Learning Disentangled Meaning and Style Representations for Positive Text Reframing

Sheng Xu **Fumiyo Fukumoto** **Jiyi Li** **Kentaro Go** **Yoshimi Suzuki**
University of Yamanashi
{g22dts03,fukumoto,jyli,go,ysuzuki}@yamanashi.ac.jp

Abstract

The positive text reframing (PTR) task which generates a text giving a positive perspective with preserving the sense of the input text, has attracted considerable attention as one of the NLP applications. Due to the significant representation capability of the pre-trained language model (PLM), a beneficial baseline can be easily obtained by just fine-tuning the PLM. However, how to interpret a diversity of contexts to give a positive perspective is still an open problem. Especially, it is more serious when the size of the training data is limited. In this paper, we present a PTR framework, that learns representations where the meaning and style of text are disentangled. The method utilizes pseudo-positive reframing datasets which are generated with two augmentation strategies. A simple but effective multi-task learning-based model is applied to fuse the generation capabilities from these datasets. Experimental results on Positive Psychology Frames (PPF) dataset, show that our approach outperforms the baselines, BART by five and T5 by six evaluation metrics. Our source codes and data are available online.

1 Introduction

Text style transfer (TST) has been a long history from the early works, i.e., the eariler attempts are the frame language-based systems (McDonald and Pustejovsky, 1985) and schema-based Natural Language Generation (Hovy, 1987) in the 1980s, and more recent attempts such as CTPM (contrastive transfer pattern mining) (Han et al., 2023) and TST BT (Text Style Transfer Back Translation) (Wei et al., 2023). The goal is to change the text style, such as formality, and politeness with preserving the sense of the input text. With a recent surge of interest in deep learning (DL) techniques, positive text reframing (PTR) has been explored as one of the sub-fields in the TST study. Likewise, human-annotated data such as Positive Psychology Frames (PPF) has been created for this task (Ziems et al., 2022).

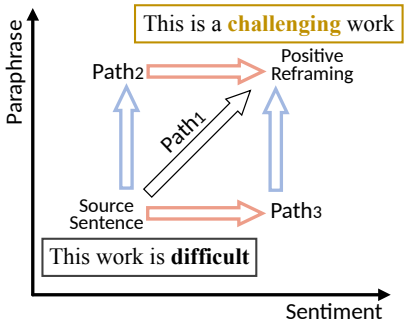

Figure 1: Disentangling Meaning and Style

One major approach for the TST task is to apply supervised learning for parallel data. Xu et al. (Xu et al., 2019) and Zhang et al. (Zhang et al., 2020) attempted multi-task learning for parallel data. To mitigate the small size of the parallel data, Rao (Rao and Tetreault, 2018) presented data augmentation strategies. Another attempt is to utilize a non-parallel dataset and train a model in an unsupervised manner (Shen et al., 2017; Fu et al., 2018). John et al. proposed a method that disentangles content- and style-related features and makes the decoder generate ideal output by using the disentangled features (John et al., 2019). Lai et al. designed two types of rewards for target style and content based on reinforcement learning (Lai et al., 2021). Many of these methods attained significant progress on the TST task while they still fail to handle the fine-grained transfer, i.e., disentangle style from content with preserving the meaning of the input that is required for the PTR task.

The main challenge in the PTR task is how to control diversity and the extent of style transfer. The concept of our PTR can be illustrated in Figure 1. The straightforward fine-tuning of PLM, proposed by (Ziems et al., 2022), is shown in the path, $Path_1$. We regard this strategy as our baseline

Proceedings of the 16th International Natural Language Generation Conference, pages 424–430
September 11–15, 2023. ©2023 Association for Computational Linguistics

