Incorporating Stylistic Lexical Preferences in Generative Language Models

Hrituraj Singh
Adobe Research
hrisingh@adobe.com

Gaurav Verma
Adobe Research
gaverma@adobe.com

Balaji Vasan Srinivasan
Adobe Research
balsrini@adobe.com

Abstract

While recent advances in language modeling has resulted in powerful generation models, their generation *style* remains implicitly dependent on the training data and can not emulate a specific target style. Leveraging the generative capabilities of a transformer-based language models, we present an approach to induce certain target-author attributes by incorporating continuous multi-dimensional lexical preferences of an author into generative language models. We introduce rewarding strategies in a reinforcement learning framework that encourages the use of words across multiple categorical dimensions, to varying extents. Our experiments demonstrate that the proposed approach can generate text that distinctively aligns with a given target author's lexical style. We conduct quantitative and qualitative comparisons with competitive and relevant baselines to illustrate the benefits of the proposed approach.

1 Introduction

With recent advances in unconstrained language generation (Radford et al., a,b; Brown et al., 2020), an emerging direction is to adapt such pre-trained language models to follow certain stylistic constraints (Wang et al., 2019; Syed et al., 2020). These approaches rely on the inherent properties of the training corpus to tailor generation to target characteristics; for example, implicitly learning author-stylized text generation by training on author-specific corpus (Syed et al., 2020) and learning to generate formal text (Wang et al., 2019). However, it is desirable to have explicit control over certain stylistic aspects in such generation, for e.g., emulating lexical choices of an author in a generation, capturing syntactic constructs, inducing sentential preferences (active vs. passive) in generation. To this end, we propose an approach to adapt a pre-trained Transformer-based language model (Vaswani et al., 2017), specifically GPT-2 (Radford et al., b), to generate text that aligns with given lexical elements of style by providing *explicit* rewards in a reinforcement learning framework.

Reinforcement learning (RL) has been successfully applied to several natural language generation tasks like summarization (Paulus et al., 2018) and paraphrase generation (Li et al., 2018). RL overcomes the 'exposure bias' (Ranzato et al., 2015) in cross-entropy based training of language models and allows for optimization with respect to non-differentiable objectives. However, existing explorations around the use of RL in generation tasks have been limited to RNN-based models due to issues surrounding stabilization of RL training on Transformer models (Parisotto et al., 2019). Parisotto et al. further conclude that a Transformer requires reordering of the normalization layer from output to the input streams along with a gated mechanism instead of residual connections to stabilize its training. Building on this, we leverage the shifted position of the normalization layers in GPT-2 to train an RL framework with GPT-2 for aligning generated text to target lexical characteristics.

Recent work by Ziegler et al. (2019) explored RL frameworks with GPT-2 to generate text with different styles. However, they treat their target characteristics (sentiment and descriptiveness) as a binary variable (viz. +ve/-ve). It is non-trivial to extend their work to generate lexically-aligned text, since each of the target dimensions is a continuous value. Our task further requires *simultaneously* aligning along multiple lexical dimensions calling for a rewarding strategy that accounts for multiple dimensions. To this end, our key contributions are: (**1**) an RL framework that introduces lexical style elements in a Transformer-based language generation model; (**2**) a rewarding scheme that incorporates continuous multi-dimensional lexical elements; (**3**) extensive experiments on multiple

authors to show the efficacy of our approach to align generation to an author's lexical preferences.

2 Author's Lexical Style

An author's writing style is a combination of several factors that include, but are not limited to, their lexical preferences, syntactic and sentential choices (e.g., active vs. passive voice, use of detached adjectival clause), discourse structure, narrative style, etc. To perfectly reproduce a given author's style, the language generation model should operate in accordance to all these factors. However, we limit ourselves to replicating an author's *lexical* style, which refers to their writing choices at *word-level*.

