain text feature will affect the outcome metric.

The hypothetical change in an input feature of observed data is defined as an intervention, and the input feature in question is termed as the *treatment variable* (t). For a binary treatment, the effect of treatment on the outcome (y) is defined as $y_1(x_i) - y_0(x_i)$ for the i^{th} text sample, where y_0 represents outcome in absence of treatment and y_1 represents outcome when treatment is applied and x_i are the other covariates (text features). The average treatment effect (ATE) is the expected effect of providing the treatment (i.e. including a specific feature) and is given by $\mathbf{E}[y_1(x_i) - y_0(x_i)]$. This can not be directly calculated as we do not know what the outcome is if a certain part of text is changed in a certain way, i.e., $y_0(x_i)$ and $y_1(x_i)$ is not known for the same i. Moreover, in observed data, the treatment assignment is not independent of baseline covariates. We account for this by employing a propensity-based scoring, which serves to balance treatment assignment in treated and untreated groups (Austin, 2011).

The propensity score is defined as the probability of treatment assignment conditional on baseline covariates, i.e. $\pi(x_i) = p(t_i = 1|x_i)$. We employ multi-layer neural networks to approximate propensity scores (Tan et al., 2019). The average treatment effect (ATE) can be estimated by inverse propensity treatment weighing (IPTW) (Austin, 2011), where each outcome is weighed by inverse probability of receiving the corresponding treatment. Thus,

$$ATE = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{t_i y_i}{\pi(x_i)} - \frac{(1-t_i)y_i}{1-\pi(x_i)}\right] \quad (1)$$

For a doubly robust estimate,[3] we augment IPTW with potential outcome model (Funk et al., 2011). The potential outcome models estimate outcomes if treatment is applied (t=1) or not applied (t=0), given the other covariates. We model potential outcome using two neural networks (for t=0, 1), trained to minimize mean squared error in predicted and actual outcome in observed articles with t=1 and t=0, respectively. The expected outcome in presence of the treatment feature is then a function of the observed outcome with treatment for the treated group and predicted outcome with treatment for the untreated group, given article features, weighted by a function of the propensity scores.

$$y_1(x_i) = \frac{t_i y_i}{\pi(x_i)} - \frac{t_i - \pi(x_i)}{\pi(x_i)}\hat{y}_1(x_i) \quad (2)$$

Similarly, the overall response in the absence of treatment is estimated as

$$y_0(x_i) = \frac{(1-t_i)y_i}{1-\pi(x_i)} + \frac{t_i - \pi(x_i)}{1-\pi(x_i)}\hat{y}_0(x_i) \quad (3)$$

The average effect of the treatment feature on the outcome is estimated as the mean of the difference of expected outcome with and without treatment.

$$ATE = \frac{1}{n}\sum_{i=1}^{n}(y_1(x_i) - y_0(x_i)) \quad (4)$$

This provides an estimate of which text features have the most impact on the outcome (target) metric.[4] The ATE of continuous treatment features can be estimated in a similar fashion, assuming a normal treatment distribution (Tan et al., 2019).

4 CaM-Gen

We present a causally-aware text generation method in VAE and Transformer-based models. We consider metric-guided generation in VAE (section 4.1) and then augment this conditional VAE (CVAE) with a causal graph to incorporate causally significant features in generative process (section 4.2). In Transformer-based text generation, we condition on target metric by modifying Transformer layers (section 4.3) and then introduce a causal feedback for controlled generation (section 4.4).

[2]This example uses semantically similar text pieces for illustration. Generation task discussed in paper does not have such parallel instances

[3]See Appendix A for more details

[4]List of features and their ATE is available in Appendix E

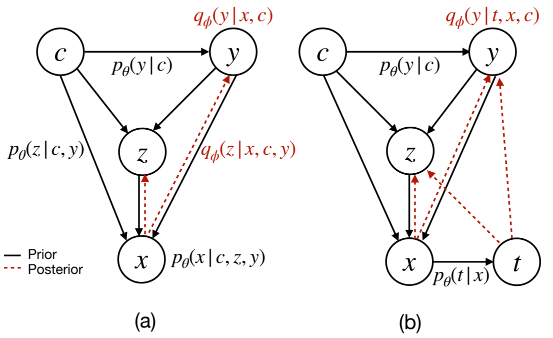


Figure 1: VAE Graph - (a) Conditional generation, (b) Causal feedback in conditional generation

4.1 Conditional Variational Autoencoder

We first adapt the CVAE architecture, inspired by (Zhao et al., 2017). As opposed to generating a response to previous utterances, we model the conditional generation as a next sentence generation task – generate the next sentence x, given the previous context c, and the target metric y.

We consider a latent variable z that captures the latent distribution over the generation space. We estimate z using the prior network $p(z|c, y)$, assuming a multi-variate Gaussian distribution. The sentence x is generated by the decoder network $p_\theta(x|c, z, y)$. The prior of the outcome metric is approximated using $p_\theta(y|c)$. Since the outcome metric depends on both the generated x and the given context c, we do not assume independence between the inputs c and y. We consider two recognition networks $q_\phi(y|x, c)$ and $q_\phi(z|x, c, y)$ to approximate the true posteriors $p_\theta(y|x, c)$ and $p_\theta(z|x, c, y)$ (graph as shown in Fig. 1a). The CVAE network can be trained using the variational lower bound[5]

$$\begin{aligned}\mathcal{L}_{\mathrm{V_{nc}}}(\theta, \phi; x, c, y) = {} & \mathbf{E}_{q_\phi(z,y|x,c)}[\log p_\theta(x|c, z, y)] \\ & - \mathbf{E}_{q_\phi(y|x,c)}\,\mathrm{KL}[q_\phi(z|x, c, y)||p_\theta(z|c, y)] \\ & - \mathrm{KL}[q_\phi(y|x, c)||p_\theta(y|c)]\end{aligned} \tag{5}$$

Intuitively, the first term is the reconstruction loss, the second term aligns latent variable z w.r.t. metric y and the generated text x, and the last term ensures that generation adheres to the target metric.

