Discovering influential text using convolutional neural networks

Anonymous ACL submission

Abstract

Experimental methods for estimating the impacts of text on human evaluation have been widely used in the social sciences. However, researchers in experimental settings are usually limited to testing a small numbers of pre-specified text treatments. While efforts to mine unstructured texts for features that causally affect outcomes have been ongoing in recent years, these models have primarily focused on the topics or specific words of text, which may not always be the mechanism of the effect. In this paper, we extend these efforts and present a flexible model utilizing convolutional neural networks for discovering clusters of similar phrases in text that are predictive of human reactions to those texts. When used in an experimental setting, this method can identify candidate text treatments and effects under certain assumptions. We apply our model to two data sets. The first concerns censorship of social media posts and enables direct validation of our model. The second investigates complaints to the Consumer Financial Protection Bureau, and demonstrates the model's ability to flexibly discover text treatments with varying textual structures.

1 Introduction

Text impacts outcomes and decisions in many domains. Researchers have investigated the effects of campaign messaging on voting (Arceneaux and Nickerson, 2010), news story framing on public opinion (Druckman, 2001), post content on censorship (King et al., 2014), clinical notes on diagnoses and treatment (Sheikhalishahi et al., 2019), and written profiles on citizenship decisions (Hainmueller and Hangartner, 2013), to name a few examples. Most experimental methods for estimating the effects of text on human evaluation randomly assign some subjects to a treatment text that is edited in a particular way to be different from a control text. Researchers typically must confine experiments to a small number of text treatments to preserve power, reinforcing the importance of choosing effective treatments. These treatments are often chosen subjectively, which may be detrimental to the study if treatments are ineffective or lack external validity. Recent literature in computational social science has sought to randomly assign unique texts to respondents and then discover treatments from these unstructured texts that have an effect on an outcome of interest (Fong and Grimmer, 2016; Pryzant et al., 2018). Our approach builds on these efforts by utilizing pre-trained contextualized word embeddings to learn influential phrases of varying lengths, rather than being constrained to learning document-level sets of topics or to a set of particular words. Additionally, our model can accommodate the inclusion of covariates to account for other meta-data that may influence the outcome.

While this model is motivated by experiments that target causal effects of text, these effects can only be estimated under rather stringent assumptions. As a result, we suggest this model to be used to aid researchers in discovering relevant text treatments to test in confirmatory analyses, as an alternative to subjectively posing text treatments. To this end it builds on recent advances in self-explaining models (Alvarez Melis and Jaakkola, 2018) and interpretation of model structures (Lyu et al., 2023).

We demonstrate the ability of our model to identify influential aspects of text by applying it to two data sets. The first consists of social media posts on Weibo, where the outcome of interest is post censorship. Censorship of these posts can be tested against an API with access to a set of known blacklisted keywords, enabling clear validation of our model. In our second application, texts are complaints submitted to the Consumer Financial Protection Bureau (CFPB), and the outcome of interest is whether a complainant received a timely response. This application highlights the complexity of human decision making based on text, and the capacity of our model to learn predictive text features of various structures.

2 Related work

While much of the related social science work has focused on learning latent "features" of a text and using those as a treatment, most NLP work has focused on improving the interpretability of black-box predictive models. This paper bridges the gap between these two by using explainable ML methods to flexibly discover latent treatments in text and discover the effects of their inclusion.

Computational social science/causal inference Prior work has generated methods to both discover treatments and estimate their effects simultaneously (Fong and Grimmer, 2016; Pryzant et al., 2018; Egami et al., 2018; Fong and Grimmer, 2021; Feder et al., 2022). These models have typically focused on estimating either topics or individual words as treatments. Our model extends this work by allowing groups of similar phrases – instead of topics or unique words – to be identified as treatments. Fong and Grimmer (2016) apply a supervised Indian buffet process to both discover features (topics) and estimate their effect on an outcome in an RCT setting. Pryzant et al. (2018) approach a similar problem but use n-gram features instead of topics and use a neural architecture with a method for extracting feature importance from the weights of the network. While their primary focus is on adjusting for text confounders, we focus on capturing concepts which can be flexibly expressed across a variety of different length n-grams. Our approach will work particularly well in instances where the outcome may be caused by flexibly expressed, but relatively short concepts instead of particular words or the full topical content of the text.

Interpretable NLP In recent years many methods have been proposed to interpret and explain NLP models, as well as meta-evaluations of those methods (Lei et al., 2016; Alvarez Melis and Jaakkola, 2018; Rajagopal et al., 2021; Alangari et al., 2023; Crothers et al., 2023; Lyu et al., 2023). These methods almost all focus on explaining and interpreting predictions at the level of individual samples. In contrast, our method is designed to learn and interpret broader patterns that occur at the corpus level. In this respect, Rajagopal et al. (2021) is closest to our work in their pursuit of learning global influential concepts across texts, though our approaches differ. For our purposes, our interpretation methods are more intuitive in their relative simplicity, and our model learns "global concepts" adaptively as convolutional neural network filters, rather than requiring global concepts to exist as n-grams in the original training data.

Individual words and tokens are not human-interpretable or individually persuasive, so like Alvarez Melis and Jaakkola (2018) we force the network to have an interpretable final layer after a representation learning component. Their goal is for the representations used in the linear classification layer to satisfy the fidelity, diversity, and grounding conditions. Our goals are different however–rather than trying to understand why the network made the prediction it did, we seek representations of features which scientists can use in follow-up experiments.

3 Extracting influential text from latent representations

Our goal is to extract clusters of phrases that represent latent, generalizable treatments that affect a particular outcome. To do this, we imagine that N texts (T_i) are randomly assigned to a process through which they are mapped to an outcome (Y_i). Let i also index the individual evaluating text i. We seek to identify and estimate the effect of an m-dimensional latent representation of those texts (Z_i) which summarizes clusters of phrases or concepts that are likely to influence the outcome in repeated experiments. We refer to Z_i as "text treatments" for text i. For example, each element of Z_i could represent the presence or absence of a certain phrase or topic, with $Z_i \in \{0,1\}^m$. Z_i could also contain real-valued elements indicating continuous text features like similarity to a certain vocabulary or alignment with a concept.

To simulate a sequential experimental setup, we follow Egami et al. (2018) in splitting our sample into training and test sets. We first train our model, using cross-validation within the training set for tuning and model selection. We then use the test data set to interpret the latent text treatments discovered and estimate their effects on the outcome under additional assumptions. Our main contribution concerns this first stage: a model which identifies a mapping between text data and text treatments (Z_i) which predict the outcome of interest.