Brooke and Hirst (2013a; 2013b) enumerate lexical style elements into **subjective, objective, literary, colloquial, abstract and concrete** categories. An author's choices of words in these categories define their lexical style. For example, Rudyard Kipling, known for classics of children's literature, had a higher tendency to use more concrete words (like, gongs, rockets, torch) unlike Abraham Lincoln, who being a political writer, used more abstract words (like freedom, patriotism) (Verma and Srinivasan, 2019; Syed et al., 2020). Since an author's style is an amalgam of preferences along these dimensions, our goal is to ensure simultaneous alignment to these multi-dimensional lexical preferences of an author.

To quantify a target author's lexical preferences, following Brooke and Hirst (2013b), we compute normalized pointwise mutual information index (PMI) of each vocabulary word with every seed word of 6 categories using their co-occurrences in the Emobank (Buechel and Hahn, 2017) corpus yielding a *raw style score* for each category, for each word in the vocabulary. An author's affinity to a particular style is calculated by the fraction of positive style words in their corpus, yielding a $6-$dimensional vector with each value $\in [0, 1]$. Unlike (Brooke and Hirst, 2013b), we do not consider formality-informality pair due to the imbalance in the number of seed lexicons provided for formality and informality. With a suitable seed lexicon, our approach however can be extended to these two characteristics as well.

3 Proposed Approach

A language model (LM) G models generation of a sequence X as a task of sampling tokens x_0 to x_m. Here, a token x_i is sampled from a probability dis-
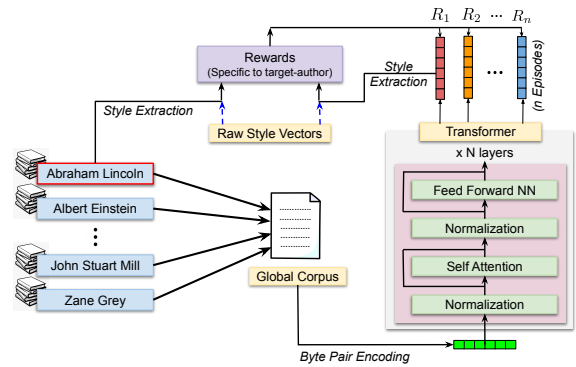

Figure 1: *Overview of the proposed approach. We aggregate a global corpus using a fixed number of paragraphs from each author. The policy generates n episodes for each input in the corpus. Each episode is rewarded with R_i based on its deviation from the target author's lexical style.*

tribution conditioned on the previously generated tokens x_0 to x_{i-1},

$$G(X \mid C) = \prod_{i=0}^{m} G(x_i \mid x_0..x_{i-1}, C) \quad (1)$$

where, $C = c_0, \ldots, c_n$ is the context for generation given by the input prompt to G which provides a sense of broader restriction on generation of X.

Episode Unrolling: An agent (G in our case) in an RL framework, learns a *policy* π to perform a set of *actions* a_i (i.e. generating tokens) resulting in a change of its *states*. The policy's action a_i at a state $S_{i-1} : \{C, x_0, \ldots, x_{i-1}\}$ results in the generation of a token x_i which takes the model to state $S_i : \{C, x_0 \ldots x_i\}$. We refer to the sequence of tokens $E : \{a_0, a_1..., a_t\}$ generated by the LM as it arrives at the terminal state S_t as an *episode*. Instead of relying on a linguistic terminal property (such as end of sentence), we utilize length of the generated sequence as the terminal property of a state. This ensures that the lexical statistics across episodes are consistent while computing the rewards.

For each context C, we unroll N episodes, X_1 to X_N, enabling the policy to *explore* the space better. Unlike the traditional multinomial sampling, we use the *nucleus sampling* (Holtzman et al., 2019) that restricts the sampling to the 'nucleus' of the distribution for generating episodes. By dissuading the choices from the long tail of the distribution, *nucleus sampling* allows the framework to exploit the policy's learning (so far) and hit a balance between *exploration* and *exploitation*.

