Counterfactuals to Control Latent Disentangled Text Representations for Style Transfer

Sharmila Reddy Nangi[1] **Niyati Chhaya**[1] **Sopan Khosla**[2*]
Nikhil Kaushik[3*] **Harshit Nyati**[4*]
[1]Adobe Research, India [2]Carnegie Mellon University, USA
[3]Cohesity Storage Solutions, India [4]Adobe Systems, India
{snangi,nchhaya,hanyati}@adobe.com[1,4]
sopank@andrew.cmu.edu[2] nikhil.kaushik@cohesity.com[3]

Abstract

Disentanglement of latent representations into content and style spaces has been a commonly employed method for unsupervised text style transfer. These techniques aim to learn the disentangled representations and tweak them to modify the style of a sentence. In this paper, we propose a counterfactual-based method to modify the latent representation, by posing a 'what-if' scenario. This simple and disciplined approach also enables a fine-grained control on the transfer strength. We conduct experiments with the proposed methodology on multiple attribute transfer tasks like Sentiment, Formality and Excitement to support our hypothesis.

1 Introduction

Counterfactual Reasoning (Bottou et al., 2013) is leveraged in structured data analysis and econometrics towards generation of alternatives and estimation of alternate scenarios. Counterfactuals describe a causal situation of the form 'If X would have (not) occurred, Y would have (not) occurred' (Molnar, 2019). In interpretable machine learning, counterfactuals have been used to explain predictions of individual instances across various types of datasets and tasks (Neal et al., 2018; Martens and Provost, 2014; Wachter et al., 2017). Laugel et al.(2018) and Neal et al.(2018) use counterfactuals towards generating training data. Counterfactual reasoning also provides us with a unique ability to generate explanations and make causal analysis on the latent space. However, this technique has never been explored in natural language generation tasks. Here, we plug-in the concept of counterfactuals to the text-style transfer task, to enable the manipulation of latent spaces towards controlled transfer of style.

Existing works in text style transfer focus on transferring a specific target attribute. Unsupervised methods based on adversarial attacks (Fu et al., 2018; she), back translation (Prabhumoye et al., 2018), learning disentangled representations(John et al., 2019) have been popular in this domain. Other techniques include deletion of style-specific words and conditionally generate sentences in the target style (Li et al., 2018; Sudhakar et al., 2019). However, all of them fail to provide a control over the target style strength i.e. a clever manipulation of the latent space is non-trivial.

Recent works on controlled text generation include (Wang et al., 2019), which brings in a transformer-based model that modifies the gradient functions leading to controlled generation in the output space. Jin et al.(2019) is an unsupervised approach integrated during end-to-end model training. The drawback in all these efforts is the lack of a prefixed logic towards controlling the latent space. Our proposed method of counterfactuals fills in this gap and provides a logical method to control the latent spaces for enabling a smooth style transfer.

Our approach is based on the premise of disentangled representation spaces inspired from John et al.(2019). Separating out the style and content representations introduce an opportunity to fine-tune, resulting in the ability to control the output sentences specific to style. We **introduce a counterfactual reasoning module for controlling latent disentangled spaces for style transfer**. Figure 1 shows an illustrative example for the variants generated through our approach. To the best of our knowledge, this is the first work leveraging such a concept towards controlled text generation. Through extensive quantitative and qualitative experiments, across attributes and datasets, we conclude that the proposed approach is effective in providing control over the style strength and also shows that the best transfer performance is on par

*Work done while authors were at Adobe Research.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 40–48
August 1–6, 2021.

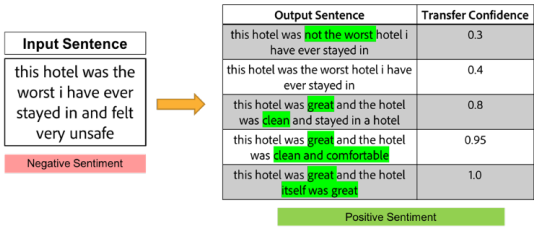


Figure 1: Example Counterfactuals showing the gradual 'control' introduced in the text style transfer.

with the existing baseline style-transfer techniques.

2 Approach

Figure 2 illustrates our proposed approach, that incorporates counterfactual reasoning to latent disentangled representations for manipulating style in text. It consists of (1) A Variational Autoencoder (VAE) model to learn the disentangled style and content representations for different stylistic attributes, (2) A Counterfactual Reasoning Module to control the latent representations for generating style variants.