Fong and Grimmer (2016, 2021) outline the conditions under which this process identifies causal effects of the text treatments on the outcome when treatments are binary. They suppose that: 1) an individual's treatment depends only on their assigned text, 2) the latent features captured by the model are sufficient to predict an evaluator's response, 3) there is a nonzero probability of each evaluator receiving any of the possible text treatments (Z_i), given unmeasured text features[1], 4) texts are randomly assigned and 5) that latent treatments are not perfectly collinear. If these assumptions hold in our setting, we can also identify treatment effects of the discovered latent features. Following the methodology of Fong and Grimmer (2016), these may be estimated using linear regression under the additional assumption that the m text treatments do not interact with each other, in addition to linear modeling assumptions in the case of continuous treatment variables.[2] However, since it is difficult to assess whether these assumptions hold – particularly assumption 2 – we recommend that when possible, practitioners use our method to suggest potential treatments for study in a more controlled experimental setting.

[1]For real-valued treatment variables, this assumption should be modified to require that the probability density function of the treatment vector is nonzero

[2]Fong and Grimmer (2016) consider the Average Marginal Component Specific Effect, which captures the effect of changing one text treatment while averaging over values of all others. For continuous treatments, the process would identify a similar effect capturing the marginal effect of incrementally increasing a text treatment. If covariates are included in the neural network model, researchers may choose to include them in the regression model as controls as well.

4 Methodology

We propose a neural network architecture that utilizes convolutional structures to identify influential text (Figure 1). As the convolutional layers learn latent text representations, sample-level covariates may also be incorporated into the model to provide additional non-textual information.

4.1 Contextual encoder

We use pre-trained BERT models (Devlin et al., 2019) to tokenize our input text samples (T_i) and to obtain context-dependent embeddings of tokens by extracting the models' final hidden states. We denote these embeddings by $e_{i,j} \in \mathbb{R}^D$, where i indexes each text sample, j indexes tokens ($u_{i,j}$), and D represents the embedding dimension. With accessibility for social scientists in mind, we work with reduced-size models (Jiao et al., 2020), and do not perform fine-tuning. Researchers with fewer constraints on their computational budgets may find improved model performance from using larger models and/or fine-tuning these models on their outcome of interest. Any model providing text embeddings could be substituted for BERT. However, we do recommend using models that encode context between tokens. We perform the embedding step just before creating a train-test split, but researchers who choose to fine-tune their embedding models should reverse these steps to fine-tune and train only on the training set.

4.2 Model architecture

Once obtained from BERT, sequences of input text embeddings $\{e_{i,j}\}_j$ for each text are passed to a one dimensional convolutional layer C, or a series of M such layers in parallel (C_l), each with flexible kernel size K_l and F filters. The number of parallel convolutional layers is determined by the number of unique kernel sizes to be considered.

In the text representation learning problem, each filter learns some latent textual feature. These features could correspond to certain vocabulary usage, grammatical structure, or tone, for example. The number of filters F per layer can be adjusted, with more filters

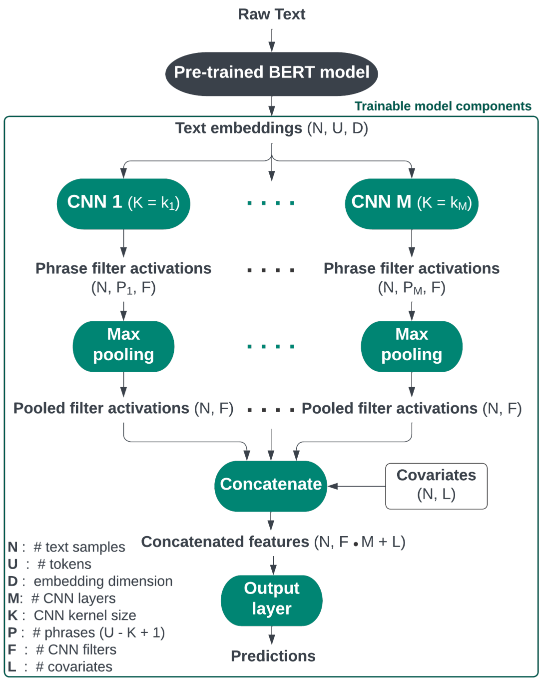


Figure 1: Model architecture

corresponding to learning more latent text features. In our implementation all convolutional layers learn the same number of filters. The kernel size K determines the size of the filter window, or the length of phrases considered by each convolutional layer. A filter f in a layer C with $K = 5$ tests the extent to which the representation learned by f is present in five-token phrases of the input text. For each phrase $p_1, \ldots, p_P$ with $P = U - K + 1$ and filter f, the convolutional operation produces a new feature $g(w \cdot p_i + b)$, where w and b are the learned weights and bias respectively for filter f, and g is the sigmoid activation function. We refer to these features as "filter activations". The filter activations $a_{i,f} \in \mathbb{R}^P$ are summarized per text sample by max pooling layers, which keep only the highest activation across a text's phrases per filter. The max-pooled activations $a_{i,f}^{pooled} \in \mathbb{R}$ are then concatenated across the parallel convolutional layers. If covariates are included in the model, those are concatenated as well. These activations and covariates $x_1, \ldots, x_L$ are passed to a final fully connected layer, where a weighted average of these values is pushed through an activation function (in our applications, sigmoid). These final activations correspond to the model predictions.

4.3 Training

The model is trained with respect to binary cross-entropy loss and Adam optimizer. Convolutional layer kernels and the final fully connected layer are subject to L2 and L1 regularization, respectively. Convolutional layers additionally receive custom activity regularization which penalizes the maximum correlation between two filter activations. This penalizes models that learn redundant filters (as measured by high correlation) to encourage convolutional layers to identify a larger number of distinct text features (Appendix A: Figure 2).

Hyper-parameters are determined according to a five-fold cross validation procedure using the training set. Because the motivation of these models is primarily interpretation of learned features, rather than prediction performance, model selection is more subjective than simply choosing the highest accuracy parameter settings. We selected models based on a combination of accuracy, degree of correlation between filter activations (i.e. feature redundancy), and the number of "useful"[3] filters learned. Parameter settings for the models selected in our applications are reported in the appendix.

The final model selected is then re-trained using the entire training set with a randomly sampled 20% serving as the validation set, and is assessed using the unseen test set.

4.4 Identifying and testing influential text features

The filters of the model's convolutional layers are trained to learn representations of text that are predictive of the outcome. To interpret the learned latent representations of the model and discover text treatments (Z_i) for each text, we utilize three model components:

1. The output filter activations of each text sample's phrases for each filter f ($a_{i,f}$);

2. The output layer weights, $w^{out} \in \mathbb{R}^{F \cdot M + L}$;

[3] Some models learn filter weights that produce near-identical activations across samples. As these filters do not meaningfully distinguish outcome predictions between texts, they are not useful for interpretation. We identify these filters by assessing the range of filter activations. We omit filters with ranges less than a threshold $t = 0.05$ wide.

