Leveraging State-of-the-Art Unsupervised Style Transfer Models for Counterfactual Text Generation

Sriram Ravula
UT Austin ECE
Sriram.Ravula@utexas.edu

Diego Garcia-Olano
UT Austin ECE
diegoolano@gmail.com

Abstract

We propose to extend current state-of-the-art unsupervised style transfer models to generate counterfactuals on real-world text corpuses. We explain the need for counterfactuals as easily-understandable examples that illustrate the reasoning behind text classification models' decisions. We expand on the similarities and differences between style transfer, adversarial attacks, and counterfactuals to elucidate our objective and note that style transfer models are more suited for our goal than adversarial text models. We review current literature on methods for style transfer and evaluate two models on the quality of their generated counterfactuals over multiple datasets.

1 Introduction

In canonical text classification and natural language inference (NLI) tasks, it is often informative to understand the reasoning behind a certain classification decision. Counterfactual explanations attempt to provide insight into a classification decision by perturbing an input example in some minimal way that leads to a change in a classifier's output for that example (Sharma et al., 2019). For natural language processing (NLP) tasks, counterfactuals attempt to maintain fluency and coherence in a text sample while making a small, semantically meaningful change. We treat the unsupervised setting, in which we do not have access to a parallel corpus of samples which have different classification labels for minimally different pairs of texts.

Style transfer and adversarial attacks are two problems which are similar to conterfactual generation and which have been studied extensively for NLP tasks. Style transfer attempts to modify the style of a text sample (i.e. its classification) while maintaining its non-style content. On the other hand, adversarial attacks attempt to ”fool” a classifier by making some minimal change to a sample which would be almost imperceptible to a human, but which would cause a classifier to change its decision. Augmenting a dataset with adversarial examples can lead to more robust training of models for downstream discriminative tasks (Ilyas et al., 2019). Although these two problems are very similar to counterfactual generation, the goals of style transfer are more relevant to the counterfactual task since style transfer (1) seeks to maintain fluency in samples and (2) is concerned more with human perception of generated samples than exploiting a weakness in a trained classifier.

In this work, we seek to extend state-of-the-art models trained for unsupervised textual style transfer to the task of generating useful counterfactuals for explaining classifier decisions. Style transfer models may alter the style (i.e. the classification-influencing attributes) of input text samples to a large degree as long as the resulting output has the correct classification. Therefore our main challenge is to minimize the change in not only content, but style of transformed samples as well.

We first describe in more formal detail the differences between style transfer, adversarial attacks, and counterfactuals to clarify why style transfer is a more suitable starting point for counterfactual text generation than adversarial attacks. Then we discuss recent work in the area of unsupervised style transfer and evaluate two state-of-the-art style transfer methods for counterfactual generation on multiple datasets.

2 Background

To illustrate the differences between an adversarial attack, style transfer, and counterfactual, we formulate the task of finding an example of each of the three as an optimization problem. First, we define some mathematical tools which can help us determine a concrete optimization objective and constraints. These tools need not be realizable or practical, and only serve to clarify the nuances between the three tasks.

For each of the tasks, our goal is to transform an input text example x with attribute y into some output x' with target attribute y'. Let $d(x_1, x_2)$ be some distance metric and let $C(x)$ be an appropriate classifier for the desired transformation task (e.g. binary sentiment classification). Let $E_c(x)$ be an encoder which preserves the content of the input x independent of its style (we can define a complementary encoder $E_s(x)$ which preserves style independent of content, but do not require it for our formulation). Finally, let $D(x)$ be an adversarially-trained ”fluency discriminator” which outputs a binary classification of 1 if x is determined to be a ”fluent” or human-like text object and 0 otherwise.

Style transfer can be formulated as follows:

$$x^* = \underset{x'}{\operatorname{argmin}}\ d(E_c(x), E_c(x')) \tag{1}$$
$$s.t.\ \ C(x') = y',$$
$$D(x') = 1.$$

Adversarial attacks can be formulated as:

$$x^* = \underset{x'}{\operatorname{argmin}}\ d(x, x') \tag{2}$$
$$s.t.\ \ C(x') = y'.$$

Lastly, **Counterfactuals** can be formulated as:

$$x^* = \underset{x'}{\operatorname{argmin}}\ d(x, x') \tag{3}$$
$$s.t.\ \ C(x') = y',$$
$$D(x') = 1.$$

The key similarity between style transfer tasks and counterfactual generation is in their constraints: both require the transformed sample to retain fluency and ”naturalness”. While adversarial attacks optimize the same objective as counterfactual generation, they do not require the transformation imposed on the input sample to be realistic. Therefore, it is more relevant to our problem to extend models trained for style transfer rather than for adversarial attacks.

As we mention earlier, our key challenge is to minimize the change in style as well as content. Under our optimization formulations, this can be seen as reconciling the different objectives for the style transfer and counterfactual generation tasks.