4.2 Causal CVAE

The above conditional generation controls the target metric as a whole, but does not directly influence specific aspects of the text that impact the outcome metric. Ideally, the latent variable z would implicitly learn these during training. However, in practice this is not so, especially in the case of limited data and multiple confounders. Besides aligning the latent space z w.r.t. x, we enable explicit causal guidance by aligning the latent space to the *causally significant features* t (features significantly impacting the target metric) in the generated text. Causal feature vector t comprises features with ATE (section 3) higher than a threshold.[6]

The posterior distribution of latent variable z is now estimated as $q_\phi(z|t, x, c, y)$. By definition, the outcome metric distribution will be affected by the causal features t in the generated x. The posterior distribution for outcome metric y can hence be approximated as $q_\phi(y|t, x, c)$. The feedback of these causal effects is propagated through the network by minimizing the KL divergence between the prior distribution $p_\theta(y|c)$ and $q_\phi(y|t, x, c)$ (Fig. 1b). The loss function[7] for causal CVAE is

$$\begin{aligned}\mathcal{L}_{\mathrm{V_c}}(\theta, \phi; t, x, c, y) = {} & \mathbf{E}_{q_\phi(z,y|t,x,c)}[\log p_\theta(x|c, z, y)] \\ & - \mathbf{E}_{q_\phi(y|t,x,c)}\,\mathrm{KL}[q_\phi(z|t, x, c)||p_\theta(z|c, y)] \\ & + \mathbf{E}_{q_\phi(z,y|t,x,c)}[\log p_\theta(t|x, c, z, y)] \\ & - \mathrm{KL}[q_\phi(y|t, x, c)||p_\theta(y|c)]\end{aligned} \tag{6}$$

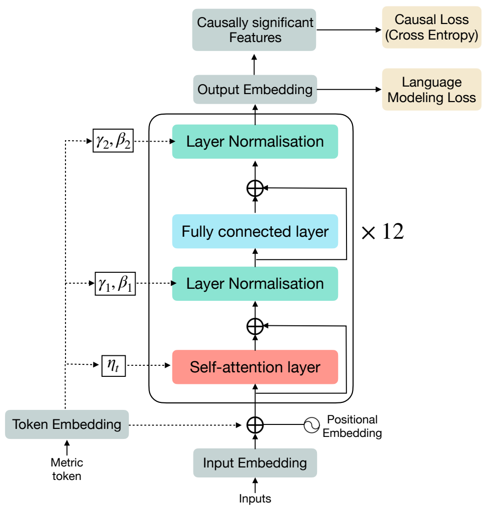


Figure 2: CaM-Gen: Transformer

4.3 Conditional generation in Transformer

The proposed Transformer model is based on the GPT-2 architecture (Radford et al., 2018), which is trained on language modeling loss for predicting the next token given all the previous tokens. The model is first pre-trained with language modeling objective on a large corpora to build understanding of language distribution enabling it to generate

[5]Proof included in Appendix B.1

[6]Significance threshold are chosen empirically, section 5.2

[7]Proof included in Appendix B.2

coherent text. Although fine-tuning with the same objective shifts the language distribution of generated text towards the fine-tuning corpus, explicitly controlling for a target metric is more nuanced. To introduce this explicit control, we use the metric to modify self-attention and normalisation layers in the Transformer blocks (Zeng et al., 2020), as shown in Fig. 2.[8] In the former, attention weights of Transformer blocks are biased towards the target by changing the query vector in attention mechanism with the affine transformation of y. In the latter, the scale and bias parameters of layer normalisation are replaced by functions of y. This ensures that the target information does not *wash away* (Park et al., 2019) and is preserved through the normalisation layers. The generative model is trained with the language modeling loss given by,

$$\mathcal{L}_G = \mathbf{E}_{x,y}\Big[-\sum_{i=1}^{n}\log P_G(x_i|x_{<i}, y)\Big] \quad (7)$$

We introduce a metric loss as a feedback for the degree of metric control achieved during generation. This is defined as the cross entropy loss between the input target metric and the projected metric for the generated text. The latter is calculated using a fastText (Joulin et al., 2016) classifier trained on the outcome on the historical text across various metrics. Such a classifier, which predicts the engagement on held-out test set with high confidence, serves as an indicator of expected engagement on generated text. The metric loss is

$$\mathcal{L}_{\text{metric}} = \mathbf{E}_{x,y,\tilde{x}=G(x,y)}\big[-y\log P_F(y|\tilde{x})\big] \quad (8)$$

$P_F(y|\tilde{x})$ denotes the probability of the outcome of the generated text $\tilde{x}$ to be the target metric y.