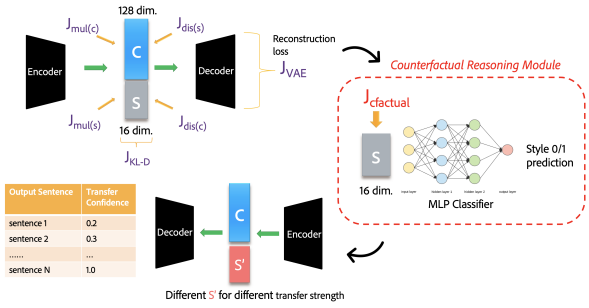


Figure 2: Proposed approach with Counterfactual Reasoning Module for Style Transfer

2.1 Learning Disentangled Representations

We adopt the model described in (John et al., 2019) for learning the disentangled content and style representations. Here, a VAE with an encoder-decoder is used to encode a sentence x into a latent distribution $H = q_E(h|x)$, guided by the loss function:

$$J_{VAE}(\theta_E, \theta_D) = J_{REC} + \lambda_{kl}\mathbb{KL}[q_E(h|x) \parallel p(h)]$$

where, θ_E and θ_D are the encoder and decoder parameters respectively. The first term encourages reconstruction, while the second term regularizes the latent space to a prior distribution $p(h)$ $(\mathcal{N}(0,1))$. We experiment with some variations of this architecture, which are detailed in section 3.

Additionally, Multi-Task $(J_{mul(s)}, J_{mul(c)})$ and Adversarial losses $(J_{adv(s)}, J_{adv(c)})$ are imposed on the latent space h to disentangle the embeddings into representing content c and style s, i.e., $h = [s; c]$, where $[;]$ denotes concatenation. These four losses ensure that the style and content information are present in, and *only* in their respective style(s) and content(c) embeddings.

Once we have the disentangled representations, our basic idea is to feed the generative model with the *same* content and a *different* style embedding to produce sentences of altering style. In (John et al., 2019), the average style embeddings of the target style is fed to the decoder. Intuitively, changing these style embeddings will produce different variants of target style sentences, but a disciplined approach to generate smooth style variants of the sentence is missing. We propose the counterfactual reasoning for this purpose.

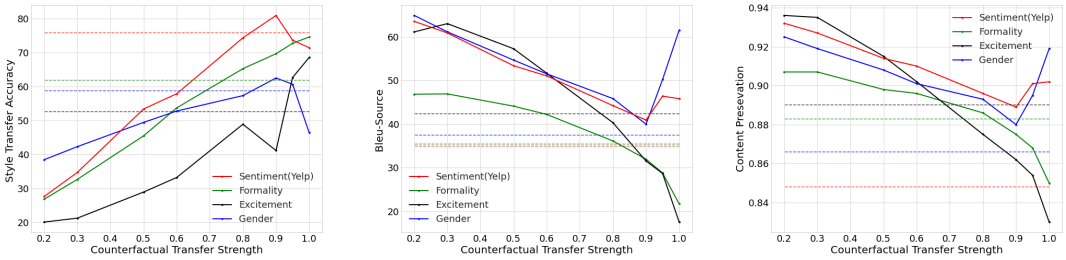
2.2 Counterfactual Reasoning Module

Counterfactuals (CF) are used for gradually changing the style representation along the target-style axis. A counterfactual explanation of an outcome Y takes the form 'if X had not occurred, Y would not have occurred'. We leverage this notion here. A Multi-layer Perceptron (MLP) classifier is trained on the disentangled style latent representations learnt by the VAE, such that every instance of style embedding s, predicts a target style (T) of a sentence. Now, the aim is to find s' such that it is close to s in the latent space but leads to a different prediction T', i.e. the target class. The CF generation loss is given by,

$$J_{cfactual} = L(s'|s) = \lambda(f_t(s') - p_t)^2 + L_1(s', s),$$

where t is the desired target style class for s', p_t is the probability with which we want to predict this target class (perfect transfer would mean $p_t = 1$), f_t is the model prediction on class t and L_1 is the distance between s' and s. The first term in the loss guides towards finding an s' that changes the model prediction to the target class and use of the L_1 distance ensures that minimum number of features are changed in order to change the prediction. λ is the weighting term. The resulting set of CFs are obtained by optimizing (Wachter et al., 2017) the following equation: $arg\min_{s'} \max_{\lambda} L(s'|s)$, subject to $|f_t(s' - p_t)| \leq \epsilon$ (tolerance parameter).