3 Related Work

A common thread among recent studies in unsupervised style transfer is the need to improve upon previous methods which learn separate style and content representations for text. These methods attempt to disentangle content and attribute embeddings for text so that the attribute or style transfer task can be posed as a simple manipulation of the learned latent style space (Sutskever et al., 2014). However, these methods typically result in output text which is too highly stylized with poor content preservation and make it difficult to tune the trade-off between content preservation and degree of stylization (Wu et al., 2019).

Wang et al. propose to learn an entangled representation of both style and content in order to maintain the integrity and fluency of language expressions (Wang et al., 2019). The authors learn latent representations of text using a transformer-based autoencoder and utilize a surrogate loss based on a classifier’s probabilistic outputs on text samples in order to provide a direction for gradient descent that only alters the style of chosen examples. This method allows the authors to control multiple aspects of sentiment and attribute at once and allows fine-tuning of the degree of style alteration. The authors report state-of-the-art results on Yelp, Amazon, and caption sentiment datasets.

Reinforcement learning (RL) approaches offer a way to perform style transfer without requiring a differentiable loss function, which is an advantage in cases with black-box classifiers whose parameters we cannot access. Luo et al. utilize policy gradient training on two parallel agents which transfer sentiment between two non-aligned corpuses of text with differing sentiment (Luo et al., 2019). The authors encode dual rewards for maintaining content and changing style, which they trade-off using the harmonic mean of the two quantities. Wu et al. approach the problem using two RL agents as well, but instead use a hierarchical system in which a ”point” agent proposes the best token in a text sample to operate on and an ”operate” agent alters the sentence with the optimal token to shift styles (Wu et al., 2019). The authors use a multi-faceted reward signal which explicitly encodes a language model reward to ensure fluency of the generated output.

Sudhakar et al. propose a three-step method in which a ”delete” transformer identifies and removes stylistic elements from text, a ”retrieve” model offers a set of possible replacement stylistic tokens for a target style, and a ”generate” transformer-based decoder outputs a final text sample in the desired style (Sudhakar et al., 2019). The authors report state-of-the-art results on sentiment, gender, and political slant attributes across five datasets. Unfortunately, (Wang et al., 2019) do not compare results with this framework, despite the fact that they test on three of the same datasets and released their study four months later.

As mentioned earlier, augmenting a dataset with adversarial examples leads to increased robustness and better classification performance during training (Ilyas et al., 2019). Kaushik et al. demonstrate a similar effect with augmenting data using counterfactual text examples (Kaushik et al., 2020). The authors find that supplementing datasets with human-generated, quality-controlled counterfactual text samples increased the performance of text classifiers trained on both original and counterfactually-revised data over simply training on original data. The main explanatory result is that spurious features in text samples (e.g. movie genre in reviews) no longer strongly contributed to sentiment classification and that the classifiers instead sought out tokens which had a stronger causal relationship with sentiment. We propose to use this downstream training of classifiers on augmented data as a means of corroborating our generated counterfactuals; if we are able to successfully utilize style transfer models for counterfactual generation, then we may expect to see a similar improvement in performance on classifiers trained on both our generated samples and the original data.

4 Methods

We extend the style transfer models of (Wang et al., 2019) (referred to as CUTAT) and (Sudhakar et al., 2019) (referred to as TDRG) to generate counterfactuals. Based on our discusssion in section 2 our main goal is to change the classification decision of a discriminative model on an example transformed by a counterfactual generator, while minimizing the degree of style transfer and change in content. We examine the two models to identify parameters and algorithm components to alter to appropriately adapt them for counterfactual generation.

4.1 CUTAT

The style transfer model introduced in ”Controllable Unsupervised Text Attribute Transfer via Editing Entangled Latent Representation” (CUTAT) consists of three main components (Wang et al., 2019). These components are:

- **Autoencoder:** a transformer-based autoencoder that learns latent embeddings z from text input x. Does not separate attribute or style words embeddings from content embeddings and instead learns an entangled representation of both. Trained only on the text corpus that test samples will originate from.

- **Classifier:** a simple discriminative network with two linear layers and a sigmoid at the output. Takes a latent representation, z, from the encoder as input. Like the autoencoder, trained only on the text corpus that the test samples originate from.

- **Fast-Gradient-Iterative-Modification (FGIM) algorithm:** given an original latent representation z, target style/attribute y', and a set of weights $\mathbf{w}$, iteratively modifies z with gradient descent using weight $w \in \mathbf{w}$ to produce a new latent representation z' that has the desired attribute y'. Depends on the classifier for cross-entropy loss on the predicted label of the modified z with respect to y' to provide a descent direction, and also for terminating the algorithm when it converges to z' with correct style y'. Performs gradient descent multiple times, once with each weight $w \in \mathbf{w}$.

The details of CUTAT immediately admit an obvious change to extend the model to counterfactual generation. The weights $\mathbf{w}$ used by the FGIM algorithm determine the degree of alteration made to the original latent embedding z. Setting a low w will move z in a neighborhood around its original location, searching for an embedding which meets the desired style. Conversely, a high w shifts z to a large degree with each step, searching for a resulting z' with a large amount of style change from z. The authors present results with $w \in \{1, 2, ..., 6\}$. For our purposes, we want a small value of w to minimize the degree of style change. In addition, a small value of w keeps the resulting z' from straying too far from its initial point and therefore from altering the non-style content too much. We set $w = 2$, since we find this to be the smallest value of w for which the FGIM algorithm consistently converges.