Rewarding Strategy: Brooke and Hirst (2013b) quantify lexical characteristics using paragraph-level statistics. Extending this, we reward the model with r at the final action a_t of the episode and give 0 reward to the intermediate actions $a_i, i \neq t$ – where t is the terminal step. The overall reward R_i for an *action* at a time step i, where the terminal time step is t and immediate reward received at step i is r_i, is given by discounting the future rewards (Sutton and Barto, 2018) with a factor γ,

$$R_i = r_i + \gamma * r_{i+1} + \gamma^2 * r_{i+2} .... + \gamma^{t-i} * r_t \quad (2)$$

Setting $\gamma = 1$, we distribute the reward uniformly over the entire sequence; all the actions in a particular sequence receive same award irrespective of their position. Since the style is considered at the sequence level, the position of token is irrelevant as long as fluency is maintained.

Defining Rewards: We define the *inclination* of a token (L_{incl}^i) to a target style category to be 1 if its raw style score (from Section 2) is positive and 0 otherwise. Averaging these across all the tokens in an episode yields the lexical alignment score $L_{epi} = \frac{1}{m} \sum_{i=0}^{m} L_{incl}^i$. Since an author's lexical style is an amalgam of several characteristics, we use a Root Mean Squared error against the author's statistics as our aggregated reward for all 6 elements and enabling a continuous adjustment of generation across the target elements. Given the lexical statistic L_{tar} of a target author, the reward $r = \frac{1}{rmse+\epsilon}$, where, $rmse = \frac{\|L_{epi} - L_{target}\|}{\sqrt{6}}$ and ϵ is a factor used to avoid division by zero.

We use our rewards in a modified **self-critical sequence training** setup (Rennie et al., 2017) because this was the most stable framework in our exploration. In our experiments, we have described the other frameworks we explored along with our intuitions on their failures in our problem. A multi-dimensional tuning requires more deviation from an existing policy compared to tuning a single dimension, calling for more exploration, enabled by our episode unrolling. For a context C, the mean reward from N unrolled episodes is used as the baseline reward r_b to reduce the variance during training. Following REINFORCE (Williams, 1992), we minimize the following loss function ,

$$J(\theta) = -(r - r_b) \sum_{i=0}^{m} log(\pi_\theta(x_i \mid x_0, \dots, x_{i-1}, C)) \quad (3)$$

where θ are policy parameters. We scale the rewards for a given context to zero-mean and unit-variance across the N episodes. Following Ranzato et al. (2015), we minimize cross entropy on the tokens from C along with $J(\theta)$ every 5 contexts (*i.e.* $5N$ episodes), so that the model does not deviate and retains its fluency. During this step, our loss is a weighted sum of cross entropy loss and RL loss (empirically set to 0.5 and 1.0).

4 Experiments

We used the $2,857$ books of 142 authors in Gutenberg corpus (Lahiri, 2014) and divided each author's corpus into *train* and *test* sets. We concatenate 50 paragraphs from each author's train corpus and use this for fine-tuning with language modelling loss for each author. To evaluate our model on unseen data, we set aside a subset of 5 paragraphs from each author's test corpus to be used as test context. Having contexts from all authors removes any bias from author-specific contexts.

We compute the average lexical vectors L_{avg} for all authors and retain top 10 authors with maximum deviation from L_{avg} for our experiments: *Charles Darwin, Albert Einstein, Michael Faraday, John Maynard Keynes, Abraham Lincoln, John Locke, John Stuart Mill, Beatrix Potter, Bertrand Russell, and Herbert Spencer.* We use the $117M$ parameters version of the GPT-2 (Radford et al., b) trained on WebText corpus and $50,257$ token invertible byte pair encoding to preserve capitalization and punctuation (Sennrich et al., 2016). The model is a $12-$layers $12-$head Transformer with embedding size of 768. Finetuning[1] GPT-2 on entire Gutenberg corpus yields **GPT-2 (Baseline)**. Finetuning further for one epoch on the target author's corpus with Causal Language Modelling (CLM) loss yields **GPT-2 + FT**. For RL finetuning, we use a batch size 1, 10 episodes for each context, context length 200, episode length 100 and 0.05 as ϵ.