The CF generator is generalizable across different stylistic attributes. To generate multiple variants for a target style, CFs are generated varying



(a) Accuracy (b) Bleu-S (c) Cosine Similarity (CP)

Figure 3: Performance of the counterfactual model on multiple datasets. Style transfer accuracy (ACC) increases and the content preservation (BLEU-S, CP) decreases with increasing transfer strength.

the probability of target specific generation (or confidence), p_t. This results in different sentence variants with a similar target style but varied degrees for transfer strength. Finally, the disentangled representations enable finer control over the style dimensions with no risk of content loss during the counterfactual reasoning stage (as the content representations are retained).

3 Experiments

3.1 Proposed models

The VAE model adapted from (John et al., 2019), with RNN encoder-decoder blocks is R-VAE. We experiment with a variation by replacing RNNs with the transformer blocks (T-VAE). T-VAE-CF uses counterfactuals for generating variants, while models with -AVG use average style embedding of the target style to enable transfer. For T-VAE, we experimented with different loss combinations.-1,-2,-3,-4 refers to the inclusion of $J_{mul(s)}$, $J_{mul(s)} + J_{adv(s)}$, $J_{mul(s)} + J_{adv(s)} + J_{mul(c}$, $J_{mul(s)} + J_{adv(s)} + J_{mul(c)} + J_{adv(c)}$, respectively along with J_{VAE} in the overall loss function.

3.2 Baselines

We compare our best transfer models (with $p_t \approx 1$) against standard unsupervised style-transfer approaches. CrossAligned (CA)(Fu et al., 2018) aligns the hidden representations of original and style transferred sentences. T-D and T-DRG (Sudhakar et al., 2019) models delete attribute related words and conditionally generate words with the target style through transformer architecture.

3.3 Implementation

The counterfactual module has a linear classifier with a sigmoid activation, taking input dim. of 16 (s) and a output dim. 2 (style label). It is trained with Adam optimizer and 0.001 learning rate is used to minimize CCE loss. The transfer strength in CF-module, p_t, is varied from 0 to 1. Experiments with the following values $(0.2, 0.3, 0.5, 0.5, 0.8, 0.9, 0.95, 1.0)$ are reported.*

3.4 Datasets

We experiment with varied style attributes using 5 datasets. YELP is used for sentiment. Human gold standard references of these datasets from (Sudhakar et al., 2019) are used for evaluation. GYAFC dataset (Rao and Tetreault, 2018) is used for Formality and a new dataset GYAFC-excite with custom annotations for excitement is created†. POLITICAL (Voigt et al., 2018) and GENDER (Reddy and Knight, 2016)(similar to (Prabhumoye et al., 2018)) are used for the respective styles. The train-dev-test split as defined by original authors are used for all experiments.

3.5 Evaluation criteria

Style transfer accuracy (ACC) is measured by a dataset-specific Fasttext style classifier (Joulin et al., 2017). The classifiers report a % accuracy of 93.6, 87.6, 82.5, 78.3, 93.5 on the Yelp, GYAFC, GYAFC-Excite, Gender and Political datasets. *Content preservation* is measured through BLEU(Papineni et al., 2002) scores calculated against the source sentences(BLEU-S) and human references (BLEU-H), if available. We compute the cosine similarity (CP) to measure the vector-space similarity‡. *Language fluency* (PPL) is reported by

*Other implementation details, hyper-parameters, compute setup, and training times are provided in the appendix

†We cannot share the GYAFC-excitement dataset due to its license

‡Sentence embeddings for CP are calculated by concatenating the min, max, and mean of its word embeddings, ex-