We can also consider improvements to the autoencoder and classifier to improve the quality of learned representations and their classes, and therefore, of generated counterfactuals. Since the heart of CUTAT is the FGIM algorithm, which is what actually produces desired samples, we have flexibility with the models we are able to consider. However, we focus now on the weights of FGIM since they are what immediately influence our counterfactual generation goals, and since the models originally used by the authors already produce good results on several datasets.

4.2 TDRG

The two style transfer models Blind and Guided Generative Style Tranformer (B-GST & G-GST) introduced in ”Transforming Delete, Retrieve, Generate Approach for Controlled Text Style Transfer” (TDRG) are based on extending (Li et al., 2018) to use classifiers based on pre-trained Transformer models, specifically a BERT-uncased model via the hugging-face library[1] as opposed to RNNs and to use a brute-force approach to discover which layer and attention head combination most contributes to a sentence's classification as opposed to the n-gram saliency method used in the prior work in order to best discern which words in an example are attributes vs content.

4.2.1 Shared pre-processing steps

For a given dataset D on which a user wants to perform unsupervised style transer, the TDRG models perform the same initial pre-processing steps of (1) fine tuning a BERT classifier on D and (2) then running a brute-force algorithm inspired by (Feng et al., 2018) to discover which single layer/head's attention weights to use as the basis for the rest of algorithm. Because of the importance of (2), we'll delve into it in greater detail, but also refer readers to section 2.1 of (Sudhakar et al., 2019) and the accompanying code[2]

The brute force mechanism (2) iterates over every layer (12) and every attention head (12 per layer) and then for example in a development set, takes that layer/head's weight vector and calculates an importance score for every token with respect to it via a softmax function. The algorithm then removes the top scoring tokens from the sentence and measures the new resulting classification score which is scored as a smoothed ratio of the probability of the true label vs that of the other label. The layer/head combination which returns the lowest average score ratio of the dev set is selected. With the idea being that removing such attributable words should make the sentence more ”neutral” and layer/head which identifies those attribute words the best will return the lowest score.

The percentage of words to remove is a hyperparameter set to 25% by default, the amount to smooth by is another hyperparamter set to 0.1 by default and the size of the dev set is set to 100 and randomly selected from the dev set.

4.2.2 Blind GST

Once the layer/head has been selected, it is then used to identify the attribute words from each example and then the training objective is to reconstruct the original sentence given solely its content words (ie, non attribute words) and its true class (either ”NEG” or ”POS” for binary classification). Appendix A shows the format of the training data used. Here the underlying model trained is a pre-trained OpenAI-Generative Pre-trained Transformer (GPT1) decoder model. The idea is that once this ”blind” model is trained to learn how to generate ”POS” and ”NEG” sentences (filling it in) given content words and label, we can simply switch which label is fed to the GPT model for our desired output.

4.2.3 Guided GST

The Guided GST is similar, except that instead of being given a label and content words as input, it is instead given attribute words and content words and asked to combine the content words and attributes to generate the original sentence during training. The model being trained again is a GPT and it is ”guided” by the attributes its fed towards generating an output whose class will match those from where the attributes where taken. At inference time, when we want to generate a new sentence for a different target class, we need a mechanism by which to first find the best target attributes to use. The resulting layer/head from the brute force search attention however may not be used because it only provides the attribute words for a given source sentence, but in this case we first need to retrieve a similar sentence from the training set of the target class and then we may use its attributes as found by the brute force mechanism earlier. Here the paper find weighted TF-IDF representations for the test

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

[2] https://github.com/agaralabs/transformer-drg-style-transfer/blob/master/Head_selection.ipynb

content sentence and all training sentences of the target class, and return the target training sentence's attributes based on cosine similarity.

Both models additionally use teacher forcing(Bengio et al., 2015) during training and during output they use beam search (k=5) to improve results.

4.2.4 Counterfactual Generation improvements

One of the immediate concerns about the current TDRG method is the brute force algorithm. The code provided by the authors is very detailed and we are able to mostly reproduce their findings for the yelp models (B-GST and G-GST) they provide (see appendix A.3) and as such we notice that the overall average scores for for the 144 layer/head combinations were quite similar (ie, no clear winner). In such an instance using just one layer/head seems to be limiting quite greatly the ability of the model to discern attributes and its quite dependent on the choice of hyper parameters.

An immediate path to thus consider is the use of more standard and less costly feature attribution techniques such as either Integrated Gradients(Sundararajan et al., 2017) or Expected Gradients (Erion et al., 2019) in place of the brute-force approach. In addition to lowering the reliance on hyper parameter tuning and human eval, this would also lower the system's reliance on its sentiment classifier model's being well calibrated (which it currently depends on for the importance score ratio it uses); an issue that is particularly relevant for pre-trained transformer models (Desai and Durrett, 2020) and important to consider regardless.