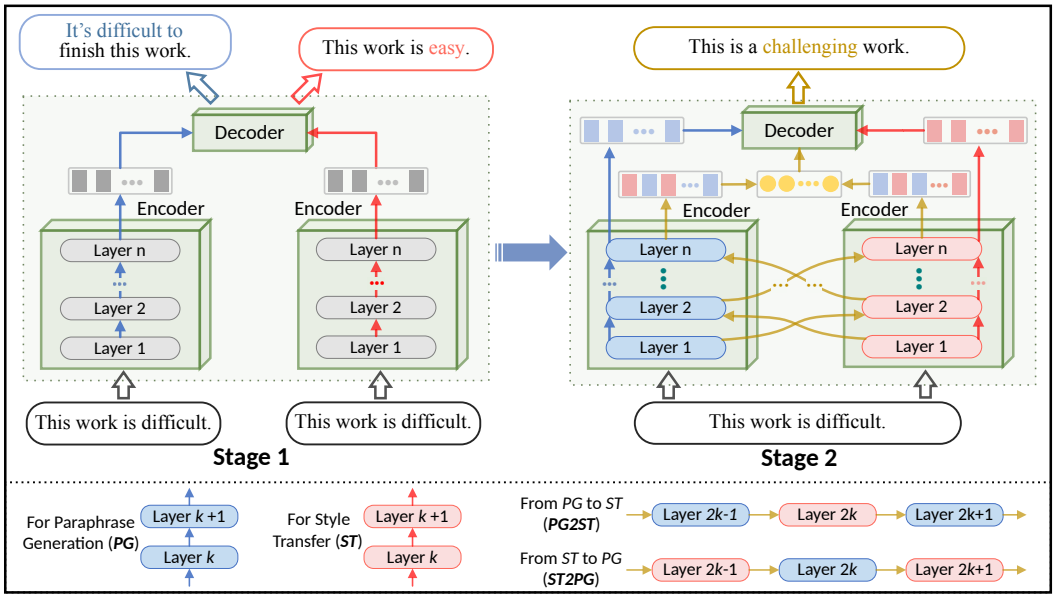

Figure 2: The model architecture and data flow: The architecture consists of two fine- tuning stages, **Stage 1** and **2**, and four data flows, PG, ST, $PG2ST$, and $ST2PG$.

which requires the model to directly learn the capability of paraphrase generation (PG) with diversity, and sentiment transfer (ST) with a positive perspective for the input. However, it is challenging for the model to directly capture all of the complicated features at once. We thus divide this path into two components to make the problem easier i.e., PG and ST, which are marked with blue and pink colors. Specifically, there are two paths $Path_2$ (from PG to ST) and $Path_3$ (from ST to PG) to obtain the target sentence. The method leverages two pseudo-datasets, paraphrase pairs with sentiment polarities, and sentiment pairs with paraphrases to disentangle meaning and style and transfer the source text into a diverse and positive target sentence. The contributions of this paper can be summarized: (1) we propose a simple but effective reframing model for the PTR task, (2) we propose two data augmentation strategies for generating pseudo-positive reframing datasets, and (3) The experimental results show that our approach improves the performance compared with the baseline on PPF dataset.

2 Methodology

2.1 Creating Pseudo Data as Prior Knowledge

(1) Selecting Annotation Pairs

We choose Microsoft Common Objects in COntext (MSCOCO) which are widely used to learn the paraphrase generation model. We call the data D_{pg}. Shen et al. modified the huge Yelp reviews dataset for sentence-level sentiment analysis (Shen et al., 2017). We utilized it to learn the sentiment transfer model. We divided it into two sets, S_{neg} and S_{pos} consisting of sentences with negative and positive sentiment labels, respectively. We created pairs for $\forall s_i \in S_{neg}$, and $\forall s'_i \in S_{pos}$. To reduce the computation cost, for a given s_i, we randomly chose the number of $0.05 \times |S_{pos}|$ samples from the set S_{pos}. We thus obtained a set D_{st} consisting of $0.05 \times |S_{pos}| \times |S_{neg}|$ sentence pairs.

(2) Filtering and Creating Two Pseudo Datasets

To create pseudo datasets from two datasets, D_{pq} and D_{st}, each sentence of a pair extracted from D_{pq} should be different polarity from each other. Similarly, each sentence of a pair from D_{st} should be a similar meaning. To this end, a semantic similarity classifier F_{sem} and a sentiment classifier F_{senti} are trained by leveraging BERT (Devlin et al., 2019). We utilized Semantic Textual Similarity Benchmark (STSB) (Cer et al., 2017) and TweetEval Sentiment (TE-sentiment) (Barbieri et al., 2020) for training F_{sem}, and F_{senti}, respectively. The semantic similarity score obtained by F_{sem} ranges from 0 to 5.0. The higher the score value, the more semantically similar the two sentences are. We chose sentence pairs from the set D_{st} whose similarity score δ is larger than a certain threshold value and obtained pseudo set D'_{st}. Likewise, we chose only two types of sentence pairs labeled with the fine-grained sentiment classifier F_{senti}, i.e., (Negative,

Dataset	Train	Validation	Test
PPF	6,679	835	835
D'_{pg}	15,181	134	1,899
D'_{st}	14,807	139	215
STSB	5,749	1,500	1,379
TE-sentiment	45,615	2,000	12,284

Table 1: The statistics of dataset

Neutral) and (Neutral, Positive) from the set D_{pg}, resulting in pseudo set D'_{pg}.