Attribute⟶		Formality		Sentiment		Excitement	
Direction⟶		Formal⟶ Informal	Informal⟶ Formal	Positive ⟶ Negative	Negative ⟶ Positive	Less ⟶ More	More ⟶ Less
Source		it is **another way** to say that they don't like **you**	**hell yeah** for the first answer that girl answered for me	i always have a **great** dish here to eat	the wine was very **average** and the food was **even less**	it is a small **enjoyable** club	**wonderful** venue for tiff
Our Approach (CF Strength)	0.3	it is way to say it	yeah girl answer that question	i always have a *great* dish here to eat	the wine was very *average* and the food was *even good*	it 's a *good* club	*wonderful* venue
	0.5	you don't like it but it is way	yeah you should answer your question	i always have a *bad* dish here to eat .	the wine was very *average* and the food was *even better*	it 's a *great* club	*great* venue for tiff
	0.8	you can say it to you	oh girl answer that question	i always *have n't* been a though to go to order	the wine had *very unique* and the food was *excellent* too	it 's a *great* club	*good* venue for tiff
	0.9	you don't like it but it is way	oh my answer is *yes*	i always *do n't have a reviews* here to eat something .	the wine was very *reasonable* and the food was *even perfect*	it's a *great* club in vegas	*nice* venue
	0.95	u can say it to u	oh my answer is to answer that question	i always have a *bad* dish to eat here .	the wine had **very authentic** and the food was also **good**	it's a *great* club *in vegas*	*good* venue
	1.0	**u** can say **u r** a way	**oh my answer** is to answer that question	i **do n't always be having** a review to go here	the wine had *very unique* and the food was *excellent*	**absolutely loved** this club	**good** venue
Base	Avg	just say that way you don't know	answer the book for him , because i love that is what	i always do n't get home from a reviewer here	the wine was top notch and the food was even more	it is a small club and a fantastic museum	venue for wonderful for the after ballet

Table 1: Examples for Formality, Sentiment and Excitement with varying CF Strength using our framework.

MODEL	SENTIMENT(YELP)					FORMALITY				EXCITEMENT			
	Acc↑	Bleu-S↑	Bleu-H↑	CP↑	PPL↑	Acc↑	Bleu-S↑	CP↑	PPL↑	Acc↑	Bleu-S↑	CP↑	PPL↑
CA	76.6	47.95	37.15	0.92	-19.97	55.27	24.83	0.90	-19.08	78.25	33.43	0.87	-10.68
T-D	85.7	71.03	**54.08**	0.96	-20.12	46.55	**70.96**	0.95	-24.95	**83.85**	**69.04**	0.94	-13.52
T-DRG	77.4	70.60	54.00	0.96	-21.08	41.23	68.12	0.95	-26.91	74.15	63.65	**0.94**	-15.68
R-VAE-AVG	88.4	34.00	31.10	0.91	-15.08	69.02	32.78	0.90	**-15.18**	71.3	41.22	0.90	**-9.63**
R-VAE-CF	77.5	34.74	31.35	0.91	**-15.04**	62.17	32.47	0.91	-16.98	53.75	42.27	0.90	-9.83
T-VAE-AVG	76.9	34.39	29.19	0.88	-21.25	61.79	35.41	0.88	-23.05	52.55	42.36	0.89	-15.33
T-VAE-CF	**89.8**	34.61	29.49	0.88	-22.58	**74.64**	21.72	0.85	-23.74	68.6	17.57	0.83	-14.60

Table 2: Style Transfer Accuracy. Values for best performing models are reported in -CF variants.[For YELP p_t =: (T-VAE-4-CF,0.9); For FORMALITY(T-VAE-1-CF,1.0); For EXCITEMENT(T-VAE-1-CF,1.0)][‡‡]

MODEL	GENDER				POLITICAL			
	Acc	Bleu-S	CP	PPL	Acc	Bleu-S	CP	PPL
T-D	50.6	**82.50**	**0.97**	-39.05	74.0	**79.40**	**0.94**	-46.74
R-VAE-AVG	52.65	50.42	0.92	**-12.57**	100.0	10.56	0.86	**-26.65**
T-VAE-AVG	58.75	37.48	0.87	-18.22	**92.4**	33.25	0.88	-30.91
T-VAE-CF	**62.55**	39.99	0.88	-18.53	73.20	43.90	0.90	-30.17

Table 3: Gender & Political [For GENDER, p_t: (T-VAE-2-CF,0.9) .For POLITICAL:(T-VAE-2-CF,1.0)]

the perplexity of trigram KL-smoothed language model(Kneser and Ney, 1995), trained on the same corpus.