Additionally, although the method does leverage ”transfer learning” in the sense of using pre-trained language and generative models, its still quite dependent on the training set it fine tunes on. For instance, in the three datasets we consider from their experiment 1, the Amazon and Yelp ones each have hundreds of thousands of training examples to retrieve similar sentences and attributes from while the Captions one is smaller and performs much less in terms of BLEU score and perplexity. Augmenting the training set of smaller datasets such as the Captions one with an in-domain data set such as the ACL IMDB dataset for movie reviews would greatly improve its performance in counterfactual generations.

5 Experiments

5.1 Datasets

We evaluate the two models on four datasets for binary classification:

- **Yelp:** sentences from Yelp business reviews with either positive or negative sentiment.
- **Amazon:** sentences from Amazon product reviews with either positive or negative sentiment (He and McAuley, 2016).
- **Captions:** image captions with either romantic or humorous style (Gan et al., 2017).
- **IMDB:** IMDB movie reviews with either negative or positive sentiment (Kaushik et al., 2020).

The Yelp, Amazon, and Captions datasets were considered by (Sudhakar et al., 2019) and (Wang et al., 2019), but not directly compared. We therefore evaluate on them again to account for differences in methodology between the works. The IMDB dataset was introduced by (Kaushik et al., 2020) and was used for the downstream task of augmenting a dataset with counterfactually-altered data to improve classifier performance, as discussed in section 3. Each dataset consists of a training, development, and test set with approximately equal class balance in each set. Each dataset also contains human-created gold standard samples with opposite classification for each sample in the test set. A summary of the datasets is presented in Table 1.

5.2 Metrics

We use several automatic evaluation metrics to assess the quality of our generated counterfactuals:

- **Accuracy:** we measure style/attribute transfer accuracy of the generated text with respect to the desired label with a fastText classifier trained on the respective training data (Joulin et al., 2017).
- **BLEU:** we measure content retention by using multi-BLEU to calculate similarity between the generated texts and human references (Papineni et al., 2002).
- **Perplexity:** we measure the fluency of the generated samples by calculating perplexity with a SRILM language model trained on the respective training data (Stolcke, 2002).

DATASET	STYLE	#TRAIN	#DEV	#TEST	#VOCAB	MAX LENGTH	MEAN LENGTH
AMAZON	NEGATIVE	277,000	1,015	500	58,991	72	16.01
	POSITIVE	278,000	985	500			
CAPTIONS	HUMOROUS	6,000	300	300	8,693	26	14.88
	ROMANTIC	6,000	300	300			
IMDB	NEGATIVE	851	122	243	25,935	490	215.81
	POSITIVE	856	123	245			
YELP	NEGATIVE	180,000	2,000	500	9,640	40	9.61
	POSITIVE	270,000	2,000	500			

Table 1: Statistics for the datasets used in this paper. Max length and mean length refer to the maximum and mean number of tokens for samples in the dataset. #Vocab refers to the number of unique tokens encountered in each dataset.

Ultimately, human evaluation is needed to accurately assess the quality of generated counterfactuals. However, due to time and resource limitations, we only consider automatic evaluation for the time being.

6 Results

The automatic evaluation results on the Yelp and Captions datasets can be seen in Table 2. The results for the Amazon dataset are soon to come.

The CUTAT model underperforms relative to the authors' reported metrics in the original work (Wang et al., 2019). There are two explanations for this: (1) differences in evaluation methodology, (2) optimizing over only one step size instead of multiple. The former is obvious and is unavoidable unless we know the exact specifications of the authors' tests. The latter results from the fact that CUTAT supports multiple step size values, $w \in \mathbf{w}$, per test example and outputs a number of style-transferred texts equal to the number of step sizes considered. As we explain in section 4, we fix $w = 2$ to minimally edit samples and therefore have fewer outputs to evaluate; for samples that would benefit from larger w, this produces suboptimal transformed outputs.

Sample results from the B-GST and G-GST models on the Yelp and Captions data can be seen in Appendix A.4.

6.1 IMDB Dataset

The methods did not perform well on generating counterfactuals for the IMDB dataset, so we omit those results. The three main challenges this dataset poses, and the reasons these methods fail are: (1) small size of the training set, (2) relatively large vocabulary, and (3) extremely large sequence length. There are only 1707 total training examples, and each one contains an entire review - unlike the Yelp and Amazon datasets which contain only sentences from reviews.

Appendix B shows an example of a failed output from CUTAT; though the output seems to be a memorized example, there are no matching examples in the dataset. CUTAT's autoencoder did not converge during training due to the small number of examples and the learned latent embeddings therefore are not very representative of the trends in the data. Since the classifier also depends on the quality of the latent embeddings for its prediction, its generalization abilities suffer and it provides poor gradient directions for the FGIM algorithm.