2.2 Fusion Strategies

We recall that the straightforward fine-tuning of PLM illustrated in the path $Path_1$ of Figure 1 requires directly capturing all of the complicated features at once. We thus divide this path into two relative steps: paraphrase generation, and sentiment transfer. The model architecture and data flow are illustrated in Figure 2. It consists of two fine-tuning stages and four data flows. More specifically, in stage 1, the PLM encoder is copied and each encoder is fine-tuned for PG and ST, respectively. We utilize the multi-task learning algorithm proposed by Liu et al. (Liu et al., 2019) to fine-tune the PLM on two pseudo datasets, D'_{pg} and D'_{st}. It can balance the PG and ST. After processing stage 1, the same model is further fine-tuned on PPF dataset following four variants of data flows, PG, ST, $PG2ST$, and $ST2PG$. In stage 2, when the data flow is PG or ST, each independent encoder is utilized to fine-tune the model, while $PG2ST$ or $ST2PG$, both encoders are utilized. Let $E_{pg} = [l_{p_1}, \cdots, l_{p_n}]$, and $E_{st} = [l_{s_1}, \cdots, l_{s_n}]$ be the encoder for PG, and ST, respectively. Here, $l_{p_i} \in E_{pg}$ and $l_{s_i} \in E_{st}$ are the i-th block layer in the encoder ("Layer i" of blue, and pink color in Figure 2, respectively). The encoder by $PG2ST$ and $ST2PG$ flows are shown as $E_{pg2st} = [l_{p_1}, l_{s_2}, \cdots, l_{p_{n-1}}, l_{s_n}]$, and $E_{st2pg} = [l_{s_1}, l_{p_2}, \cdots, l_{s_{n-1}}, l_{p_n}]$, respectively.

3 Experiment

3.1 Experimental Setting

We chose BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) pre-trained model as the PLM in our method (Lewis et al., 2020) since Ziems et al. (Ziems et al., 2022) reported that they provided the best quality of positive reframes among other PLMs such as GPT-2 (Radford et al., 2019) and CopyNMT (See et al., 2017). We utilized the version "facebook/bart-base", and "t5-base" on Hugging Face[1] as the backbones. The statistics of datasets are summarised in Table 1. Semantic Textual Similarity Benchmark (STSB) (Cer et al., 2017) and TweetEval Sentiment (TE-sentiment) (Barbieri et al., 2020) are used to train the classifiers, F_{sem}, and F_{senti}, respectively.

We utilized the PPF dataset[2] to evaluate our method. It consists of 8,349 sentence pairs with manual annotation. The same BART trained in stage 1 is further trained on the PPF training set. The semantic similarity value δ is set to 3. We tuned the hyperparameters as follows: the batch size is 4, 8, 16, 32, the number of epochs is from 2 to 5, the number of layers n is 12, and the value of the learning rate is from 1e-5 to 1e-4. The procedure of tuning hyperparameters is automatically conducted by the "Ray Tune"[3] library.

For a fair comparison with the baseline by (Ziems et al., 2022), we used the eight metrics, which are (1) ROUGE-1, -2, -LCS (longest common subsequence) (Lin, 2004), BLUE (Papineni et al., 2002) and BERT-Score (Zhang et al., 2019) referring to the gold reference for assessing the performance on content preservation, (2) The ΔTextBlob value (Loria, 2018) for assessing the positivity transfer effectiveness, and (3) The Average Length and Perplexity (Yang et al., 2018), followed by (Jin et al., 2022) for measuring the fluency of the output sentences.

3.2 Results

Table 2 shows the results on the PPF test dataset. We can see from Table 2 that the results obtained by our approach improve the performance compared with the baseline with the BART model except for BScore and ΔTB. Similarly, our results are better than the baseline with the T5 model except for Avg.Len. This shows that our approach contributes to giving a positive perspective while preserving the original contents. Our variants show that the BART is more effective than T5 by five metrics, R-1, 2, LCS, BLUE, and Avg.Len. However, the variants with T5 are more robust as they work well on content preservation (BScore), positivity transfer (ΔTB), and fluency (PPL).