4 Results and Analysis

Transfer Control. Figure 3 shows the performance of CF variants across metrics for different styles. The CF generated variants from T-VAE-CF (solid lines) are compared against the reference values which take avg. embeddings (T-VAE-AVG) for target style (dotted lines). To recollect, the higher the CF transfer confidence (strength), the closer is the generated variant to the target attribute. Thus, the ideal performance is to have the highest accuracies for the highest CF confidence values (see figure 3(a)). Note that CF strength = 1 alludes to perfect transfer. This is difficult to achieve as CF in the representation space may not be generated for such a strict target. Hence, the variants generated with near perfect transfer target (CF strength = 0.8,0.9,0.95) show the best performance across metrics. The low transfer accuracies for models with low CF confidence establishes the ability of the model to stay near the source when the target strength is low. All models implemented with transfer control report improved performance w.r.t BLEU scores establishing the utility of the alternatives generator.

Table 2, 3 compares baselines with the proposed models. Note that the evaluation metrics for text style-transfer cannot be compared in isolation. There is always a trade-off between content preservation and transfer accuracy. Amongst the baselines, we observe that T-D and T-DRG report high content preservation with some loss in accuracy, but these models only cater to generating a single output sentence and there is no provision to generate the variants. Note that in most style dimensions, T-VAE based models show highest performance in transfer accuracy with good content preservation (CP), but, lower BLEU-S score. The lower BLEU-S scores indicates the ability of our model to generate variants that are not mere repetition of the input samples. R-VAE models show impressive perplexity values. For the

cluding stopwords(Fu et al., 2018)

political dataset, R-VAE baseline shows very high transfer accuracy but takes a tremendous hit in content preservation (BLEU), which is improved with the use of counterfactuals. Examples in Table 1 illustrate the gradual changes introduced by T-VAE-CF across different styles.

Human Evaluation: We conducted a crowdsourcing based experiment (through Amazon Mechanical Turk) to understand both - (A) How baselines compare to the generated text and (B) The interpretation of control as seen by human annotators. For the first experiment, the annotators were presented with sentences generated by our model, baselines and ground truth to evaluate and rank. Specifically, they were asked to score each of the output sentences on a Likert scale of range 1-5 across three aspects - transfer strength, content preservation and fluency. The key takeaways highlight that the sentences generated by our model are at par in terms of grammar and fluency and are better in terms of transfer control. As against text generated by baselines, the text generated by our proposed models is preferred by humans 70% of times (inter-annotator agreement 0.42).

For the second experiment to evaluate the control, we presented the sentence variants generated through different CFs (by varying p_t) and asked the annotators to rank them from best to worst based on their transfer strength. On an average, 60% individuals could grade the gradual control as intended by the model. If we bucket the sentences into low (with $p_t < 0.4$) and high groups (with $p_t > 0.7$), the annotators' preference for bucketing the output into the right confidence goes up to 73% on average (68% for low, and 81% for high), hence, confirming our hypothesis towards using CF for controlled generation.

5 Conclusion

We introduce the use of counterfactual reasoning towards controlling the latent disentangled representations for text style transfer. Experiments not only establish the superiority of the proposed models across standard metrics for a multitude of styles but also illustrate the utility of the gradual control variable in this model. We further validate the use for CF via a human evaluation establishing improved text attribute transfer.

References

Jennifer L Aaker. 1997. Dimensions of brand personality. *Journal of marketing research*, pages 347–356.

Martín Arjovsky and Léon Bottou. 2017. Towards principled methods for training generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14(65):3207–3260.

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

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *AAAI*.

Zhijing Jin, Di Jin, Jonas Mueller, Nicholas Matthews, and Enrico Santus. 2019. IMaT: Unsupervised text attribute transfer via iterative matching and translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3097–3109, Hong Kong, China. Association for Computational Linguistics.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Florence, Italy. Association for Computational Linguistics.

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.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *ICASSP*, pages 181–184. IEEE Computer Society.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2018. Comparison-based inverse classification for interpretability in machine learning. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations*, pages 100–111, Cham. Springer International Publishing.

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.

Weizhi Li, Gautam Dasarathy, and Visar Berisha. 2020. Regularization via structural label smoothing. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 1453–1463. PMLR.

Dayiheng Liu, Jie Fu, Yidan Zhang, Chris Pal, and Jiancheng Lv. 2020. Revision in continuous space: Unsupervised text style transfer without adversarial learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8376–8383.

David Martens and Foster Provost. 2014. Explaining data-driven document classifications. *MIS Q.*, 38(1):73–100.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119. Curran Associates, Inc.