4.4 Causal guidance in Generative Model

The addition of the target metric as control in input embedding, self-attention mechanism or layer normalisation guides the generative model towards the target metric by shifting the language distribution of the generative model. However, an explicit guidance of different aspects of text that influence the outcome metric is absent. To achieve this, we add causal guidance in the generation process. We introduce a causal loss in the above Transformer model to lead the generated text to adopt causally significant features (t). The output tokens generated from the Transformer are fed into an SVM that extracts these features from the generated text. The model is then trained with the additional objective of minimizing the cross entropy loss between the target metric and the predicted outcome metric based on these causal features in output text.

$$\mathcal{L}_{\text{causal}} = \mathbf{E}_{x,y,\tilde{x}=G(x,y)}\big[-y\log P_{F'}(y|t(\tilde{x}))\big] \quad (9)$$

where $P_{F'}$ is the expected outcome metric given the causal features $t(x)$, estimated using a fastText model trained on causal features extracted from observed data. The proposed causal loss aims at ensuring that the causal features in generated text adheres to target metric, by isolating the effect of causal features in text from its context.

The resultant loss optimized by the proposed model is a weighted sum of these losses, i.e. $\mathcal{L} = \lambda_G\mathcal{L}_G + \lambda_{\text{metric}}\mathcal{L}_{\text{metric}} + \lambda_{\text{causal}}\mathcal{L}_{\text{causal}}$, where λ_G, λ_{metric}, λ_{causal} are weights for different losses selected by hyper-parameter tuning on validation set.

Equivalence between Causal CVAE and Transformer: In the VAE-based models, we consider the context c and discuss the next sentence (x) generation task. At token-level, c is similar to the context $x_{<i}$ in the next token (x_i) generation objective. Thus, the decoding term in CVAE loss (first term in Eq. 5) is equivalent to $\mathcal{L}_G$ (Eq. 7) in the Transformer model. Similarly, the KL divergence between metric prior and posterior distribution in $\mathcal{L}_{V_{nc}}$ (last term in Eq. 5) can be equated to the metric loss in Eq. 8. The corresponding term in $\mathcal{L}_{V_c}$ (last term in Eq. 6) serves as the causal loss, similar to $\mathcal{L}_{causal}$ in Eq. 9. With minor adjustments, this causal guidance framework can be extended to other generative networks in a similar fashion.

Dataset	Metric	Low	Med.	High
Webhose (Total:39192)	Participation	20482	9181	9529
	Replies	20440	9262	9490
NYT (Total:9403)	Comment	3160	3075	3168
	Upvote	3122	3126	3155

Table 1: Number of samples in across metrics

5 Experiments

Datasets. We experiment with 2 text datasets: NYT comments, which comprises articles with comments and metrics such as upvote and comments count and the Webhose[9] dataset comprising of articles and comments with metrics such as total participation on articles, replies count, and various

[8] η, γ, β are the scale/bias parameters in respective layers (details in Appendix C)

[9] https://webhose.io/free-datasets/news-articles-that-mention-corona-virus/

Metric/ Dataset	Model	Variation	Control (↑) % accuracy	Perplexity (↓)	BLEURT (↑)	ROUGE (↑) 1	2	L
Participation (Webhose)	Transformer	Baseline GPT-2	51.93	16.27	-0.98	0.010	0.0	0.002
		$\mathcal{L}_G$	59.94	15.14	-0.81	0.110	0.013	0.085
		$\mathcal{L}_G + \mathcal{L}_{metric}$	62.78	**3.03**	-0.83	0.113	0.012	0.074
		Causal model (our)	**69.86**	3.19	-0.79	**0.201**	**0.022**	**0.130**
	CVAE	Baseline CVAE	51.37	34.37	-0.80	0.113	0.010	0.063
		metric-guided	54.43	28.21	-0.69	0.179	0.017	0.099
		Causal model (our)	55.66	30.03	**-0.71**	0.130	0.012	0.079
Replies (Webhose)	Transformer	Baseline GPT-2	51.79	17.76	-0.91	0.005	0.0	0.005
		$\mathcal{L}_G$	59.87	13.94	-0.85	0.051	0.004	0.043
		$\mathcal{L}_G + \mathcal{L}_{metric}$	60.17	3.48	-0.79	0.107	0.011	0.070
		Causal model (our)	**68.27**	**3.12**	-0.81	**0.211**	**0.022**	**0.133**
	CVAE	Baseline CVAE	50.58	38.41	-0.89	0.046	0.001	0.035
		metric-guided	56.14	20.58	-0.8	0.124	0.002	0.072
		Causal model (our)	60.00	30.24	**-0.76**	0.031	0.001	0.022
Comments (NYT)	Transformer	Baseline GPT-2	37.24	27.45	-0.83	**0.140**	**0.088**	**0.135**
		$\mathcal{L}_G$	49.85	23.59	-0.87	0.095	0.051	0.088
		$\mathcal{L}_G + \mathcal{L}_{metric}$	53.82	14.99	-0.89	0.10	0.011	0.052
		Causal model (our)	**54.36**	**13.18**	**-0.81**	0.10	0.01	0.049
	CVAE	Baseline CVAE	39.12	58.35	-1.41	0.059	0.002	0.031
		metric-guided	44.42	41.64	-1.32	0.069	0.003	0.036
		Causal model (our)	54.59	40.02	-1.29	0.064	0.003	0.032
Upvotes (NYT)	Transformer	Baseline GPT-2	39.49	27.44	-0.83	**0.132**	**0.080**	**0.127**
		$\mathcal{L}_G$	46.02	23.57	-0.88	0.077	0.032	0.070
		$\mathcal{L}_G + \mathcal{L}_{metric}$	53.66	14.93	-0.82	0.110	0.011	0.053
		Causal model (our)	**53.96**	**13.19**	**-0.80**	0.103	0.010	0.051
	CVAE	Baseline CVAE	37.06	72.68	-0.89	0.057	0.002	0.031
		metric-guided	43.21	65.94	-0.84	0.064	0.002	0.036
		Causal model (our)	59.54	57.70	-0.84	0.056	0.001	0.030

Table 2: Automatic Evaluation for Webhose (Participation, Reply count) and NYT (Comments, Upvotes) Datasets. The causal Transformer model beats all other methods on metric control while achieving comparable fluency.

social media reactions for these articles. These metrics are used as target goal for article text generation. We filter and pre-process[10] this data resulting in 39k article data which we use for our training with a train-dev-test split of 80-10-10 (Table 1). We categorize the target metrics into high, medium, and low classes, resulting in categorical target goal (e.g. high/ low replies count).