We can make some changes to the IMDB data to improve model performance in future iterations. First, we can augment the dataset with 1707 additional, human-created training examples matched to the current training set, with opposite sentiments. (Kaushik et al., 2020) collect human references for each training example in their original dataset, but we only use human-created references for the test set in the current study. Second, we can further augment the dataset with the entire ACL IMDB movie review dataset, which contains 50,000 reviews with either positive or negative sentiment. For the current study, we used only the filtered, randomly-sampled version of the dataset introduced by (Kaushik et al., 2020). Finally, we can break up the movie reviews by individual sentences, each with their own sentiment classification. This has the potential to greatly increase the size of our training set and decrease sequence length, but must be done carefully so that, for example, all

METHODS	DATASET					
	YELP			CAPTIONS		
	ACCURACY	BLEU	PERPLEXITY	ACCURACY	BLEU	PERPLEXITY
CUTAT	43.8%	22.9	55.5	69.7%	17.6	19.9
B-GST	65.6%	20.2	39.9	67.7%	11.4	14.1
G-GST	56.4%	18.3	52.1	67.8%	11.1	15.8

Table 2: Automatic evaluation results.

sentences in an overall negative movie review are not classified as negative regardless of content.

References

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *CoRR*, abs/1506.03099.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers.

Gabriel G. Erion, Joseph D. Janizek, Pascal Sturmfels, Scott Lundberg, and Su-In Lee. 2019. Learning explainable models using attribution priors. *CoRR*, abs/1906.10670.

Shi Feng, Eric Wallace, Mohit Iyyer, Pedro Rodriguez, Alvin Grissom II, and Jordan L. Boyd-Graber. 2018. Right answer for the wrong reason: Discovery and mitigation. *CoRR*, abs/1804.07781.

C. Gan, Z. Gan, X. He, J. Gao, and L. Deng. 2017. Stylenet: Generating attractive visual captions with styles. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 955–964.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 507–517, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. In *NeurIPS*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *IJCAI*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Shubham Sharma, Jette Henderson, and Joydeep Ghosh. 2019. Certifai: Counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models. *ArXiv*, abs/1905.07857.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *INTERSPEECH*.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. Transforming delete, retrieve, generate approach for controlled text style transfer. In *EMNLP/IJCNLP*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *NeurIPS*.

Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun. 2019. A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *ACL*.

A Training Format for TDRG

A given yelp example with negative sentiment:
`ever since joes has changed hands it ’s just gotten worse and worse .`

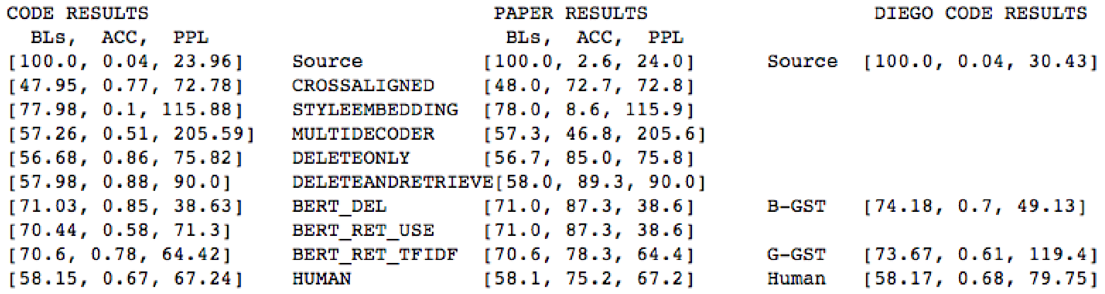
A.1 B-GST model

`<NEG> <CON_START> ever since joes has changed hands it ’ s gotten and . <START> ever since joes has changed hands it ’s just gotten worse and worse . <END>`

A.2 G-GST model

`<ATTR_WORDS> worse ’s <CON_START> ever since joes has changed hands it ’ s gotten and . <START> ever since joes has changed hands it ’s just gotten worse and worse . <END>`