Christoph Molnar. 2019. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/.

Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. 2018. Open set learning with counterfactual images. In *The European Conference on Computer Vision (ECCV)*.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Sudha Rao. 2017. Are you asking the right questions? teaching machines to ask clarification questions. In *Proceedings of ACL 2017, Student Research Workshop*, pages 30–35, Vancouver, Canada. Association for Computational Linguistics.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality 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 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Sravana Reddy and Kevin Knight. 2016. Obfuscating gender in social media writing. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 17–26, Austin, Texas. Association for Computational Linguistics.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "transforming" delete, retrieve, generate approach for controlled text style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3269–3279, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Rob Voigt, David Jurgens, Vinodkumar Prabhakaran, Dan Jurafsky, and Yulia Tsvetkov. 2018. RtGender: A corpus for studying differential responses to gender. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *CoRR*, abs/1711.00399.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

A VAE Models - Further Details

RNN-based (R-VAE). We adopt the model described in John et al.(2019) to disentangle the content and style representations with a recurrent neural network (RNN)-based VAE. The RNN encoder with Bi-GRUs (Cho et al., 2014) learns the hidden representation $q_E(h|x)$ by reading the input $x = (x_1, x_2, ..., x_n)$ sequentially. The RNN decoder, then decodes sequentially over time, predicting the probabilities of each token conditioned on the previous tokens and the latent representation. The reconstruction loss, which is the key loss for the generation objective, is the negative-log-likelihood loss as follows:

$$J_{REC} = \mathbb{E}_{h\sim q_E(h|x)}[-\sum_{t=1}^{n} \log P],$$

$$where \quad P = p(x_t|h, x_1, ..., x_{t-1})$$

The hidden space, h, is separated into 2 spaces while disentangling the style (s) and content (c) representations. Disentanglement is achieved using well-defined auxilliary losses.

Transformer-based (T-VAE). Transformers (Vaswani et al., 2017) have gained popularity for text generation due to their robust architectures. We introduce a transformer-based VAE inspired from Wang et al.(2019). The transformer encoder has a multi-headed self-attention block followed by a feed forward network (FFN). The decoder is similar to the encoder with an additional encoder-decoder attention block. Given an input sentence $x = (x_1, x_2, ..., x_n)$, the transformer encoder, E_{trans} learns a hidden word representation $(z_1, z_2, ..., z_n)$. They are pooled to get a sentence representation z, which is further encoded into a probabilistic latent space $q_E(h|x)$. A sample from this latent representation is given as an input to the encoder-decoder attention block in the decoder. The decoder reconstructs the input sentence x with condition on h. We adopt the label smoothing regularization (Li et al., 2020) while training, for performance improvement. The reconstruction loss (J_{REC}) is :

$$\mathbb{E}_{h\sim q_E(h|x)}[-\sum^{|x|}((1-\epsilon)\sum_{i=1}^{v}\bar{p_i}log(p_i)+ \frac{\epsilon}{v}\sum_{i=1}^{v}log(p_i))]$$

where, v is the vocabulary size, ϵ is the label smoothing parameter, p_i and $\bar{p_i}$ are the predicted and the ground truth probabilities over the vocabulary at every time step for word-wise decoding.

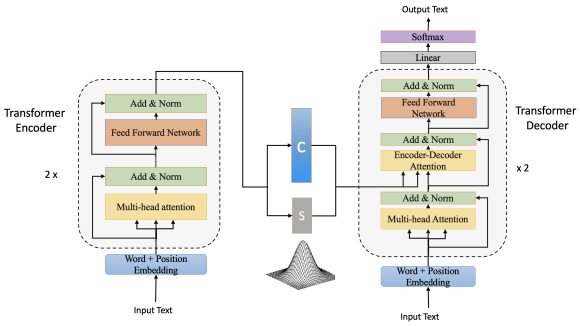


Figure 4: Transformer-based: T-VAE

KL Annealing. We also use an Adam optimiser and KL cost annealing technique (Bowman et al., 2016) to train our model. KL cost annealing refers to slow increase in the weight of the KL term (λ_{kl}) in the loss function from 0 to 1. This aids the training process as the model is warm-started to minimize the reconstruction loss in the initial iterations, followed by a gradual inclusion of KL loss term in the subsequent iterations.