Training details. For causal model, we use two sequential feed forward neural networks with 5 dense layers of size 128, each followed by an activation layer, for the treatment and potential outcome network trained with Adam optimizer (Kingma and Ba, 2014). The parts of speech (POS) are extracted using the POS tagging in textblob[11] library. Both treatment and potential outcome networks are trained on 90-10 train-test split over 10 epochs.

For CVAE, we use a bidirectional recurrent neural network (bi-RNN), which encodes each context sentence to a fixed 300-sized vector. We pass these vectors through another GRU network with one hidden-layer of 600-dimension, resulting in the context vector c. The decoder network is also a one-layer GRU with dimensionality 400. The end-to-end model is trained with an Adam optimizer.

We use a Transformer model with 16 multi-attention heads with latent dimension of 768 and a vocabulary size of 50527 with BPE encoding (Sennrich et al., 2016). We use the GPT-2 (Radford et al., 2018) model with $117M$ parameters pre-trained on the WebText dataset to initialize our model and then fine-tune it with NYT and Webhose datasets using our causal metric-guided framework. For causal variants, the causal vector t is extracted from the generated text based on a pre-determined list of causally-significant features (identified beforehand using ATE analysis in section 3).

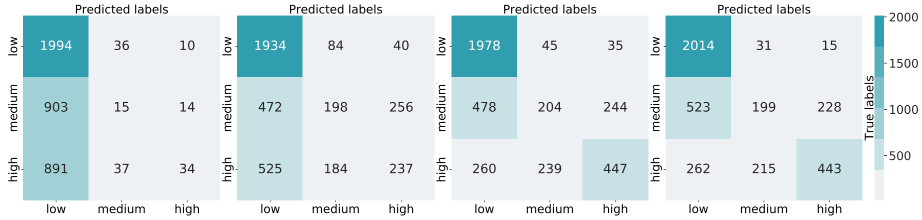
5.1 Evaluation metrics

Control: We measure target control accuracy against predicted outcome metric in the generated text using fastText classifiers trained on available data. The classifiers have test accuracy of 79.8%, 81.4%, 80% and 79.9% for participation, replies, comment, upvotes counts, respectively.

Fluency: We measure the text fluency and the

[10] Preprocessing details in Appendix D.

[11] https://textblob.readthedocs.io



(a) Baseline GPT2 (b) $\mathcal{L}_G$ (c) $\mathcal{L}_G + \mathcal{L}_{metric}$ (d) Causal

Figure 3: Class-wise performance for Transformer-based model variants.

Treatment	Loss	Accuracy
Word Count	0.1791	0.9301
Sent Count	0.2268	0.9266
Noun Count	0.1520	0.9520
Verb Count	0.1437	0.9592
Adjective Count	0.2133	0.9349
Adverb Count	0.1863	0.9431
Pronoun Count	0.1522	0.9377

(a) Propensity scoring model

Outcome metrics	MAE	Accuracy
Upvotes	0.1357	0.9157
Replies count	0.2359	0.8455
Discussion depth	0.2549	0.8322
Comment count	0.1438	0.9104

(b) Potential outcome model

Table 3: Loss and test accuracy of models used for causal effect identification

language model quality using perplexity, ROGUE (Lin, 2004) and BLEURT (Sellam et al., 2020) scores. The perplexity is a measure of likelihood of the generated sentence on a language model. We use a pre-trained GPT-2 model to evaluate text perplexity. A lower value is preferred. BLEURT is a pre-trained evaluation metric based on BERT (Devlin et al., 2019) that provides a robust measure for reference-based text generation. We calculate ROGUE and BLEURT scores against reference articles in test data with same keywords and target.

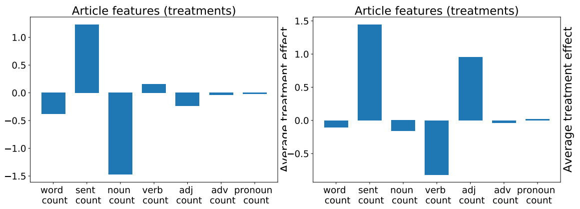
5.2 Results

We compare causal and non-causal variants of the proposed CVAE and Transformer-based models. In the Transformer variants, we evaluate the performance with metric added as a guide in embedding, attention, and normalisation layers, trained with $\mathcal{L}_G$ (Eq. 7). Next, we introduce the metric loss to add feedback for adherence to target metric, training the model with $\mathcal{L}_G + \mathcal{L}_{\text{metric}}$ (Eq. 8). The final proposed causal model is trained with $\mathcal{L}_G + \mathcal{L}_{\text{metric}} + \mathcal{L}_{\text{causal}}$ (Eq. 9). For CVAE, non-causal and causal models are trained with $\mathcal{L}_{V_{nc}}$ and $\mathcal{L}_{V_c}$ (Eq. 5, 6) respectively. We fine-tune a GPT-2 (Radford et al., 2018) model with metric token added to the prompt for control, similar to (Keskar et al., 2019), and use it as a baseline. We also use the method proposed by (Zhao et al., 2017) as the baseline CVAE model.