3. The input text samples (T_i).

The filter activations represent how strongly each phrase corresponds with the text representation learned by each filter. The final output layer weights determine how each text representation contributes to the ultimate outcome prediction. Finally, the original input text samples provide context for the phrases that activate highly on each filter. This last component is most subjective to interpretation. Because input text embeddings are context-dependent via the pretrained BERT models, each phrases' embeddings contain more information than just the text tokens that make up the phrase, which lack the context of the rest of the sample. However, due to the difficulty of interpreting text embedding dimensions, the context that human readers assign to phrases when reading an entire sample may not align with context encoded by the embedding models.

To facilitate interpretation of the general concepts and patterns that each filter has learned and to assign manual labels to each filter, we pull phrases from the test sample which have the highest filter activations for each filter and refer to the corresponding full text samples for context. We verify how these concepts are related to the outcome by the corresponding final output layer weights, and by the relationship between the filter activation values and the true outcome values.

The objective of this interpretation process depends on whether the researcher wishes to directly estimate the effects of the identified latent features in the test set under assumptions described in Section 3, or if they wish to discover concrete text features to test in a follow up experiment. In the first scenario, the max-pooled filter activations ($a^{pooled}_{i,f}$) may be considered directly as the sample-level latent text treatments (Z_i), with the total number of filters across convolutional layers corresponding to the dimension m of the treatment vector. Researchers could also choose to binarize these features, for example by defining $Z_{i,f} = \mathbf{1}[a^{pooled}_{i,f} > \bar{a}^{pooled}_{f}]$ where $\bar{a}^{pooled}_{f}$ is the median of ($a^{pooled}_{i,f}$). This avoids the more stringent modeling assumptions needed for estimating effects of continuous treatments, though it may complicate interpretation. In either case, this process provides the researcher an understanding for what the latent text treatments represent and therefore the effects that they are estimating. In the second scenario, we see model interpretation as a more general tool to guide the researcher's process for obtaining concrete text treatments. Here, a second set of text treatments, $\tilde{Z}_i$, are established which are not latent in the same sense as Z_i, because researchers control the definition of these treatments. For example, researchers could define $\tilde{Z}_i$ as the inclusion or absence of the manual labels assigned to each filter as keywords in experimental texts, or as indicators of different tones or concepts identified by filters.

4.5 Evaluation methods

In our application of this model to predicting social media post censorship, we have ground-truth explanations of which phrases led to censorship. We show that our trained model and interpretation methods recover the most commonly labeled reasons for censorship. With our application to the CFPB data set, we compare our findings to those in Egami et al. (2018), who both discover latent text treatments via topic modeling and test their effects.

5 Experiments

5.1 Weibo post censorship

Dataset and setup For our first application, we use a sample of 28,386 Weibo posts from the Weibo-Cov dataset (Hu et al., 2020). These are social media posts on the topic of COVID and were posted in February 2020 on Weibo.[4] To obtain the censorship label for each post, we use the content review API from Baidu. The API is a classifier that returns the probability of censorship for each post. The API only returns a probability of 1 when a social media post includes words or phrases that are on Baidu's blacklist. As the API also returns the flagged keywords and phrases, this enables us to validate whether our model can recover keywords and phrases that led to censorship.

We train our model to predict whether or not a post was flagged by the API to be cen-

[4]The creators of this data set anonymized identifiable information in posts to protect the privacy of individual users.

F	w^{out}	β	Top extracted phrases (translated)	Known censored phrase
1	1.4	0.22	"[CLS]Wuhan Institute of Virology Party","Wuhan Institute of Virology Specialty","[CLS]Wuhan Institute of Virology'","? Created by the Wuhan virus"	"Wuhan virus"
2	1.3	0.24	"Profiting from national disasters, such people", "Chinese virus said that some people", "Profiting from national disasters, such as some people", "Profiting from national disasters, some people dare to make money,"	"Profiting from national disasters"
3	1.2	0.25	"Secretary of the Provincial Party Committee of a province", "Chen Quanjiao of the Poison Institute stated", "Renowned Secretary of the Hubei Provincial Party Committee", "Remdesi of the Poison Institute."	"Provincial party secretary"
9	0.91	0.07	"Diagnosis and Shincheonji Teaching", "Always waiting for Shincheonji Teaching", "No guarantee of payment time"	"Shincheonji Church"
10	0.77	0.11	"Jiang Chaoliang is in Wuhan"	"Jiang Chaoliang"

Table 1: Frequent censorship rationale is learned by the model. The first column distinguishes filters in order of the second column, the weight assigned to max-pooled filter activations $a_{i,f}^{pooled}$ in the final model layer. The third column shows the coefficients from regressing the labels on $a_{i,f}^{pooled}$. The fourth column lists filters' unique top 4 most associated phrases from the test set. The fifth column associates each filter with a commonly reported censored phrase.

sored with probability 1. Although this outcome is not determined by direct human decision making, it reflects a more general policy of censorship, and allows us to validate our model with the outcome explanations. We can view the keyword and phrase blacklist as a decision maker that is perfectly consistent with these human-defined preferences. To tokenize and embed these texts, we use a pre-trained BERT Chinese language model provided by the Joint Laboratory of HIT and iFLYTEK Research, `MiniRBT-h288` (Yao et al., 2023).[5] This model has an embedding dimension of 288 and 12.3M parameters. The embeddings from the BERT model's last hidden state are used as the input features to our model architecture (see Figure 1). Examples of posts in this data set, their censor probabilities, and their censor words (when applicable) with English translations are shown in Appendix A Table 3. Appendix A Table 4 shows the top 10 censor words across all censor-probability-one samples, their translations, and the proportion of censored samples corresponding to each.

Results The trained model obtains an accuracy score of 0.87 on the test set. This performance indicates that the model has learned useful representations of Weibo posts from this time period which are predictive of censorship.

[5]Model is licensed under Apache License 2.0.

We highlight our interpretation of the most relevant representations in Table 1, with interpretation of all representations included in Appendix A Table 7. We find that the two most commonly censored phrases, "Wuhan virus" (23.9% of censored posts) and "national crisis" (4.9% of censored posts) are clearly identified by the model in the first and second model filters – the phrases which activate most highly on these filters contain almost exactly these phrases. The max-pooled activations for these filters also contribute the most to the model's final prediction of censorship, as seen in the w^{out} column of this table. The most highly-activating phrases for filters 3 and 9 share in common two other known censored phrases, "Provincial party secretary" and "Shincheonji Church," and the highest activated phrases for filter 10 concentrate exactly around the same phrase, which relates to a fifth known censor phrase "Jiang Chaoliang." The complete set of representation interpretations in Appendix A demonstrates that there is some amount of redundancy in the keywords learned by filters. Their differences in sentence structure and context could be illuminating in other settings, though in this case we know that it is solely the inclusion of these phrases which affects the outcome. As a proof-of-concept, we include the effect estimates we obtain by regressing the labels on the max-pooled filter activations of the test sample texts, though assumptions for identification of a causal effect are not met (for one, texts are not randomized amongst evaluators). Though the magnitude of the estimated effects differ from the output layer weights (in large part because the output layer weights correspond to a sigmoid rather than linear activation), they are in relative agreement about which text treatments are found to be most influential for censorship.