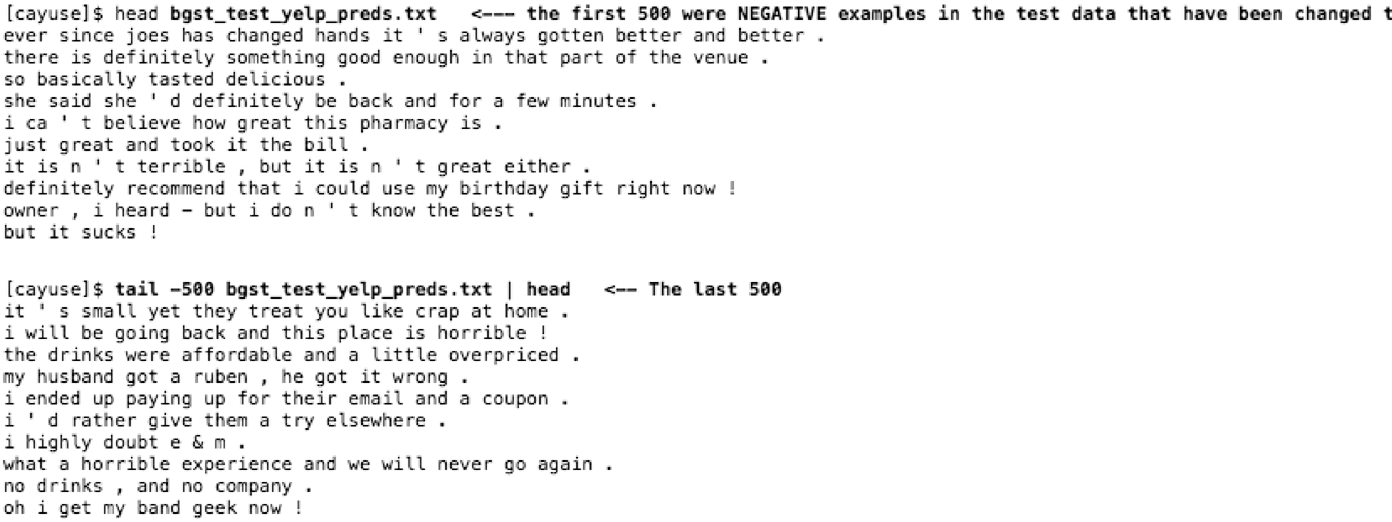
A.3 TDRG reproducibilty



```
CODE RESULTS                                    PAPER RESULTS                       DIEGO CODE RESULTS
  BLs,   ACC,   PPL                               BLs,   ACC,   PPL
[100.0, 0.04, 23.96]    Source            [100.0, 2.6, 24.0]      Source  [100.0, 0.04, 30.43]
[47.95, 0.77, 72.78]    CROSSALIGNED      [48.0, 72.7, 72.8]
[77.98, 0.1, 115.88]    STYLEEMBEDDING    [78.0, 8.6, 115.9]
[57.26, 0.51, 205.59]   MULTIDECODER      [57.3, 46.8, 205.6]
[56.68, 0.86, 75.82]    DELETEONLY        [56.7, 85.0, 75.8]
[57.98, 0.88, 90.0]     DELETEANDRETRIEVE[58.0, 89.3, 90.0]
[71.03, 0.85, 38.63]    BERT_DEL          [71.0, 87.3, 38.6]      B-GST   [74.18, 0.7, 49.13]
[70.44, 0.58, 71.3]     BERT_RET_USE      [71.0, 87.3, 38.6]
[70.6, 0.78, 64.42]     BERT_RET_TFIDF    [70.6, 78.3, 64.4]      G-GST   [73.67, 0.61, 119.4]
[58.15, 0.67, 67.24]    HUMAN             [58.1, 75.2, 67.2]      Human   [58.17, 0.68, 79.75]
```

Figure 1: Our numbers vs those reported in the paper for TDRG on the yelp dataset

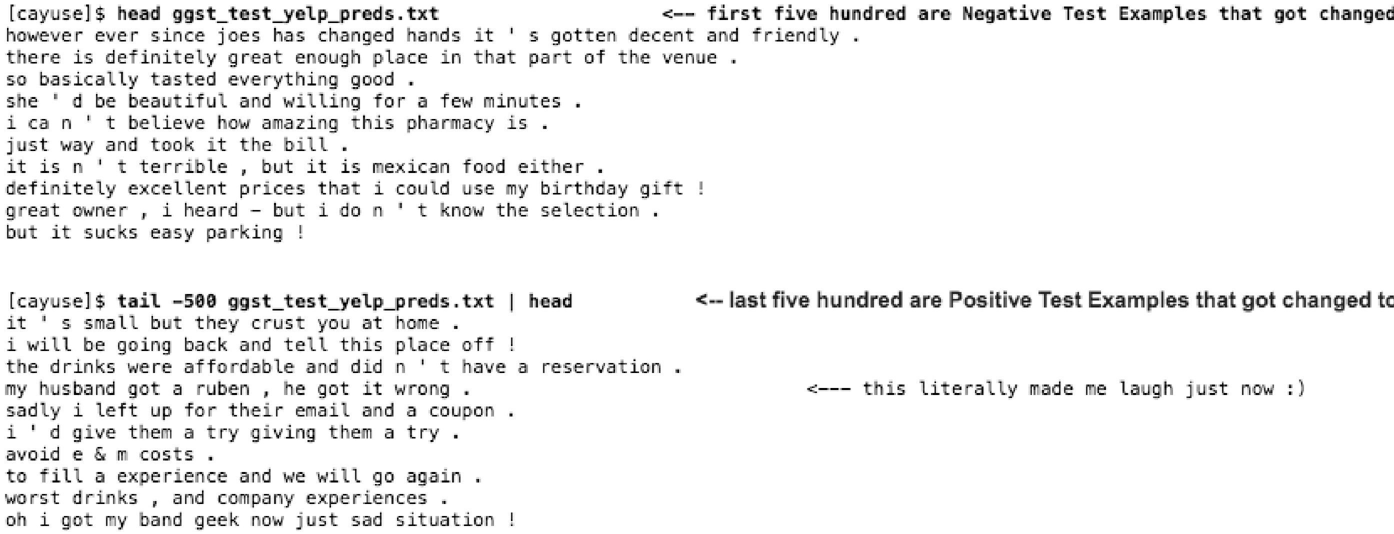
A.4 B-GST and G-GST sample results for Yelp test set



```
[cayuse]$ head bgst_test_yelp_preds.txt   <--- the first 500 were NEGATIVE examples in the test data that have been changed t
ever since joes has changed hands it ' s always gotten better and better .
there is definitely something good enough in that part of the venue .
so basically tasted delicious .
she said she ' d definitely be back and for a few minutes .
i ca ' t believe how great this pharmacy is .
just great and took it the bill .
it is n ' t terrible , but it is n ' t great either .
definitely recommend that i could use my birthday gift right now !
owner , i heard - but i do n ' t know the best .
but it sucks !

[cayuse]$ tail -500 bgst_test_yelp_preds.txt | head   <-- The last 500
it ' s small yet they treat you like crap at home .
i will be going back and this place is horrible !
the drinks were affordable and a little overpriced .
my husband got a ruben , he got it wrong .
i ended up paying up for their email and a coupon .
i ' d rather give them a try elsewhere .
i highly doubt e & m .
what a horrible experience and we will never go again .
no drinks , and no company .
oh i get my band geek now !
```

