

# Personalized Text Generation with Fine-Grained Linguistic Control

Bashar Alhafni<sup>1</sup>, Vivek Kulkarni<sup>2</sup>, Dhruv Kumar<sup>2</sup>, Vipul Raheja<sup>2</sup>

<sup>1</sup>New York University Abu Dhabi

<sup>2</sup>Grammarly

alhafni@nyu.edu, firstname.lastname@grammarly.com

## Abstract

As the text generation capabilities of large language models become increasingly prominent, recent studies have focused on controlling particular aspects of the generated text to make it more personalized. However, most research on controllable text generation focuses on controlling the content or modeling specific high-level/coarse-grained attributes that reflect authors' writing styles, such as formality, domain, or sentiment. In this paper, we focus on controlling fine-grained attributes spanning multiple linguistic dimensions, such as lexical and syntactic attributes. We introduce a novel benchmark to train generative models and evaluate their ability to generate personalized text based on multiple fine-grained linguistic attributes. We systematically investigate the performance of various large language models on our benchmark and draw insights from the factors that impact their performance. We make our code, data, and pretrained models publicly available.<sup>1</sup>

## 1 Introduction

With the evolution of large language models (LLMs), applications involving text generation have become increasingly prominent (Kaddour et al., 2023). In recent years, there has been a flurry of writing and conversational assistants that harness the power of LLMs at each stage of the writing lifecycle (Raheja et al., 2023; Gómez-Rodríguez and Williams, 2023). At the same time, there has been a significant emphasis on research that enables the generation of personalized text – text that reflects the author's style and content more accurately. For instance, Figure 1 presents two movie reviews that are conveyed in different ways. The reviews differ linguistically in terms of the number of tokens and the distributions of adverbs, adjectives, and nouns.

Several works explore the personalization of various NLP tasks such as Lexical Simplification (Lee

<sup>1</sup><https://github.com/balhafni/personalized-gen>

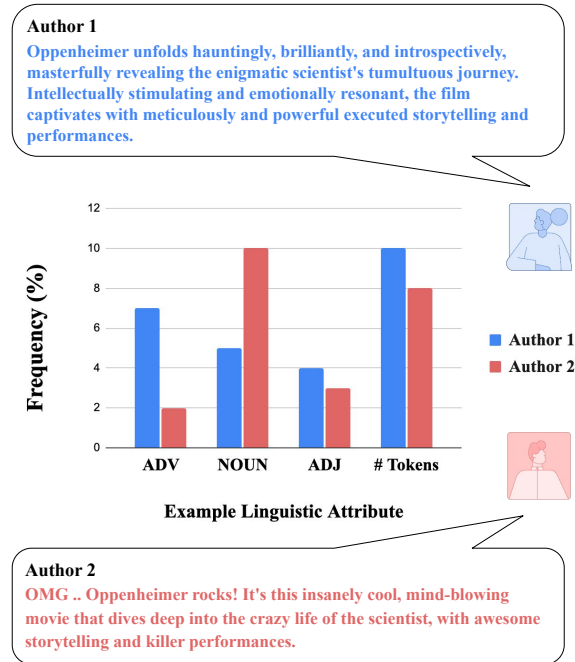


Figure 1: An example highlighting the stylistic linguistic differences between two reviews about the *Oppenheimer* movie. The bar chart indicates the frequencies of the adverbs (ADV), nouns (NOUN), adjectives (ADJ), and the number of tokens (# Tokens). The blue and red bars represent the frequencies based on the reviews written by Author 1 and Author 2, respectively.

and Yeung, 2018; Bingel et al., 2018), Dialogue Modeling (Zhang et al., 2018; Mazaré et al., 2018; Madotto et al., 2019; Ma et al., 2021; Zhong et al., 2022), Machine Translation (Rabinovich et al., 2017), Grammatical Error Correction (Nadejde and Tetreault, 2019), Summarization (Takatsu et al., 2021), and Title Generation (Salemi et al., 2023) among others. However, a key question remains unanswered: How effective are LLMs at personalizing the text they generate? While undoubtedly, there has been a lot of work on controllable text generation (e.g., using either control codes (Keskar et al., 2019) or adapters (Hu et al., 2023)), many of these works model the author's style either by using

Dataset	# Docs	# Authors
<b>Blogs</b>	24,913	140
<b>IMDb62</b>	38,693	62
<b>Amazon</b>	42,542	49
	106,148	251

Table 1: Data statistics of our benchmark.