Model validation We find that this model and our interpretation methodology successfully recovers the phrases which cause the most posts to be censored. In a setting without oracle knowledge of the censored phrases, we feel confident that researchers would be able to use this model to determine at least five of the most common censored phrases with only access to the posts and the final outcome variable. We ran additional evaluations which demonstrated that almost all of the top 10 phrases were learned to be influential by the model, even if some were less easily identifiable in our interpretation process. In these evaluations, the trained model received two constructed data sets containing placebo text data and text data containing one of the top censored phrases. In one data set, the placebo texts are fully randomly sampled sequences of characters while the test texts also include an embedded censored keyword. In the other, the placebo texts are movie reviews from an unrelated data source, and test texts are fake Weibo posts containing censored phrases generated by ChatGPT. In both evaluations, texts with embedded censored phrases obtain much higher median filter activation values compared to the placebo texts for all but two of the top 10 censor phrases (Appendix A Figures 3 and 4).

5.2 Consumer Financial Protection Bureau complaint response

Dataset and setup For our second application, we use a dataset from Egami et al. (2018) of 54,816 consumer complaint narratives submitted to the Consumer Financial Protection Bureau[6] from March of 2015 to February of 2016. The outcome variable indicates whether or not the complainant received a timely response from the company filed against. Due to strong imbalance in the outcome variable, we proceed with a subsample of complaints which received a timely response (5136 timely and 1712 non-timely responses) combined with a class-weighted loss function, which we found to perform best during training in terms of the F1 score. We also utilize product type information included in the dataset as an additional set of covariates, which describes which financial product the complaint concerns (ex. mortgage, debt collection). To tokenize and embed the complaint texts, we use a pretrained BERT English language model trained by Google Research (Turc et al., 2019; Bhargava et al., 2021). To obtain word embeddings, we use the last hidden states from `bert-tiny`[7], which has an embedding dimension of 128 and 4M parameters.

Results The trained model obtains an accuracy score of 0.73 and an F1 Score of 0.59 on the test set. Given the limited size of the data set used, the class imbalance, and the relative complexity of this learning task, it is not completely surprising that this model achieves a lower performance compared to the previous application. However, we believe that the representations learned by the model could provide meaningful insights to inform a hypothetical researcher seeking to uncover text treatments.

Table 2 summarizes interpretation of the representations learned by this model. Three filters which receive a weight of < 0.003 on the output layer are omitted, as these have little influence on the model's predictions. In this application, we do not have access to the true reasons that complaints receive or do not receive timely responses, and can imagine that a variety of text features could impact this outcome. We infer that formal or polite language and references to past attempts for resolution may be positively associated with timely responses, and that rehashing conflicts over claims, referencing disputed debt collection, and discussing frustrating past communications may be negatively associated with timely responses. We also include effect estimates from regressing

[6] Data is publicly available for download at: https://www.consumerfinance.gov/data-research/consumer-complaints/. The CFPB removes personal information from complaints.

[7] Model is licensed under Apache License 2.0.

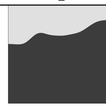
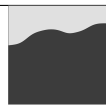
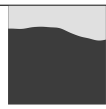
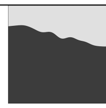
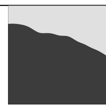
F	w^{out}	β	Top extracted phrases	Inferred Concept	CD plot
1	1.6	0.08	"rei mb urse d immediately", "additionally , ex per ian", "late fee charged . please", "contacts lacking mandatory legal documentation", "xx , 2015 . please"	Formal language, pleading	
2	1.4	0.04	"deposit that more than covered", "connection is dropped and clear", "been over 30 days since", "entered every wednesday and there", "tried on more than xx"	Past attempts for resolution	
3	-0.92	-0.03	"that i wrote a check", ". he claims the address", "no matter what i say", ". she claimed a reference", "no longer need a prep"	Conflicting/false claims	
4	-1.2	-0.05	"this was a fraudulent debt collector ,", "i received a statement indicating a ", "i was the victim of identity theft", "this battle over a debt that is", "i owe mon ies for alleged damages"	Disputed debt collection	
5	-1.3	-0.13	"voice mail messages stating they have attempted", "was trying to convince my father was", "of someone who could ' ve been", "then started asking why i was been", "by someone who did not want to"	Frustrating communications	

Table 2: CFPB model interpretation. Columns 1, 2, and 4 correspond to those in Table 1. The third column shows the coefficient from regressing the label on $a_{i,f}^{pooled}$ and product type. The fifth column contains a manual interpretation of the top extracted phrases. The sixth column displays conditional density plots for the max-pooled filter activations. The x-axis of these plots represents the activation value. The y-axis indicates estimated probability of belonging to the positive class (dark gray).

the test set labels against the texts' corresponding max-pooled filter activations and the product type covariate as a control. Again, we believe it is unlikely that the assumptions necessary for causal interpretation of these effects are met. However, the estimates could still act a useful tool for a researcher exploring possible text treatments to test in a follow-up experiment. They align with the final output layer weights and imply that the inclusion of formal language/pleading or of references to frustrating communications may be text treatments worth investigating further.

Model evaluation The full CFPB data set was also analyzed in Egami et al. (2018), who use a topic modeling approach to identify and test text treatments. The majority of their identified text treatments align with product types. Their results imply that the inclusion of identified "loan" and "detailed complaint" topics each have the strongest positive effects on timely response, and that the inclusion of identified "debt collection" or "threat" (seemingly related to debt collection or credit reporting) topics each have a negative effect. These results supports our finding of a negative association between the discussion of disputed debt collection and timely response, though the other features that our model identifies deviate from the topics in Egami et al. (2018). Another analysis of CFPB data, Pryzant et al. (2021), finds that perceived politeness in complaints may have a positive effect on reducing response time, which supports our finding for the first filter shown in Table 2. The capacity of our model to detect both of these kinds of text features - topics and tone - in clusters of phrases highlights its flexibility at picking up a variety of different text qualities.

6 Conclusion

We present a new method to discover influential text features represented by clusters of phrases of flexible length. Our approach is inspired by and builds upon previous work in computational social science and interpretable NLP, and provides experimenters with a quantitative tool for identifying promising text treatments to test in follow up experiments. When researchers are willing to make stronger identification assumptions discussed in Section 3, text treatments identified by using the model can also be used to estimate causal effects on the test test directly. Our applications demonstrate the ability of our model to learn useful latent text representations and its capacity to recover known influential text features.

Limitations

Small BERT models used out-of-the-box In this paper, we do not investigate how model performance could be affected by fine-tuning the pre-trained BERT models, or by using larger models to obtain higher dimensional word embeddings. Future work investigating how benefits from these changes trade-off with reduced computational efficiency would be relevant to researchers using this method.