Figure 2: Positive Yelp Reviews generated from Negative ones (top image) and Negative Yelp Reviews generated from Positive ones (bottom image) using B-GST

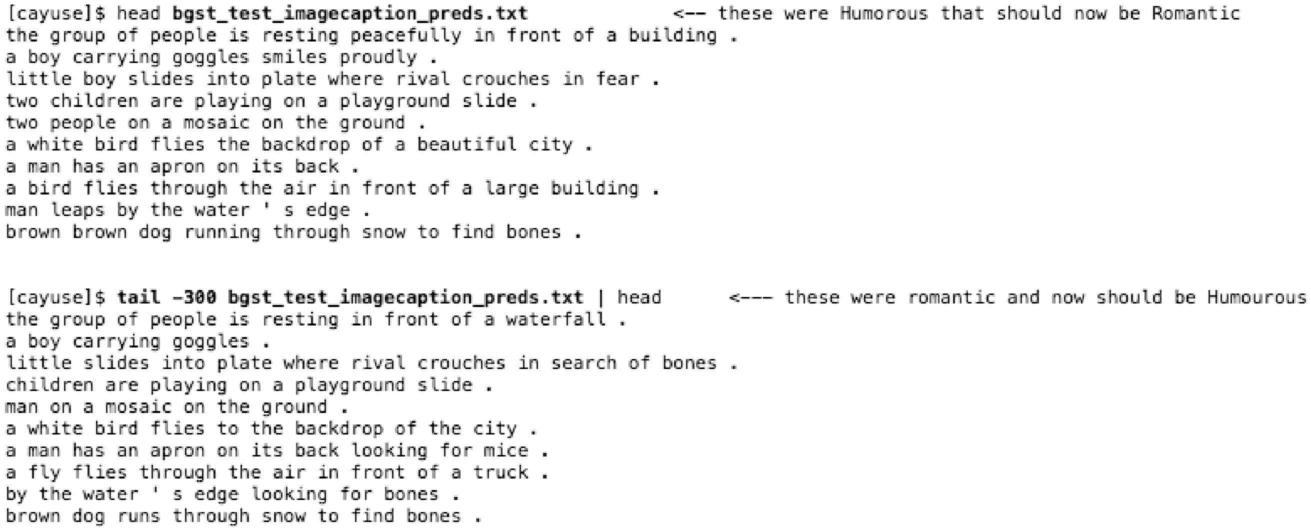


```
[cayuse]$ head ggst_test_yelp_preds.txt                  <-- first five hundred are Negative Test Examples that got changed
however ever since joes has changed hands it ' s gotten decent and friendly .
there is definitely great enough place in that part of the venue .
so basically tasted everything good .
she ' d be beautiful and willing for a few minutes .
i ca n ' t believe how amazing this pharmacy is .
just way and took it the bill .
it is n ' t terrible , but it is mexican food either .
definitely excellent prices that i could use my birthday gift !
great owner , i heard - but i do n ' t know the selection .
but it sucks easy parking !
```

```
[cayuse]$ tail -500 ggst_test_yelp_preds.txt | head           <-- last five hundred are Positive Test Examples that got changed to
it ' s small but they crust you at home .
i will be going back and tell this place off !
the drinks were affordable and did n ' t have a reservation .
my husband got a ruben , he got it wrong .                          <--- this literally made me laugh just now :)
sadly i left up for their email and a coupon .
i ' d give them a try giving them a try .
avoid e & m costs .
to fill a experience and we will go again .
worst drinks , and company experiences .
oh i got my band geek now just sad situation !
```

Figure 3: Positive Yelp Reviews generated from Negative ones (top image) and Negative Yelp Reviews generated from Positive ones (bottom image) using G-GST