As seen in Table 2, adding metric as explicit guide improves accuracy both in Transformer and CVAE models, and the causal models outperforms all other variants in the same architecture. Additionally, our variants are at par in text quality, with the Transformer models performing notably better on language fluency than CVAE models. We attribute this to generative pre-training with large corpus equipping Transformer-based language model with fluent language generation. Note that, given the free-form nature of generative task, the references considered for ROUGE and BLEURT are a poor fit as the generation space could be pretty large. This is reflected in low scores for these metrics across all models. Hence, low perplexities are a better indication of generation fluency.

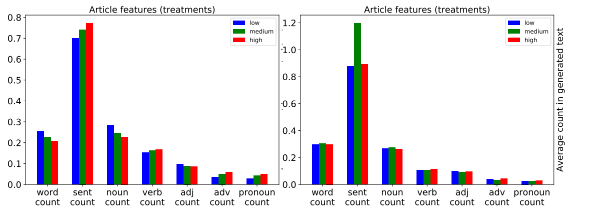
Causal CVAE exhibits better metric control than the non-causal and baseline CVAE, but performs poorer than the causal Transformer model. This could also be an artifact of language quality, since the underlying classifiers are trained on fluent language. Across Transformer variations, addition of metric loss and causal guidance improves metric control, validating our hypothesis. It is interesting to note that the perplexity drops substantially on adding the metric loss in Transformer-based model. This could raise the question on how additional losses (constraints) could result in more fluent generation. We emphasize that, in baseline and all other variants, the constraint is on the target metric. Thus, both baseline GPT-2 and modified Transformer (with only $\mathcal{L}_G$) attempt to align their generation space to this target. An inadequate alignment of generation space to the desired control is likely to result in noisy generations. In that sense, metric/causal do not add more constraints, rather add feedback to meet the specified constraint (goal), leading to more controlled and less noisy generations. This would potentially explain higher perplexities observed in the first two variants.

Class-wise Performance. Table 2 aggregates results across target classes. To compare the performance across high/medium/low class, we record class-wise metric accuracy. Fig. 3 shows confusion matrices for Transformer-based variants with high/medium/low participation count as tar-



(a) Participation Count (b) Replies Count

Figure 4: Average treatment effect of features like word count, sentence count, POS tag counts across metrics.



(a) Causal Model (b) Baseline Model

Figure 5: Comparison of textual features in text generated by causal vs baseline Transformer model

get. Across methods, we observe that controlling for medium target metric is harder than either of the other classes. Compared to the baseline, variants with causal guidance and metric loss show improved performance for both high and low target class. Our proposed causally-guided Transformer model is the best performing model on per class-level as well, confirming the efficacy of our proposed approach across different target classes.

Causal Feature Identification. Table 3 shows the accuracy of the propensity scoring and potential outcome models. Our propensity scoring models have accuracy > 0.92 for all treatment features and the potential outcome model performs well for *Upvote* and *Comment count*. We use these as target metrics in generative models for NYT dataset. Similar analysis on Webhose data yields *Participation* and *Replies count* as target metric. Fig. 4 shows Average Treatment Effect (ATE) of various text features on these outcome metrics. We empirically choose significance level of 0.1 and consider features with ATE of greater than 0.1 (in magnitude) as 'causally significant' features. We include these as causal features in the generative models.

Causal Analysis. We note that the fastText classifiers used for metric evaluation have relatively low accuracy (although much better than a random 33% classification). We attribute this to high variability in the text and unpredictability of resulting engagement. As discussed previously, a causal analysis of historical text accounts for semantic and topical variation. Similarly, a causal analysis of generated data, and subsequent comparison with historical trends, could compensate for any potential inadequacies of classifier-based evaluation. To this end, we perform a causal analysis of the text generated by the baseline and our proposed model.

We generate text with high, medium and low target participation count (*pcount*) as target and record average value of various treatment features (Fig. 5). Here, the word and sentence counts are normalised and POS features are fraction of words with certain POS tag over total number of words in the generated text. We test the adoption of 'causally significant' features in the causal model by analyzing feature distributions of text generated by causal model and baseline Transformer model across classes (high/medium/low). For instance, word count has a negative ATE on *pcount* (Fig. 4a). Thus, we would expect a text with higher word count to have lesser pcount. As seen in Fig. 5a, our causal model with 'high' target *pcount* generated articles with lower word count on average than the causal model with 'low' target (red and blue bars in first group in Fig. 5a respectively). Similar trends are observed across other 'causally-significant' treatment features. In contrast, the text generated by baseline model (Fig. 5b) either do not show significant variation in these features across text generated with high, medium and low target or the difference is inconsistent, reflecting the lack of control over aspects of text in baseline models where generation is only guided by target metric. As these features, by definition, significantly impact the outcome; this analysis adds further confidence in stronger adherence to the target metric in our proposed causal approach over the baseline.

6 Conclusion

We present a framework for causally-aware metric-guided generation in VAE and Transformer-based models. We successfully identify causally significant text features using causal analysis and incorporate them into the generative model. We show that integrating causal guidance in guided generation enables better control over the target metric, while maintaining language quality. Our proposed causally-guided Transformer model shows improved performance across datasets. Moreover, we show that the generated text adheres to these causal features, in line with their observed effect in historic data. This exploration opens up avenues for leveraging causality for controlled generation.