A.1 Loss Functions

Auxiliary loss functions are used to achieve the text rewriting objectives. Note that the reconstruction loss is the primary loss generation but this does not take into consideration the style or the controlled generation.

We use Multi-task and Adversarial losses on the latent space h to disentangle the embeddings into representing content c and style s (i.e., $h = [s; c]$, where $[;]$ denotes concatenation) separately.

Style-oriented losses. Multitask Loss ensures that the style space s is discriminative for the style. We train a style classifier on s jointly with the autoencoder loss.

$$J_{mul(s)}(\theta_E; \theta_{mul(s)}) = -\sum_{l\in labels} t_s(l)\, log(y_s(l))$$

Dataset	Style	#train	#dev	#test	Source
Yelp	Positive	270K	2000	500	https://github.com/lijuncen/Sentiment-and-Style-Transfer/tree/master/data/yelp
	Negative	180K	2000	500	
GYAFC	Formal	48K	2000	950	https://github.com/raosudha89/GYAFC-corpus
	Informal	48K	2000	1250	
GYAFC-excitement	Exciting	36K	1990	1000	NA
	Non-Exciting	36K	1990	1000	
Political	Democrat	270K	2000	28K	http://tts.speech.cs.cmu.edu/style_models/political_data/
	Republican	270K	2000	28K	
Gender	Male	1.34M	2250	267K	http://tts.speech.cs.cmu.edu/style_models/gender_data/
	Female	1.34M	2250	267K	

Table 4: Datasets

where $\theta_{mul(s)}$ are the parameters for style multitask classifier, y_s is the style probability distribution predicted by the classifier and t_s is the ground truth style distribution.

Adversarial loss for style is introduced to ensure that the content space c is not-discriminative of the style. An adversarial classifier is trained, that deliberately discriminates the true style label using the content vector c, with the following loss.

$$J_{dis(s)}(\theta_{dis(s)}) = -\sum_{l \in labels} t_s(l) log(y_s^{'}(l))$$

where $\theta_{dis(s)}$ are the parameters for style adversary, $y_s^{'}$ is the style probability distribution predicted by the classifier on the content space.The encoder is then trained to learn a content vector space c, from which its adversary cannot predict style information. The objective is to maximize the cross entropy $H(p) = -\sum_{i \in labels} p_i log(p_i)$ with:

$$J_{adv(s)}(\theta_E) = H(y_s^{'}|c; \theta_{dis(s)})$$

Content-oriented losses. Multi-task loss aims to ensure that all content information is in the content space c. We define the content information using a bag-of-words (BoW) concept. Here, *part-of-speech* tags , i.e. *nouns* are used. (Liu et al., 2020; DBL) argue nouns in the text are considered as attribute-independent content. This definition allows a generic content loss for all style dimensions as against the previous work where content is defined as bag-of-words in a sentence, excluding stopwords and specific style (sentiment) related lexicon. The content multitask loss is analogical to style multitask loss as follows:

$$J_{mul(c)}(\theta_E; \theta_{mul(c)}) = -\sum_{w \in content} t_c(w)\, log(y_c(w))$$

Adversarial loss for content ensures that the style space does not contain content information. A classifier (content adversary), is trained on the style space to predict the content (BoW) features. Then similar to style, encoder is trained to learn s, from which this adversary cannot predict content information.

$$J_{dis(c)}(\theta_{dis(c)}) = -\sum_{w \in content} t_c(w) log(y_c{'}(w))$$

$$J_{adv(c)}(\theta_E) = H(y_c{'}|s; \theta_{dis(c)}),$$

Training with these losses along with reconstruction loss ensures that the latent space is disentangled, resulting in the final loss given by,

$$J_{total} = J_{VAE} + \lambda_{mul(s)} J_{mul(s)} - \lambda_{adv(s)} J_{adv(s)} + \lambda_{mul(c)} J_{mul(c)} - \lambda_{adv(c)} J_{adv(c)}$$

B Dataset details

The brief descriptions for datsets are as follows:

YELP: Reviews from Yelp. Each review is labeled with a sentiment class - positive or negative. The task is to change the label while rewriting.

GYAFC: Corpus created from a subset of Yahoo Answers. Each sample is tagged either formal or informal. The task is to switch the label.