Testing the inclusion of covariates Although including covariates in the final layer allows us to account for them in model predictions, the extent to which they allow us to "control" for document-level features when learning latent text treatments is unclear. An analysis of our model's performance on a set of texts with known meta-data confounding and for which effects can be validated would be useful.

Trade-off between experimental costs and less-interpretable treatments Under the assumptions discussed in Section 3, researchers may estimate causal effects by directly testing the identified latent text treatments. This simplifies the experimental pipeline, but as in Egami et al. (2018) and Fong and Grimmer (2016), comes with the drawback of requiring the researcher to somewhat subjectively interpret the identified latent text treatments that are being tested. Alternatively, researchers may use their interpretations of the discovered latent text features to inspire "manifest" text treatments (ex. specific keywords, sentence structures) to test in confirmatory settings. In this case, the text features being tested would be known and manipulated by the researcher, allowing clearer interpretation of effects and weaker assumptions. The downside here would be the requirement of researchers to run follow-up experiments.

Incorporating uncertainty in latent treatments Our paper does not provide guidance for incorporating the uncertainty involved in identifying and estimating the latent text treatments into causal effect estimates.

Designing experimental texts We generally recommend using our model to guide the selection of text treatments for use in follow-up experiments. Designing experimental texts to isolate treatments of interest is a non-trivial task, and is left to the experimenter. In many cases, it is challenging to imagine altering a specific part of a text without affecting surrounding text that is not directly manipulated. This makes it difficult to establish causality for a specific text feature, rather than for the aggregate differences between a set of texts. This is a known challenge of making causal inferences with text, and relates to the strong ignorability assumption discussed in Section 3.

Ethics Statement

For any model designed to extract persuasive concepts, there is a risk that bad actors could use it to improve their ability to manipulate others. Many other tools exist which could presumably be used for this purpose, so we believe that the benefits of having this model open source outweigh this risk. An example of this kind of trade-off can be seen in the context of the model' s application to censorship. When governments utilize human censors, they could potentially use this model to identify new keywords to add to an automated censorship blacklist to improve efficiency. On the other hand, the model can also be used to reverse engineer the process and reveal censorship policies, as we demonstrate. Acknowledging the possibility for misuse, we believe that the opportunities for productive and socially beneficial application are greater.

References

Nourah Alangari, Mohamed El Bachir Menai, Hassan Mathkour, and Ibrahim Almosallam. 2023. Exploring Evaluation Methods for Interpretable Machine Learning: A Survey. *Information*, 14(8):469.

David Alvarez Melis and Tommi Jaakkola. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Kevin Arceneaux and David W. Nickerson. 2010. Comparing Negative and Positive Campaign Messages: Evidence From Two Field Experiments. *American Politics Research*, 38(1):54–83.

Prajjwal Bhargava, Aleksandr Drozd, and Anna

Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics.

Evan Crothers, Herna Viktor, and Nathalie Japkowicz. 2023. Faithful to Whom? Questioning Interpretability Measures in NLP.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

James N. Druckman. 2001. On the Limits of Framing Effects: Who Can Frame? *The Journal of Politics*, 63(4):1041–1066.

Naoki Egami, Christian J. Fong, Justin Grimmer, Margaret E. Roberts, and Brandon M. Stewart. 2018. How to Make Causal Inferences Using Texts.

Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. 2022. Causal Inference in Natural Language Processing: Estimation, Prediction, Interpretation and Beyond.

Christian Fong and Justin Grimmer. 2016. Discovery of Treatments from Text Corpora. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1600–1609, Berlin, Germany. Association for Computational Linguistics.

Christian Fong and Justin Grimmer. 2021. Causal Inference with Latent Treatments. *American Journal of Political Science*, 64(2).

Jens Hainmueller and Dominik Hangartner. 2013. Who Gets a Swiss Passport? A Natural Experiment in Immigrant Discrimination. *American Political Science Review*, 107(1):159–187.

Yong Hu, Heyan Huang, Anfan Chen, and Xian-Ling Mao. 2020. Weibo-COV: A Large-Scale COVID-19 Social Media Dataset from Weibo.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for Natural Language Understanding.

Gary King, Jennifer Pan, and Margaret E. Roberts. 2014. Reverse-engineering censorship in China: Randomized experimentation and participant observation. *Science*, 345(6199):1251722.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2023. Towards Faithful Model Explanation in NLP: A Survey.

Reid Pryzant, Sugato Basu, and Kazoo Sone. 2018. Interpretable Neural Architectures for Attributing an Ad's Performance to its Writing Style. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 125–135, Brussels, Belgium. Association for Computational Linguistics.

Reid Pryzant, Dallas Card, Dan Jurafsky, Victor Veitch, and Dhanya Sridhar. 2021. Causal effects of linguistic properties. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4095–4109, Online. Association for Computational Linguistics.

Dheeraj Rajagopal, Vidhisha Balachandran, Eduard H Hovy, and Yulia Tsvetkov. 2021. SELFEXPLAIN: A self-explaining architecture for neural text classifiers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 836–850, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Seyedmostafa Sheikhalishahi, Riccardo Miotto, Joel T Dudley, Alberto Lavelli, Fabio Rinaldi, and Venet Osmani. 2019. Natural Language Processing of Clinical Notes on Chronic Diseases: Systematic Review. *JMIR Medical Informatics*, 7(2):e12239.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*.

Xin Yao, Ziqing Yang, Yiming Cui, and Shijin Wang. 2023. Minirbt: A two-stage distilled small chinese pre-trained model.