References

John Aldrich. 1995. Correlations genuine and spurious in pearson and yule. *Statist. Sci.*, 10(4):364–376.

Peter C. Austin. 2011. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate Behavioral Research*, 46(3):399–424. PMID: 21818162.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(null):993–1022.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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. Association for Computational Linguistics.

Michele Jonsson Funk, Daniel Westreich, Chris Wiesen, Til Stürmer, M. Alan Brookhart, and Marie Davidian. 2011. Doubly Robust Estimation of Causal Effects. *American Journal of Epidemiology*, 173(7):761–767.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596, International Convention Centre, Sydney, Australia. PMLR.

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

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. *arXiv e-prints*, page arXiv:1909.05858.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Artidoro Pagnoni, Kevin Liu, and Shangyan Li. 2018. Conditional Variational Autoencoder for Neural Machine Translation. *arXiv e-prints*, page arXiv:1812.04405.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Michael J. Paul. 2017. Feature selection as causal inference: Experiments with text classification. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 163–172, Vancouver, Canada. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Hrituraj Singh, Gaurav Verma, and Balaji Vasan Srinivasan. 2020. Incorporating stylistic lexical preferences in generative language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1074–1079, Online. Association for Computational Linguistics.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3483–3491. Curran Associates, Inc.

Fei Tan, Zhi Wei, Abhishek Pani, and Zhenyu Yan. 2019. User response driven content understanding with causal inference. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1324–1329.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Victor Veitch, Dhanya Sridhar, and David Blei. 2020. Adapting text embeddings for causal inference. In *Proceedings of the 36th Conference on Uncertainty in*

Artificial Intelligence (UAI), volume 124 of *Proceedings of Machine Learning Research*, pages 919–928, Virtual. PMLR.

Gaurav Verma and Balaji Vasan Srinivasan. 2019. A lexical, syntactic, and semantic perspective for understanding style in text. *CoRR*, abs/1909.08349.

Gaurav Verma, Balaji Vasan Srinivasan, Shiv Kumar Saini, and Niyati Chhaya. 2020. Modeling causal impact of textual style on a targeted goal. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 59–60, New York, NY, USA. Association for Computing Machinery.

Kuo-Hao Zeng, Mohammad Shoeybi, and Ming-Yu Liu. 2020. Style Example-Guided Text Generation using Generative Adversarial Transformers. *arXiv e-prints*, page arXiv:2003.00674.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 521–530, Austin, Texas. Association for Computational Linguistics.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, Vancouver, Canada. Association for Computational Linguistics.

Ethics Statement. We recognise and acknowledge the ethical implication of a generative framework and are committed to following ethical practices. The results included in the paper are meant to establish academic applicability of the proposed framework alone. The work is not intended to enable fake information. The technology discussed in the paper needs to be coupled with fact checking, de-biasing and review workflows for practical utility.

A Conditional Variational Autoencoder

A.1 Non-Causal CVAE

The graph for non-causal conditional generation using variational autoencoder is shown in Fig. **??**. As discussed in section 4.1, we approximate the intractable posterior distribution $p_\theta(z|x,c,y)$ with the recognition network $q_\phi(z|x,c,y)$, where

$$q_\phi(z|x,c,y) = q_\phi(z,y|x,c)q_\phi(y|x,c) \tag{10}$$

The variational parameters ϕ are chosen such that the approximate posterior distribution $q_\phi(z|x,c,y)$ is as close to the true posterior distribution $p_\theta(z|x,c,y)$ as possible. This is done by minimizing the KL divergence between the two distributions. Thus,

$$\phi^* = \operatorname*{argmin}_\phi \mathrm{KL}[q_\phi(z,y|x,c)||p_\theta(z,y|x,c)], \tag{11}$$

where the KL divergence is given by,

$$\begin{aligned}
&\mathrm{KL}[q_\phi(z,y|x,c)||p_\theta(z,y|x,c) \\
&= \mathbf{E}_{q_\phi(z,y|x,c)}\left[\log \frac{q_\phi(z,y|x,c)}{p_\theta(z,y|x,c)}\right] \\
&= \mathbf{E}_{q_\phi(z,y|x,c)}\Bigg[\log q_\phi(z,y|x,c) \\
&- \log \frac{p_\theta(x,c,z,y)}{p_\theta(x|c)}\Bigg].
\end{aligned} \tag{12}$$

Rearranging equation 12 gives,

$$\begin{aligned}
\log p_\theta(x) &= \mathrm{KL}[q_\phi(z,y|x,c)||p_\theta(z,y|x,c) \\
&+ \mathbf{E}_{q_\phi(z,y|x,c)}\big[\log p_\theta(x,c,z,y) \\
&- \log q_\phi(z,y|x,c)\big]
\end{aligned} \tag{13}$$

We want to minimize the KL divergence term on R.H.S. of equation 13. Since, the KL divergence is ≥ 0, the variational lower bound on the log likelihood $\log p_\theta(x)$ is given by

$$\begin{aligned}
\mathcal{L}(\theta,\phi;x,c,y) &= \mathbf{E}_{q_\phi(z,y|x,c)}\big[\log p_\theta(x,c,z,y) \\
&- \log q_\phi(z,y|x,c)\big] \\
&= \mathbf{E}_{q_\phi(z,y|x,c)}\big[\log[p_\theta(x|c,z,y)p(z,y|c)] \\
&- \log q_\phi(z,y|x,c)\big] \\
&= \mathbf{E}_{q_\phi(z,y|x,c)} \log p_\theta(x|c,z,y) \\
&- \mathrm{KL}\left[q_\phi(z,y|x,c)||p_\theta(z,y|c)\right]
\end{aligned} \tag{14}$$