We explored Self-Critical Sequence Training (SCST) (Rennie et al., 2017) and Proximal Policy Optimization (PPO) (Schulman et al., 2017) for our RL setup and chose our episode unrolled SCST because of its stability. Figure 2 shows the mean reward curves averaged over 3 randomly drawn authors; SCST does not help in improvement of rewards, perhaps due to the lesser exploration carried out in the vanilla setup leading to little-to-no-improvement in the lexical style of the generation.

---

[1]Trained on 8 V100 GPUs for 12 hours

Model	Literary		Colloquial		Abstract		Subjective		Concrete		Objective		Overall		Perplexity
	abs	rel	abs	rel	abs	rel	abs	rel	abs	rel	abs	rel	abs	rel	
GPT-2 (Radford et al.)	0.246	1.117	0.081	1.880	0.127	1.708	0.323	0.430	0.270	1.778	0.064	2.413	0.297	0.518	53.82
GPT-2 (Baseline)	0.236	1.255	**0.060**	1.761	**0.101**	1.696	0.293	0.579	0.251	1.700	0.088	2.340	0.283	0.518	**37.54**
GPT-2 + FT	0.242	1.192	0.070	1.757	0.115	1.622	0.298	0.467	0.237	1.700	0.066	2.321	0.283	0.503	38.43
GPT-2 + RL (5K Episodes)	0.174	0.906	0.062	1.395	0.120	1.211	0.222	0.323	0.147	1.295	0.047	1.572	0.221	0.372	38.81
GPT-2 + RL (10K Episodes)	**0.160**	0.869	0.066	**1.328**	0.118	**1.169**	**0.212**	0.323	0.143	1.253	0.046	1.525	**0.213**	0.359	38.57
GPT-2 + FT + RL (5K Episodes)	0.176	0.922	0.067	1.433	0.129	1.231	0.232	0.347	0.154	1.296	**0.041**	1.649	0.226	0.382	38.91
GPT-2 + FT + RL (10K Episodes)	0.162	**0.869**	0.064	1.344	0.126	1.170	0.213	**0.314**	**0.139**	**1.245**	0.048	**1.503**	0.214	**0.358**	38.59

Table 1: Results from our Quantitive Evaluation: 'abs' error for each dimension is calculated as the absolute difference between the target value and obtained value for that dimension while 'rel' error is the absolute deviation of the relative order of the dimension based on the L_1 norm. 'Overall' abs is the RMSE between output and target 6 dimensional vectors while 'Overall' rel is the average rel error across all dimensions. Perplexity indicates the deviation of the model from its fluent generation.
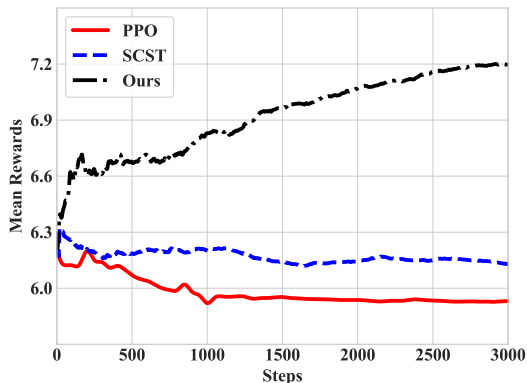

Figure 2: *Mean Reward Curves as averaged over all ten different authors for the different algorithms*

PPO rewards goes down a bit and stays almost constant, perhaps due to the failure of critic. Approximating continuous lexical score is challenging unlike binary positive or negative style – the ones for which PPO has been successful (Ziegler et al., 2019), hence the critic finds it difficult to approximate the value functions. Our episode unrolling based SCST explores enough to improve rewards quickly while frequent cross-entropy objective training ensures that the improvements do not come at cost of fluency. Note that the rewards saturate after a few steps as the nucleus of the distribution gets shifted towards target lexical style.