A Appendix

Demonstration of increasing the filter activation correlation penalty

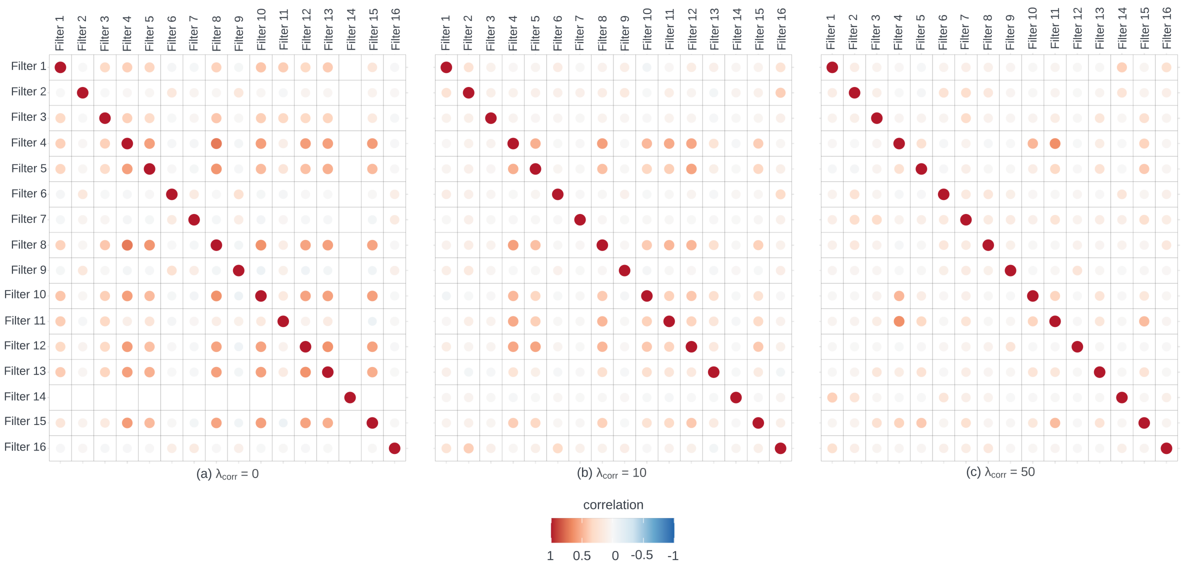


Figure 2: Correlation grids for filter activations when the correlation penalty is increased from (a) 0 to (b) 10 to (c) 50 for the censorship model. Darker red indicates a pairwise correlation that is closer to 1, darker blue indicates a pairwise correlation that is closer to -1, and white indicates a pairwise correlation close to 0.

Example posts from the Weibo censorship data set

Weibo post [translation]	Censorship probability	Censor keywords
武汉病毒所致信全所职工和研究生一首小诗，童年是一道彩虹，童年是一缕阳光。我把色我的童年印在一张张照片上，陪伴着我快乐地成长。[A letter from the Wuhan Institute of Virology to all employees and graduate students of the Institute: A little poem, childhood is a rainbow, childhood is a ray of sunshine. I printed my childhood on a photo and grew up happily with me.]	1.0	武汉病毒 [Wuhan virus]
疫情当前大发国难财，所售口罩均为三无产品怒怒怒说发货没有快递单号，退款均需扣费，请尽快查处怒怒怒 (tagged usernames omitted) [In the current epidemic situation, there is profiteering at the expense of the nation. All the masks sold are substandard products. Anger, anger, anger! It is claimed that shipments are made without providing a tracking number, and refunds will be subject to charges. Please investigate and resolve this issue as soon as possible. (tagged usernames omitted)]	1.0	国难财 [Profiting from national disasters]
点赞遵义遵义：一手抓防控一手抓经济，遵义复工复产全面铺开一手抓防控一手抓经济，遵义复工复产全面铺开转发理由: 转发微博 [Thumbs up for Zunyi. Zunyi: One hand focuses on epidemic prevention and control, and the other hand promotes economic development. Zunyi has comprehensively resumed work and production. Thumbs up for Zunyi. One hand focuses on epidemic prevention and control, and the other hand promotes economic development. Zunyi has comprehensively resumed work and production. Reason for reposting: Reposting Weibo.]	0.5	
韩红捐赠的救援车进入雷神山韩红爱心慈善基金会捐赠的救护车进入雷神山了，整整齐齐的一排，谢谢韩红老师以及捐款的人嗷!! 转发理由: good good good [The rescue vehicle donated by Han Hong entered Leishen Mountain. The ambulances donated by Han Hong Charity Foundation entered Leishen Mountain. They were lined up neatly. Thank you, Teacher Han Hong and those who donated! ! Reason for forwarding: good good good]	0.0	

Table 3: Sample posts from the Weibo post censorship data set. The first column contains sample posts and their translations into English. The second column is the probability of censorship, and the third column contains associated censorship keywords (when applicable) as returned by the Baidu API.

Most common censor keywords

Censor keywords	Translation	%
武汉病毒	Wuhan virus	23.9
国难财	Profiting from national disasters	4.9
抗肺炎	Anti-pneumonia	3.7
副省长	Deputy Governor	3.6
安倍晋三	Shinzo Abe	3.5
蒋超良-省委书记	Jiang Chaoliang-Secretary of the Provincial Party Committee	2.7
不作为 & 当地政府	Inaction & local government	2.4
省委书记	Provincial party secretary	2.3
省长	Governor	1.9
新天地教会	Shincheonji Church	1.9

Table 4: The 10 most common censor keywords in the Weibo post censorship data set. The first two columns contain words and phrases on Baidu's blacklist of censor keywords and their translations. The third column contains the percentage of justifications corresponding to each censor word/phrase.

Hyper-parameter	Value
Number of tokens per sample	150
Number of filters per convolutional layer	8
Kernel sizes of conv. layers	5, 7
Conv. layer kernel regularizer penalty	0.001
Conv. layer activity regularizer penalty	3
Output layer kernel regularizer penalty	0.0001
Learning rate	0.0001

Table 5: Hyper-parameter settings for the censorship model used to produce our reported results. This model has 27 681 trainable parameters total. During parameter tuning and the final model training, all models were trained for 100 epochs with early stopping (patience = 15) and batch sizes of 32.

Tuned hyper-parameter	Values considered in tuning
Number of filters per conv. layer*	4, 8, 16
Kernel sizes of conv. layers	5, 7, 5 and 7
Conv. layer kernel regularizer penalty	0, 0.0001, 0.001
Conv. layer activity regularizer penalty	0, 1, 3
Output layer kernel regularizer penalty	0.0001, 0.001, 0.01
Learning rate	0.00001, 0.0001, 0.001

Table 6: The censorship model parameter tuning process searched models with combinations of the above hyper-parameter values. Each model utilized 9.3 minutes of CPU time on average during training. The tuning procedure considered 486 different parameter settings, and with 5-fold cross validation for each setting utilized a total of 375 CPU hours across 4 cores. Each core was allocated 50GB of memory. Tuning was performed on a shared-resource computing cluster associated with our institution. *Models were required to have 8 or 16 total filters across convolutional layers. Combinations with one convolutional layer with 4 features, and models with two convolutional layers with 16 features each, were omitted from the tuning procedure.