Using equation 10, we get

$$\begin{aligned}
&\mathrm{KL}\left[q_\phi(z,y|x,c)||p_\theta(z,y|c)\right] \\
&= \mathbf{E}_{q_\phi(y|x,c)} \mathrm{KL}\left[q_\phi(z|x,c,y)||p_\theta(z|c,y)\right] \\
&+ \mathrm{KL}\left[q_\phi(y|x,c)||p_\theta(y|c)\right]
\end{aligned} \tag{15}$$

Replacing in equation 14, we get the variational lower bound for non-causal CVAE as

$$\begin{aligned}
&\mathcal{L}(\theta,\phi;x,c,y) = \mathbf{E}_{q_\phi(z,y|x,c)} \log p_\theta(x|c,z,y) \\
&- \mathbf{E}_{q_\phi(y|x,c)} \mathrm{KL}\left[q_\phi(z|x,c,y)||p_\theta(z|c,y)\right] \\
&- \mathrm{KL}\left[q_\phi(y|x,c)||p_\theta(y|c)\right]
\end{aligned} \tag{16}$$

A.2 Causal CVAE

As discussed in section 4.2, we add causal guidance in CVAE framework by adding the treatment vector t for aligning the latent space of the Variational Autoencoder. The posterior distribution for the causal-CVAE graph in Fig. 1 is approximated by $q_\phi(z|x, c, y)$. Similar to equation 14, we get the variational lower bound for causal CVAE as

$$\begin{aligned}\mathcal{L}(\theta, \phi; t, x, c, y) &= \mathbf{E}_{q_\phi(z,y|t,x,c)}\big[\log p_\theta(t, x, c, z, y) \\ &\quad - \log q_\phi(z, y|t, x, c)\big] \\ &= \mathbf{E}_{q_\phi(z,y|t,x,c)}\big[\log[p_\theta(t|x, c, z, y) \\ &\quad p_\theta(x|c, z, y)p(z, y|c)] \\ &\quad - \log q_\phi(z, y|t, x, c)\big] \\ &= \mathbf{E}_{q_\phi(z,y|t,x,c)} \log p_\theta(t|x, c, z, y) \\ &\quad + \mathbf{E}_{q_\phi(z,y|t,x,c)} \log p_\theta(x|c, z, y) \\ &\quad - \mathrm{KL}\left[q_\phi(z, y|t, x, c)||p_\theta(z, y|c)\right].\end{aligned} \tag{17}$$

The conditional posterior $q_\phi(z, y|t, x, c)$ is given by

$$q_\phi(z|t, x, c, y) = q_\phi(z, y|t, x, c)q_\phi(y|t, x, c). \tag{18}$$

Thus,

$$\begin{aligned}&\mathrm{KL}\left[q_\phi(z, y|t, x, c)||p_\theta(z, y|c)\right] \\ &= \mathbf{E}_{q_\phi(y|t,x,c)} \mathrm{KL}\left[q_\phi(z|t, x, c, y)||p_\theta(z|c, y)\right] \\ &+ \mathrm{KL}\left[q_\phi(y|t, x, c)||p_\theta(y|c)\right].\end{aligned} \tag{19}$$

Using this in equation 17 gives us the variational lower bound for causal CVAE as

$$\begin{aligned}\mathcal{L}(\theta, \phi; t, x, c, y) &= \mathbf{E}_{q_\phi(z,y|t,x,c)} \log p_\theta(t|x, c, z, y) \\ &+ \mathbf{E}_{q_\phi(z,y|t,x,c)} \log p_\theta(x|c, z, y) \\ &- \mathbf{E}_{q_\phi(y|t,x,c)} \mathrm{KL}\left[q_\phi(z|t, x, c, y)||p_\theta(z|c, y)\right] \\ &- \mathrm{KL}\left[q_\phi(y|t, x, c)||p_\theta(y|c)\right]\end{aligned} \tag{20}$$

B Conditional generation in Transformer

As discussed in section 4.3, we modify attention and normalisation layers in a transformer architecture for adding metric as a guide. Inspired by Zeng et al. (2020), we introduce the metric as follows:
(1) Input embedding: The metric control y is directly added to the token and position embeddings of the input to the first transformer layer. This enables control by slanting the input representation towards the target metric.
(2) Self-attention: In self-attention mechanism of transformers, each input token is weighted with respect to other positions in the input. For each token x_t, query q_t, key k_t and value v_t is calculated using learned weight matrices W^Q, W^K and W^V respectively. The attention score for token x_t is computed by a compatibility function of the corresponding query q_t with the keys k_i of other tokens and the attention vector is computed as the weighted average of these attention scores with the value vector v_t. This can be written as

$$\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{21}$$

where d_k is the dimension of the key vector k_t. We modify this attention calculation to introduce the control y by changing the query vector in the above equation to $q_t = \eta_t(y)$, where η_t denoted an affine transformation. Modifying the query vector according to the specific target metric allows for biasing attention weights towards the target and capturing target control in the context representation, which aids in targeted decoding and generation.
(3) Layer Normalisation: Classically, the layer normalisation in transformers is calculated as

$$\mathrm{LayerNorm}(\nu) = \gamma\frac{\nu - \mu}{\sigma} + \beta, \tag{22}$$

where μ and σ are the mean and standard deviation of the elements in ν and γ and β are the scale and bias parameters. The metric control, y, is used to modulate hidden representations of the generative model via normalisation layers. The scale and bias parameters in the layer normalisation are replaced as functions of y, namely $\gamma(y)$ and $\beta(y)$ in the above equation. As discussed in Park et al. (2019), normalisation layer applied on input with same target control would *wash away* the target information captured in the input to normalisation layer. Adding target control in the scale and bias parameter ensures that the control is preserved through the normalisation layers of transformer.
Training details: For fine-tuning, we prepend the input sentence with *metric* identifiers, to keep the input layer unchanged. We, then, extract the prepended metric token and use it to modify attention and normalisation layers as described earlier. The output of final transformer layer is fed into a pre-trained fastText model to estimate the fitment of generated text to the target metric class in the form of metric loss[12]. During inference, the genera-