We calculate the lexical vector L_{seq} for each generated *paragraph* and compute the error against the target author's lexical vector L_{tar}. For dimension-specific error, we take absolute difference between target and generated value for each author and average across all 10 authors. For overall error, we calculate the RMSE between L_{tar} and L_{seq}. We report the perplexity of the model on the contexts in the test set to measure its deviation from its general generation capabilities. We also quantify the alignment in relative ordering of target dimen-

sions using the L_1 norm between L_{seq} and L_{tar}. This evaluates the generation of models on their ability to achieve the target author's relative ordering. In Table 1 along with our absolute error differences along with the L_1 norm as the deviation in the relative order. The overall deviation is computed by adding the L_1 norms across all dimensions and dividing by the maximum possible deviation (which is 18 for a 6 dimensional vector). Our evaluations show the success of our approach in aligning the generation across all lexical dimensions, while CLM fine-tuning does not yield significant improvements over the baseline. We also notice that our model achieves lexical alignment after going through training on just $5k$ episodes evidenced by an insignificant decrease in error for $10k$ episodes. Our approach also ensures that the model has not lost its general generation capabilities which is evident with a marginal drop in perplexity, indicating a minimal trade-off between the lexical alignment and fluency. Infact, the perplexity scores in our case are very significantly lower than the score obtained with an out of domain GPT-2.

In the qualitative example in Table 2, note that our method has infused scientific phrases (e.g., **'observations', 'theory', 'experimental'**) and abstract phrases (e.g., *'self-contained', 'sense of belonging', 'intimate connection'*) for Albert Einstein and Abraham Lincoln, respectively. There is also topic-level alignment with the target authors – a concomitant of meeting the target author's lexical preferences. Fine-tuning GPT-2 on Gutenberg corpus induces literary words like *'allurement'*, *'serviceable'*, *'shop-lenderer'* and *'governess'*; perhaps because Gutenberg corpus contains several literary words not encountered in WebText. Fine-tuning on author-specific corpus (GPT-2 + FT) induces generic stylistic changes, but not necessarily along lexical dimensions. Our approach incorporates lexical preferences most evidently.

Context (originally written by D. H. Lawrence): till now he had a tight, hot place in his chest, with suppressing the pain, and holding himself in. There seemed no air when he breathed. But he walked almost lightly. To his orderly he was at first cold and just and indifferent: he did not fuss over trifles. So that his servant knew practically nothing about him, except just what orders he would give, and how he wanted them obeyed. That was quite simple. Then the change gradually came. He had served the Captain for more than a year, and knew his duty. This he performed easily, as if it were natural to him. The officer and his commands he took for granted, as he took the sun and the rain, and he served as a matter of course. It did not implicate him personally. I have said that the sole effect of my somewhat childish experiment–that of looking down within the tarn had been to deepen the first

---

GPT-2 ((Radford et al., b): singular truth of the Pilgrimage of the Infant. I asked how I was to give my obedience, and, having been told that I must respond by force with violence, I found I had nothing else, less a price and hunger, but the unimaginable change of love and the uniformity of his love that he manifested, in his actions and his conduct...