Interpretation of all learned filters by the censorship model

F	w^{out}	β	**Top extracted phrases (translated)**	**Known censored phrases**
1	1.4	0.22	“[CLS] 武汉病毒所党”,“验武汉病毒所专”,“[CLS] 武汉病毒所’”,“？ 武汉病毒所辟”,“。武汉病毒所所” [“[CLS]Wuhan Institute of Virology Party”,“Wuhan Institute of Virology Specialty”,“[CLS]Wuhan Institute of Virology’”,“? Created by the Wuhan virus”, “. Wuhan Institute of Virology”]	“Wuhan virus”
2	1.3	0.24	“国难财”，如此人”,“汉病毒所说某中”,“国难财比如某些”,“国难财也敢发，”,“国难财，有些人” [“Profiting from national disasters, such people”, “Chinese virus said that some people”, “Profiting from national disasters, such as some people”, “Profiting from national disasters, some people dare to make money,”,“Profiting from national disasters, some people”]	“Profiting from national disasters”
3	1.2	0.25	“个省的省委书记”,“毒所陈全姣声明”,“任湖北省委书记”,“毒所的 remdesi” [“Secretary of the Provincial Party Committee of a province”, “ Chen Quanjiao of the Poison Institute stated”, “ Renowned Secretary of the Hubei Provincial Party Committee”, “ Remdesi of the Poison Institute.”]	“Provincial party secretary”
4	1.2	0.12	“病毒所党委”,“病毒所所长”,“病毒所研究”,“病毒所联合” [“Party Committee of the Institute of Virology”, “Director of the Institute of Virology”, “Research of the Institute of Virology”, “Union of the Institute of Virology”]	“Wuhan virus” (using context of phrases within samples)
5	1.2	0.06	“病毒所回应 6 大”,“病毒所所长已经”,“病毒所所长”（正” [“The top 6 responses from the Institute of Virology”, “Director of the Institute of Virology has been”, ”Director of the Institute of Virology” (positive)]	“Wuhan virus”
6	1.1	0.11	“那些发国难”,“上是发国难”,“授旗。省委”,“期间发国难”,“任湖北省委” [“Those who caused national calamity”, “The one who caused national calamity”, “granted the flag. Provincial Party Committee”, ”During the national crisis”, ”Served as Hubei Provincial Party Committee”]	“National crisis”
7	1.1	0.11	“武汉病毒所” [“Wuhan Institute of Virology”]	“Wuhan virus”
8	1.0	0.15	“发国难财！”,“发国难财” [“Profiting from national disasters! ”,“Profiting from national disasters”]	
9	0.91	0.07	“确诊与新天地教”,“一直等新天地教”,“不保证打款时间” [”Diagnosis and Shincheonji Teaching”, ”Always waiting for Shincheonji Teaching”, ”No guarantee of payment time”]	“Shincheonji Church”
10	0.77	0.11	“蒋超良在武” [”Jiang Chaoliang is in Wuhan”]	“Jiang Chaoliang”
11*	-0.38	-	“2020 我们需要的是”,“: 辛苦啦，希望”,“！ 辛苦了！ 抱抱”,“，东西都来不及”,“？ 有坚持有希望” [“What we need in 2020 is”,“:Thank you for your hard work, hope”,“! Thanks for your hard work! Hug”,“, it’s too late for anything”,“? ”Persistence and hope”	
12*	-0.48	-	“购买防护及消毒”,“武汉加油！ 转发”,“铁、公交等公共”,“距离接触等条件”,“交往增多，临省” [“Purchase protection and disinfection”, “Come on Wuhan! Forward”, “Railway, bus and other public places”, “Distance contact and other conditions”, “Increased exchanges, close to the province”]	
13*	-0.66	-	“战疫，我们”,“疫情，我们” [”Fight the epidemic, we”, ”Fight the epidemic, we”]	
14*	-0.80	-	“上报的防疫”,“召开的疫情”,“条件的传染”,“其来的疫情” [“Reported epidemic prevention”, “Convened epidemic”, “Conditional infection”, “Occurring epidemic”]	

15	-1.1	-0.09	“国加油！ 心”,“国加油！ 加”,“子里凉凉了”,“[CLS] 春暖花开”,“待春暖花开” [“Come on country! Heart”, ”Come on country! Add”, ”It’s getting cold inside”, ”[CLS] The flowers are blooming in the spring”, ”Waiting for the flowers to bloom in the spring”]	
16	-1.2	-0.04	“leban 乐班营业”,“今天是疫情开工”,“机器。泪泪家里”,“今天, 20200202, ”,“过去, 老伙伴们” [“leban Leban is open for business”, “Today is the start of the epidemic”, “Machine. Tears at home”,“Today, 20200202,”,“In the past, old friends”]	

Table 7: Full results of censorship model filter interpretation. The first column distinguishes filters in order of the second column, the weight assigned to max-pooled filter activations $a_{i,f}^{pooled}$ in the final model layer. The third column shows the coefficient from regressing the label on $a_{i,f}^{pooled}$. The fourth column lists the unique phrases within the top 5 test set phrases that were most associated with each filter. The fifth column associates filters with one of the top 10 most commonly reported censor words in the data set (blank if none are applicable). *The associated max pooled filter activations had a range of less than 0.05, and therefore were omitted from interpretation and the regression to estimate β.