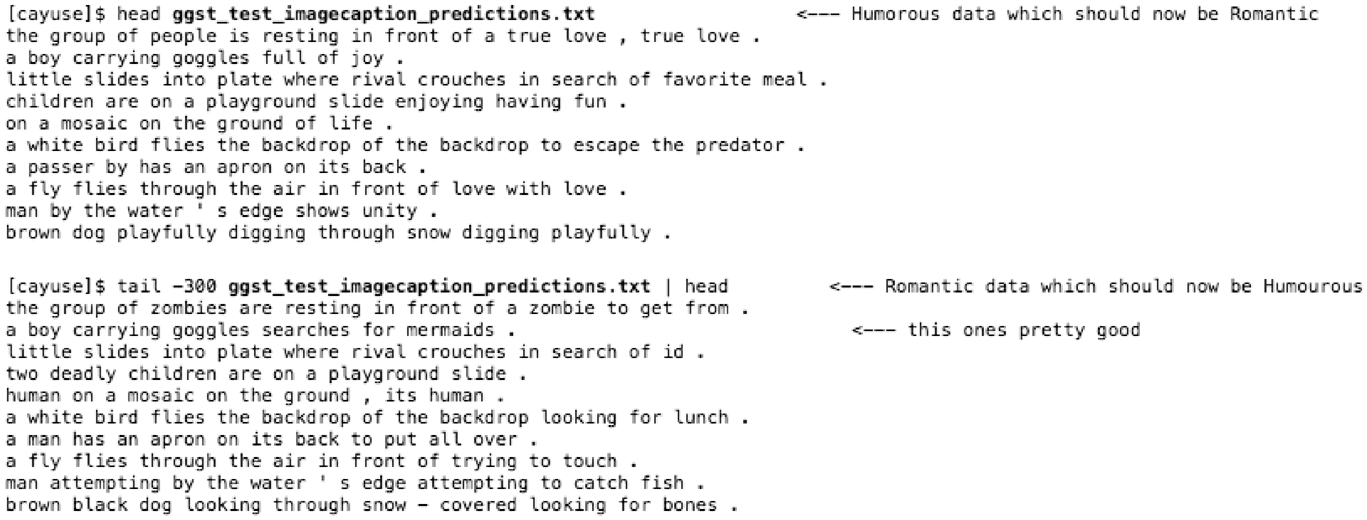
A.5 B-GST and G-GST sample results for Image Caption test set



```
[cayuse]$ head bgst_test_imagecaption_preds.txt            <-- these were Humorous that should now be Romantic
the group of people is resting peacefully in front of a building .
a boy carrying goggles smiles proudly .
little boy slides into plate where rival crouches in fear .
two children are playing on a playground slide .
two people on a mosaic on the ground .
a white bird flies the backdrop of a beautiful city .
a man has an apron on its back .
a bird flies through the air in front of a large building .
man leaps by the water ' s edge .
brown brown dog running through snow to find bones .
```

```
[cayuse]$ tail -300 bgst_test_imagecaption_preds.txt | head      <--- these were romantic and now should be Humourous
the group of people is resting in front of a waterfall .
a boy carrying goggles .
little slides into plate where rival crouches in search of bones .
children are playing on a playground slide .
man on a mosaic on the ground .
a white bird flies to the backdrop of the city .
a man has an apron on its back looking for mice .
a fly flies through the air in front of a truck .
by the water ' s edge looking for bones .
brown dog runs through snow to find bones .
```

Figure 4: Romantic Captions generated from Humorous ones (top image) and Humorous Captions generated from Romantic ones (bottom image) using B-GST



```
[cayuse]$ head ggst_test_imagecaption_predictions.txt          <--- Humorous data which should now be Romantic
the group of people is resting in front of a true love , true love .
a boy carrying goggles full of joy .
little slides into plate where rival crouches in search of favorite meal .
children are on a playground slide enjoying having fun .
on a mosaic on the ground of life .
a white bird flies the backdrop of the backdrop to escape the predator .
a passer by has an apron on its back .
a fly flies through the air in front of love with love .
man by the water ' s edge shows unity .
brown dog playfully digging through snow digging playfully .

[cayuse]$ tail -300 ggst_test_imagecaption_predictions.txt | head        <--- Romantic data which should now be Humourous
the group of zombies are resting in front of a zombie to get from .
a boy carrying goggles searches for mermaids .                               <--- this ones pretty good
little slides into plate where rival crouches in search of id .
two deadly children are on a playground slide .
human on a mosaic on the ground , its human .
a white bird flies the backdrop of the backdrop looking for lunch .
a man has an apron on its back to put all over .
a fly flies through the air in front of trying to touch .
man attempting by the water ' s edge attempting to catch fish .
brown black dog looking through snow - covered looking for bones .
```

Figure 5: Romantic Captions generated from Humorous ones (top image) and Humorous Captions generated from Romantic ones (bottom image) using G-GST

B CUTAT IMDB example

Original Review

if you have n't seen this , it 's terrible . it is pure trash . i saw this about 17 years ago , and I 'm still screwed up from it .

Human Reference

if you have n't seen this , it 's incredible . it is pure gold . i saw this about 17 years ago , and I 'm still hype about it .

CUTAT Output

i have to say anything about this movie three times here . it at the greatest point of all time i 've seen , it 's late it a lot of suspense , but i have probably not only zombie , cry and nostalgia and almost so many people in the same <UNK> <UNK> <UNK> like an important and