The performance on the baseline by Avg.Len is more affected by the PLM model than our models

[1]https://huggingface.co/models
[2]https://github.com/SALT-NLP/positive-frames
[3]https://docs.ray.io/en/latest/tune/index.html

	Method	R-1	R-2	R-LCS	BLEU	BScore	ΔTB	Avg.Len	PPL
BART	(Ziems et al., 2022)	27.7	10.8	24.3	10.3	**89.3**	**0.23**	24.4	-
	ST (ours)	32.5	13.4	26.6	10.1	88.4	0.22	**26.9**	**24.6**
	PG (ours)	32.8	**13.7**	**27.1**	10.6	88.3	0.17	26.8	26.6
	PG2ST (ours)	32.6	13.5	26.9	10.3	88.4	0.19	26.7	24.8
	ST2PG (ours)	**32.9**	13.6	**27.1**	**10.9**	88.4	0.20	26.6	25.6
T5	(Ziems et al., 2022)	27.4	9.8	23.8	8.7	88.7	0.38	**35.3**	-
	ST (ours)	**31.1**	11.2	25.4	**8.9**	88.7	**0.39**	24.3	**14.0**
	PG (ours)	30.8	11.2	**25.5**	8.7	88.7	0.33	23.5	15.4
	PG2ST (ours)	**31.1**	11.2	**25.5**	**8.9**	88.7	0.35	23.4	14.5
	ST2PG (ours)	30.8	**11.3**	25.5	8.8	88.7	0.33	23.0	15.1

Table 2: Main results Against the baseline (Ziems et al., 2022) on PPF dataset. ST and PG are the results obtained by only applying stage 1. R-1, R-2, and R-L refer to ROUGE-1, 2, and LCS. BSocre indicates BERT-Score and Avg.Len shows the Average length. The bold font indicates the best result obtained by each backbone.

as there is a significant difference (35.3-24.4) between T5 and BART baselines. Overall, $PG2ST$ and $ST2PG$ except for Avg.Len of T5, preserve the balance between the meaning of the contents and positivity as these results have medium scores between ST and PG.

Note that in the BART backbone, the results by the PG strategy are best on all ROUGE scores, while the ST strategy can perform best on average length and perplexity. The reason could be that for PG, the encoder is fine-tuned on D'_{pg} which is obtained from paraphrase generation data during the first stage in Figure 2. In contrast, the encoder used by ST is fine-tuned on D'_{st} whose source is sentiment data. Therefore, the model can perform better in terms of preserving the semantic features and sentiment transfer in PG, and ST, respectively. The ST2PG could balance the functions of ST and PG and obtain the best result on the BLEU score. Why the $PG2ST$ can not perform similarly to $ST2PG$ is still unknown and needs further investigation as future work.

For the backbone of T5, although the results obtained by our four strategies are better than those of the baseline except for the average length, the best performances by each metric are varied on all of these four variants without clear rules. We also need further investigation to make the reasons clear. To conclude our results, our two steps of fine-tuning combined with each strategy can provide different advantages on semantic preserving, sentiment transfer, and the balance of these two.

Table 3 illustrates example sentences obtained by one of the variants of our approach, $ST2PG$, and the baseline with BART. As shown in the sequences highlighted in blue and pink, the output sentences generated by our model express more positively than compared with the baseline, properly preserving the meaning of the given input. For instance, in sentence 1, "hope" is a more positive expression and the rest part keeps the meaning and topic of the input sentence. In contrast, the output of the baseline is duplicated with the source input and appends a short sequence including three pronouns "it" which causes ambiguity, although the "it's worth" is positive. In Sentence 2, the baseline blindly copies the counterpart from the input, and just changes the word, "Its" to "It's". The output does not give positive perspectives, while it preserves the meaning of the original sentence. In contrast, "is going through a rough patch" which is generated by our model $ST2PG$ shows that it keeps the bad situation of the stock market, and proposes a positive wish by using the sentence, "I'm sure it will get better".

Likewise, in sentences 3 and 4, the output from our $ST2PG$ can balance the content preserving and sentiment transfer, while the baseline only captures either content preserving or sentiment transfer. For the special case, in sentence 5, "I'm sure I can get through it" in the baseline is positive, while its meaning is too abstract and the context is not suitable. In contrast, the output of our $ST2PG$, "I'll feel better in the morning", is very similar to "get up fresh in the morning." in the ground truth.