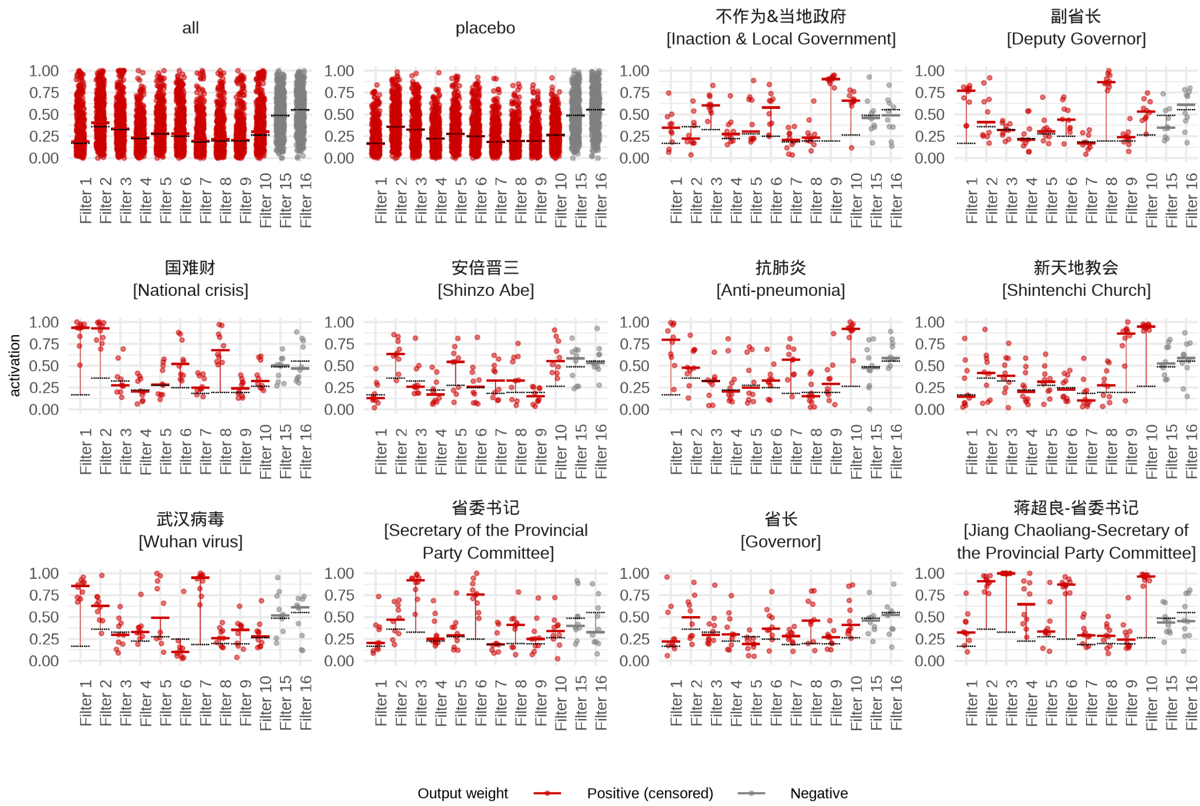


Figure 3: Validation test that the censorship model learns latent features strongly aligned with censor words. This simulated test data set contains 500 texts which are constructed by randomly sampling characters according to the probability distribution of characters in the full censorship data set. 10 of the most frequent censor words in the data set are inserted into 100 of these samples. Filter activation plots are shown for the samples corresponding to each censor word tested, as well as for the "placebo" fully random samples and all samples in aggregate for comparison (scatter points). We compare the median activation of the censor word samples (solid lines) to the median activation of the placebo samples (dotted lines) on each filter with activation range above 0.05. Vertical lines connect these median values, with longer lines indicating a larger difference between values. Filters with a positive output layer weight (predicted as more associated with censorship) are shown in red, with negative output layer weight filters in gray.

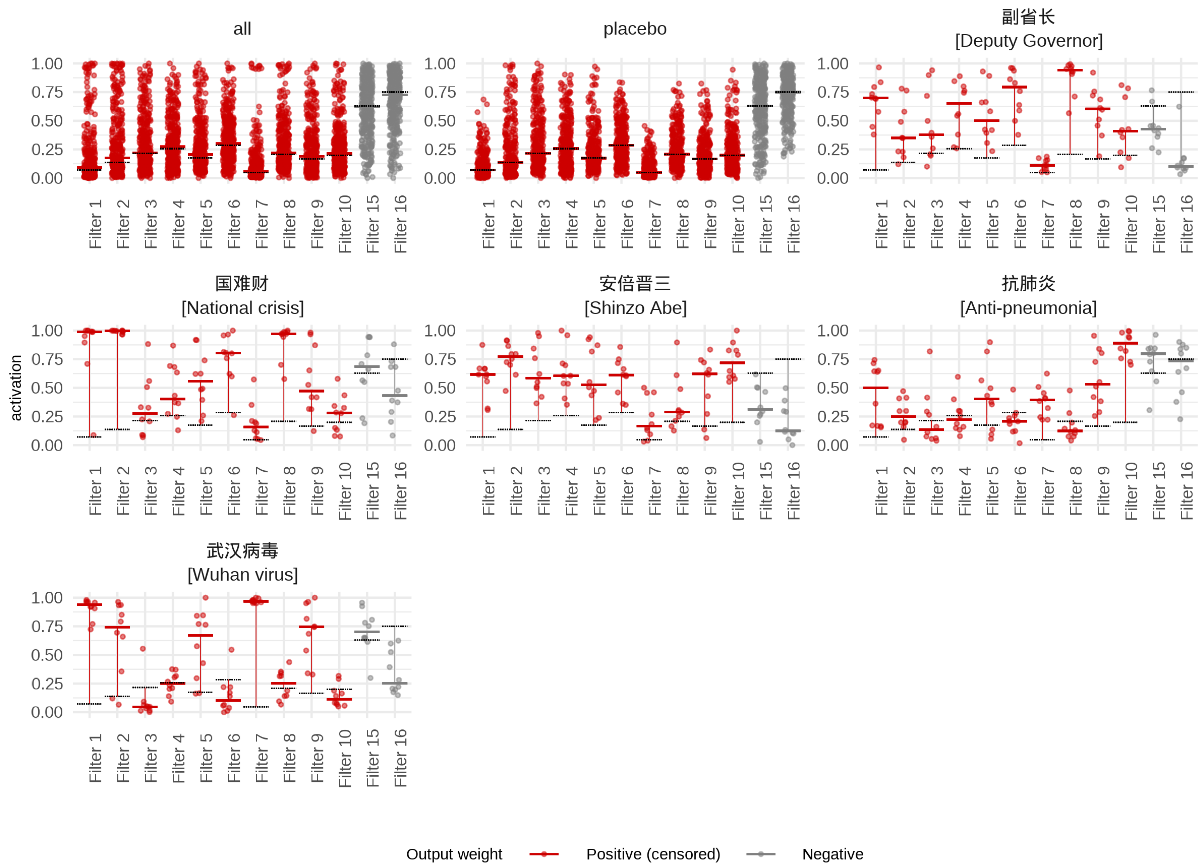


Figure 4: Validation test that the censorship model learns latent features strongly aligned with censor words. This data set combines 300 randomly sampled texts from the Kaggle Doubon Movie Short Comment data set, which is unrelated to the censorship data set, and 50 test samples each containing one of the 5 most frequent censor words generated. These test samples were generated using ChatGPT 3.5, which was prompted to create fake Weibo posts using the censor words. Filter activation plots are shown for the samples corresponding to each censor word tested, as well as for the "placebo" fully random samples and all samples in aggregate for comparison (scatter points). We compare the median activation of the censor word samples (solid lines) to the median activation of the placebo samples (dotted lines) on each filter with activation range above 0.05. Vertical lines connect these median values, with longer lines indicating a larger difference between values. Filters with a positive output layer weight (predicted as more associated with censorship) are shown in red, with negative output layer weight filters in gray.

Hyper-parameter	Value
Number of tokens per sample	250
Number of filters per convolutional layer	4
Kernel sizes of conv. layers	5, 7
Conv. layer kernel regularizer penalty	0.0001
Conv. layer activity regularizer penalty	0
Output layer kernel regularizer penalty	0.01
Learning rate	0.001

Table 8: Hyper-parameter settings for the CFPB model used to produce our reported results. This model has 6172 trainable parameters total. During tuning and the final model training, all models were trained for 100 epochs with early stopping (patience = 15) and batch sizes of 32.

Tuned hyper-parameter	Values considered in tuning
Number of filters per convolutional layer*	4, 8, 16
Kernel sizes of conv. layers	5, 7, 5 and 7
Conv. layer kernel regularizer penalty	0, 0.0001, 0.001, 0.01
Conv. layer activity regularizer penalty	0, 1, 3, 5
Output layer kernel regularizer penalty	0.0001, 0.001, 0.01

Table 9: The CFPB model parameter tuning process searched models with combinations of the above hyper-parameter values. Records of computational resources used for this parameter tuning process are no longer available to us. Based on those used to train the final model (7.2 minutes of CPU time), we estimate that the tuning procedure, which considered 384 different parameter settings with 5-fold cross validation for each, would have utilized about 230 CPU hours across 3 cores each with 40GB of memory. Tuning was performed on a shared-resource computing cluster associated with our institution. *Models were required to have 4, 8 or 16 total filters across convolutional layers. Combinations producing a model with two convolutional layers with 16 features each were omitted from the tuning procedure.