GYAFC-Excitment: The task here is to convert the sentences from 'exciting' to 'non-exciting'. We create a subset of the GYAFC data where annotators (using Amazon Mechanical Turk), were asked to tag the sentence to be either showing excitement or not. Excitement follows the definition as given by (Aaker, 1997). We follow annotation scheme provided by Rao(2017).

POLITICAL: Comments from Facebook posts from United States Senate and House members. Each comment is labelled is with either Republican or Democrat tag. Task is to interchange between the two.

GENDER: Reviews from Yelp for food businesses. Each review is labeled with either male or female based on the author of the review. Task is to switch between the two.

Table 4 refers to the number of sentences in train-dev-test split available for each dataset. The URL

link to the data files are also provided for each of them.

C Implementation details

The dimensions of c and s are set to 128 and 16 respectively. The posterior probability distributions (μ, σ) learnt for the respective content and style also have the same dimensions. The learnt hidden state representation is converted to 128 (c) and 16 (s) with a linear layer.

For R-VAE, hidden state dimension is set to 256. For the T-VAE, the embedding size, latent layer and the self-attention layers all are set to 256. The inner dimension of FFN in the transformer is set to 1024. Each of the encoder and decoder is stacked with two layers of transformer blocks. We used the Adam optimizer for the VAE and the RMSProp optimizer for the discriminators, following stability tricks in adversarial training (Arjovsky and Bottou, 2017). Each optimizer has an initial learning rate of 10^{-3}. Models are trained for 50 epochs. Figure 4 illustrates the architecture of T-VAE.

Word embeddings initiated with word2vec (Mikolov et al., 2013) are trained on respective training sets. Both, the autoencoder and the discriminators are trained once per mini batch with $\lambda_{mul(s)}$, $\lambda_{mul(c)}$, $\lambda_{adv(s)}$, and $\lambda_{adv(c)} = 1$. The label smoothing parameter in the transformer loss ϵ is set to 0.1. The KL-Divergence penalty is weighted by $\lambda_{kl}(s)$ and $\lambda_{kl}(c)$ on style and content, respectively. During training, we also used the sigmoid KL annealing schedule

The hyper-parameter weights in the loss function $\lambda_{mul(s)}$, $\lambda_{mul(c)}$, $\lambda_{adv(s)}$, and $\lambda_{adv(c)}$ are chosen to be 1, as the values were Observed to be converging over iterations.

We implement our model based on Pytorch 0.4. We trained our models on a machine with 4 NVIDIA Tesla V100-SXM2-16GB GPUs. On a single GPU, our transformer model with all the losses (T-VAE-4) took approximately 0.4 s to train for one step with a batch of size 128. It takes around 10 hours to train our model on 1 GPU. Table 5 depicts the runtime details for all the model variations.

For our counterfactual generator model, we use the counterfactual model from Alibi library in Python[§]. On an average it takes 3 seconds to generate a counterfactual for a given input representation and transfer strength (p_t).

[§] Alibi Counterfactual Module

Dataset	Model	Batch Size	#batches in 1 epoch	Runtime for 1 epoch
Yelp	T-VAE-1	128	2375	247.32s
	T-VAE-2	128	2375	373.75s
	T-VAE-4	128	2375	1108.34s
Formality	T-VAE-1	32	3157	667.85s
	T-VAE-2	32	3157	944.97s
Excitement	T-VAE-1	64	1200	580.61s
	T-VAE-2	64	1200	602.99s
Gender	T-VAE-1	32	3156	333.58s
	T-VAE-2	32	3156	492.12s
Political	T-VAE-1	128	4233	751.92s
	T-VAE-2	128	4233	1050.30s

Table 5: Runtime details of model variations across different datasets

Dataset	Model	Counterfactual Module MLP Classifier	
		CCE Loss (Validation)	Accuracy (Validation)
Yelp	T-VAE-1	0.05	99.25
	T-VAE-2	0.04	99.31
	T-VAE-4	0.04	99.37
Formality	T-VAE-1	0.36	94.09
	T-VAE-2	0.33	97.43
Excitement	T-VAE-1	0.34	96.73
	T-VAE-2	0.22	96.87
Gender	T-VAE-1	0.11	96.17
	T-VAE-2	0.12	96.56
Political	T-VAE-1	0.005	99.992
	T-VAE-2	0.003	99.998

Table 6: Validation loss and accuracy for MLP classifier in counterfactual

Further details of our model summary and generated sentences are present here : https://bit.ly/34DYHP5