[12] The computing infrastructure and hyper-parameter details are included in Appendix E

Feature ↓	Average Treatment Effect			
Dataset →	Webhose		NYT	
Metric →	Participation	Replies	Comment	Upvote
Word count	-0.3816	-0.1034	-0.1034	-0.0171
Paragraph count	0.0079	0.0038	0.0025	0.0078
Sentence count	1.2308	1.4453	0.0203	-0.0498
Images Count	NA	NA	0.0279	0.0387
Links Count	NA	NA	-0.0459	-0.0225
Slideshow Count	NA	NA	0.0456	-0.0077
Noun count	-1.4758	-0.1589	-0.0062	-0.0239
Verb count	0.1591	-0.8179	0.0386	0.0214
Adjective count	-0.2364	0.9527	-0.0012	-0.0008
Adverb count	-0.0372	-0.0372	-0.0173	-0.0037
Pronoun count	-0.01949	0.0203	-0.0069	-0.0153

Table 4: Average Treatment Effect of various article features on Comment count and Upvotes count for Webhose and NYT data

tion is conditioned on the prompt, which is a combination of the topic and keywords. During training, the keywords and topic for the article is prepended to the input along with a *{start of text}* token. Thus, the input is *{metric token}+{topic}+{start of keyword token}+{keywords}+{start of text token}+{article text}*. The keywords and topics are available for the NYT dataset for each article, and are extracted from input text using topic modeling (Blei et al., 2003) as described in next section.

C Data Processing

Webhose Covid-19 Dataset: We use the Webhose dataset available at https://webhose.io/free-datasets/news-articles-that-mention-corona-virus/ that has $410,120$ data points in total. We choose the subset of this dataset limited to English. To remove any outliers, we heuristically choose articles with word count more than 30 but less than 5000 words in the article. The data contains engagement on various news articles in form of participation count, replies count and various other social media likes and share metrics. The social media metrics includes PinInterest, LinkedIn, Google+ shares and like, shares and comments on Facebook. Most of these are very sparse in the dataset, for instance, less than $\sim 12k$ data points have Facebook comments as non-zero. Thus, we choose participation count and replies count as good indicators to the engagement on the article and use these as our target metrics. We consider only the articles with participation count > 1, leaving us with 39192 data points in total. The metric value for participation count and replies count vary from $1-297$ and $0-5751$ respectively with a mean and standard deviation of $14.37, 27.90$ and $129.91, 446.71$. To control for these metrics in our models, we convert these to categorical variable with the threshold of 2 and 21 for participation count. The low bucket is the largest bucket with least standard deviation in the value of metric; the medium and high categories have almost same number of data points as shown in Table 1 in the paper. Similarly for replies count, the threshold is 2 and 32 with equal size of medium and high categories.

As mentioned earlier, the context for generative models includes keywords and topic of the article, that acts as "prompt" during inference stage. For webhose data, the keywords are not directly available in the dataset, NYT-comments dataset has keywords. We extract the keywords as top n ($n = 10$) words from the articles using TF-IDF vectors. The topics are extracted by topic modeling using Latent Dirichlet Allocation (LDA) (Blei et al., 2003). We choose 20 topics with a seed of 23 and then represent the topic of each input article as the corresponding topic identifier ranging from 1-20. For transformer-based model, the keyword and topic tokens are added to the pre-trained tokenizer.

D Causal Features

The various textual features considered for causal effect are as listed in Table 4. The average treatment effect on NYT data metrics – Comment count and Upvote count is as shown. Here, the significance level is empirically chosen as 0.01. Thus, features with $|\text{ATE}| > 0.01$ on comment count or upvote count y are included in the corresponding causal generative model. For Webhose data, we choose significance level of 0.1 and consider features with ATE of greater than 0.1 in magnitude as 'causally significant' features.

E Reproducibility checklist

E.1 Hyper-parameters

The causal feature identification models are trained on a train-test split of 90-10, using a random seed 23 with stratified sampling over the outcome values, for over 10 epochs in batches of size of 5.

For transformers, we use HuggingFace[13] implementation of GPT-2 and make the model and training changes as described in the paper. The hyper-parameters are kept the same as the original implementation for uniformity. For the loss term mentioned in equation 11 of the paper, we set λ_G, λ_{metric}, λ_{causal} as 1. We train these models with a batch size of 2 for over 3 epochs. The training time over 4 GPUs was about 14 hours for webhose data and about 5 hours for NYT dataset.

For the CVAE model, we use adam optimizer. We initiate the training with the learning rate of 0.001 with learning rate decay of 0.6. We train the models over 30 epochs with an early stopping criteria of 0.996 threshold.

E.2 Resources

All the training experiments were run on a 4 GPU machine with 64-bit 16 core tesla v100 processor and 100 GB RAM.

[13] https://github.com/huggingface/transformers