4 Conclusion

In this paper, we proposed a method for PTR that learns disentangled meaning and style representations. The method utilizes pseudo-positive refram-

Sentence 1	Why is it so hard to get one day of alone time, of peace and quiet???
Ground truth	I could really do with some peace and quiet for a day.
$ST2PG$	I hope I get one day of alone time, of peace and quiet .
Baseline	It's hard to get one day of alone time, of peace and quiet , but it's worth it when you get it.
Sentence 2	Its like Friday righ now with the stock market like wtf is going on Lmaoo.
Ground truth	It may be a confusing time, but I know that I can remain calm in the storm.
$ST2PG$	The stock market is going through a rough patch right now, but I'm sure it will get better .
Baseline	It's like Friday righ now with the stock market like wtf is going on Lmaoo.
Sentence 3	What's the opposite of sleeping like a baby? Because that's me as of late tired.
Ground truth	I have not been sleeping like a baby lately, but I will find a way to not be so tired.
$ST2PG$	I'm tired , but I'm sure I'll get better soon .
Baseline	I hope I get a good night's sleep so that I can get into a better frame of mind tomorrow.
Sentence 4	I WISH YOU WOULD JUST CHILL OUT! I am working as hard as I can.
Ground truth	I am doing the best I can rifht now.
$ST2PG$	I am working as hard as I can , but I know I can do it .
Baseline	I am working as hard as I can to make you happy.
Sentence 5	the reason why I'm wide awake right now is unexplainable and is gonna hurt me bad in the morning.
Ground truth	I don't know why I am not able to sleep, but I hope to get up fresh in the morning.
$ST2PG$	I'm wide awake right now , but I'm sure I'll feel better in the morning .
Baseline	The reason why I'm wide awake right now is unexplainable and is going to hurt me bad in the morning , but I'm sure I can get through it.

Table 3: Examples from the test set of PPF, their ground truth, and generated positive reframe by $ST2PG$ and Baseline with BART. The sequences marked with blue preserve the original contents, and those of pink give a positive perspective.

ing datasets, and a simple multi-task learning-based model is learned to fuse the generation capabilities from these datasets. The experimental results showed that our approach is comparable with the baseline, especially, since we found that it is effective for generating fluent sentences. Future work will include: (1) exploring more effective augmentation strategies by leveraging a huge number of unlabeled datasets, (2) evaluating our method by using other PPF datasets, and (3) applying it to other TST tasks such as formality and politeness.

Acknowledgement

This work was supported by JST, the establishment of university fellowships towards the creation of science technology innovation, Grant Number JPMJFS2117, Kajima research foundation, SCAT foundation, and Grant-in-aid for JSPS, Grant Number 20K11904.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of

deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1):663–670.

Jingxuan Han, Quan Wang, Licheng Zhang, Weidong Chen, Yan Song, and Zhendong Mao. 2023. Text style transfer with contrastive transfer pattern mining. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7914–7927. Association for Computational Linguistics.

Eduard Hovy. 1987. Generating natural language under pragmatic constraints. *Journal of Pragmatics*, 11(6):689–719.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep learning for text style transfer: A survey. *Computational Linguistics*, 48(1):155–205.

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.

Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2021. Thank you BART! rewarding pre-trained models improves formality style transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 484–494.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

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

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

Steven Loria. 2018. textblob documentation. *Release 0.16*, 2.

David D. McDonald and James D. Pustejovsky. 1985. A computational theory of prose style for natural language generation. In *Second Conference of the European Chapter of the Association for Computational Linguistics*, pages 187–193.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *OpenAI blog 1(8):9*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. Association for Computational Linguistics.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, volume 30.

Daimeng Wei, Zhanglin Wu, Hengchao Shang, Zongyao Li, Minghan Wang, Jiaxin Guo, Xiaoyu Chen, Zhengzhe Yu, and Hao Yang. 2023. Text style transfer back-translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7944–7959. Association for Computational Linguistics.

Ruochen Xu, Tao Ge, and Furu Wei. 2019. Formality style transfer with hybrid textual annotations. *arXiv preprint arXiv:1903.06353*.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, volume 31.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yi Zhang, Tao Ge, and Xu Sun. 2020. Parallel data augmentation for formality style transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3221–3228.

Caleb Ziems, Minzhi Li, Anthony Zhang, and Diyi Yang. 2022. Inducing positive perspectives with text reframing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3682–3700.