GPT-2 (Baseline): singular comfort I had felt to-day. The idea of it came back to me, with all its allurement in the old days, over and over again; that it had been a serviceable accomplishment, but a training is apt to all but overwhelm the rest. That, too, was the fact of the plunge, the sudden crisis of this large one. With the active man and the active woman, with the mistress and the husband, with the shop-lenderer and the governess

---

Albert Einstein (GPT-2 + FT): singular feeling in me–or to strengthen the second. I don't exactly know the two; but both of them, even in the early days of my illness, occurred in the true light. I felt sorry, indeed, for my sailor, for his predicament, and for some of the miserable conditions of the household. Perhaps, in so doing, I had really helped myself. It was not at all like a stock with its cartridges. The quartermaster and his company, the Earl of

---

Albert Einstein (Ours) singular **feature** of the **observations** of the **experimental** method of physics. I shall leave the first of these observations (I don't wish to show that the experiments which have been referred to are successful) to the reader who, during a similar period of the subject, considers the relations of natural and experimental **phenomena** to the general **principles** of **inductive theory**.

Abraham Lincoln (GPT-2+FT): singular feeling in me–and to restore the deepness of the conviction that I was a woman; but, in a personal way, I had achieved what was possible only by looking down. The emotion still hovered; and I recollect at that instant a certain feeling–I say feeling–as I struck the buckle. The feeling made me feel that I must be rather an offensive person to be of any use or inconvenience to me. The thought had not, of course, touched

---

Abraham Lincoln (Ours): singular aspect of this *pleasant* and yet *intimate connection* which that early talk had formed with the man I now occupy, and, instead of being *self-contained* and thoroughly historical, it had quite come to be represented by a carefully and strongly expressed recollection, as an aggregation of our *relations* and the signs and symptoms of an *attachment*. Indeed, the meeting with Meade's *sense of belonging*, as an influence on his future, had been rendered infinitely more *interesting* by the clear *knowledge* of a more definite direct relation to myself

---

Table 2: **Qualitative Example.** Generations by different models for the same input prompt. Our model is able to effectively incorporate target author's vocabulary without compromising much on fluency. Here, **Ours** refers to **GPT-2 + FT + RL (10K)**.

5 Conclusion and Future Work

We proposed an approach to incorporate lexical choices of a target author in the generations of a Transformer-based LMs. Our quantitative and qualitative evaluations illustrate that our proposed method is successful in aligning lexical characteristics of generation with target author. We believe that our work can also lead to *rewriting* of the input content tailored to certain characteristics, if we can design additional rewards to retain content. We have not performed a complete human evaluation due to the high-level of required exper-

tise among the annotators for this task, as pointed by Syed et al. (2020). Designing the feedback mechanism for such a human evaluation is nontrivial and has been left as a part of the future work along with designing rewarding schemes to capture other author-specific characteristics (e.g., syntactic choices, discourse structure). Despite the lack of such evaluation, these results are promising and offer a plausible line of research for replication of an author's style.

References

Julian Brooke and Graeme Hirst. 2013a. Hybrid models for lexical acquisition of correlated styles. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 82–90.

Julian Brooke and Graeme Hirst. 2013b. A multi-dimensional Bayesian approach to lexical style. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 673–679, Atlanta, Georgia. Association for Computational Linguistics.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Sven Buechel and Udo Hahn. 2017. EmoBank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 578–585, Valencia, Spain. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Shibamouli Lahiri. 2014. Complexity of Word Collocation Networks: A Preliminary Structural Analysis. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 96–105, Gothenburg, Sweden. Association for Computational Linguistics.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878.

Emilio Parisotto, H Francis Song, Jack W Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant M Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. 2019. Stabilizing transformers for reinforcement learning. *arXiv preprint arXiv:1910.06764*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. a. Improving language understanding by generative pre-training.

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

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

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

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*.

Bakhtiyar Syed, Gaurav Verma, Balaji Vasan Srinivasan, Anandhavelu Natarajan, and Vasudeva Varma. 2020. Adapting language models for non-parallel author-stylized rewriting. In *AAAI*, pages 9008–9015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Gaurav Verma and Balaji Vasan Srinivasan. 2019. A lexical, syntactic, and semantic perspective for understanding style in text. *arXiv preprint arXiv:1909.08349*.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2019. Harnessing pre-trained neural networks with rules for formality 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 3564–3569.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.