a unique control code or a dense embedding and do not really evaluate how effective these models are at generating text that reflects a fine-grained set of stylistic features. For instance, authors may exhibit variations in the distribution of discourse markers in their writing. To what extent can models reflect fine-grained stylistic differences and account for them when generating text? We seek to advance this line of research further by introducing a novel benchmark that evaluates the ability of generative models to accurately reflect fine-grained stylistic attributes spanning multiple linguistic dimensions. Our benchmark enables us to gain valuable insights into how model-generated text responds to deliberate variations in style across different dimensions, providing a thorough understanding of the model’s sensitivity. In addition to developing a benchmark, we systematically explore and analyze the performance of LLMs on this benchmark and draw key insights into the efficacy of various models and what factors drive performance. Moreover, we also propose improvements to baseline models that can significantly improve performance on this newly introduced benchmark for multi-attribute personalized style transfer. In a nutshell, our key contributions are as follows:

- **Benchmark** We introduce a benchmark for the task of personalized text generation with fine-grained linguistic control.
- **Linguistic Feature Analysis** We perform an extensive characterization of the authors’ styles in terms of linguistic features spanning across multiple linguistic dimensions including lexical, rhetoric, and syntactic features.
- **Improved Models** We evaluate the effectiveness of various baseline models on this task and provide insights on the performance of current LLMs.

## 2 Benchmark Construction

In this section, we discuss the construction process of our benchmark in terms of datasets, linguistic attributes, and evaluation metrics.

### 2.1 Task Definition

Given a set of stylistic attributes encapsulating the writing style of a particular author and an input textual prompt, the goal is to generate text that adheres to the provided attributes. More formally, we define a set of  $n$  attributes  $\{a_1, a_2, \dots, a_n\}$ , where each attribute  $a_i$  has a set of possible values  $\{v_1, v_2, \dots, v_k\}$ . Given a possible assignment to these values, our goal is to generate text that conforms to these values.

### 2.2 Datasets

To construct our benchmark, we draw on multiple sources of author-grounded data spanning multiple domains. We chose samples from various datasets while ensuring that the number of authors we intend to model remains reasonable with respect to the models’ capacity and the need to avoid excessive sparsity. We derive our benchmark from the following datasets:

**Blogs Authorship Corpus** consists of 681.3K blogs written by 19,320 authors on *blogger.com* in 2004 (Schler et al., 2006). The dataset includes information about the authors in terms of their age, gender, industry, and astrological signs. Each author is associated with one of 40 industrial categories. We select the largest five industrial categories in terms of the number of blogs to include in our benchmark. Namely, we focus on the following categories: Technology, Education, Arts, Internet, and Communication-media. Out of this subset, we select blogs that have at least 50 words and authors who wrote at least 100 blogs each. In the end, we ended up with 24,913 blogs written by 140 authors.

**IMDb62** contains 62,000 movie reviews written by 62 prolific users of the Internet Movie Database (IMDb) where each user wrote 1,000 reviews (Seroussi et al., 2014). Out of all reviews, we only keep the ones with at least 50 words. In total, we ended up with 38,693 reviews written by 62 authors.

**Amazon Reviews** comprises 233.1M product reviews on Amazon published between 1996 and 2018 (Ni et al., 2019). We use the publicly available 5-core version of the corpus which consists of

Features Type	Attribute
Lexical	# Tokens
	# Sentences
	Readability Score
Morpho-Syntax	# POS Tags
	# Syntax Relations
Discourse	# Rhetorical Relations

Table 2: Summary of the linguistic attributes we model.

157.3M reviews written by 9.8M users where each user wrote at least five reviews. Out of this rich set, we selected users who wrote at least 2,800 reviews with each review containing at least 50 words. This resulted in 42,542 reviews written by 49 authors.

Table 1 presents a summary of the datasets we use in our benchmark. Putting all the selected samples together leaves us with 106,148 textual examples written by 251 authors to constitute our benchmark. For each author, we randomly sample examples to create train (80%), development (10%), and test (10%) splits. This ensures that all 251 authors modeled during training will also have examples in the development and test sets. Altogether, we have a total of 84,824 examples for training (Train), 10,506 examples for development (Dev), and 10,818 examples for testing (Test).

### 2.3 Controllable Linguistic Attributes

To operationalize an author’s linguistic style, we extract various linguistic attributes from the author’s text. In particular, we represent each author’s linguistic style as a vector of attributes – linguistic attributes that capture patterns in lexical usage, morpho-syntactic information, and discourse markers. More specifically, for each author  $A$ , we associate a feature vector  $\phi(A)$  that represents the author’s linguistic style. Each dimension of  $\phi(A)$  corresponds to attributes from the following feature families as summarized in Table 2:

**Lexical Features** We model the lexical usage for each author by using three lexical features: *number of tokens*, *number of sentences*, and the readability level for each text written by a specific author. We use spaCy (Honnibal and Montani, 2017) to tokenize and parse the text to obtain the number of tokens and the number of sentences. We measure the *readability* of the text (i.e., the ease with which a reader can understand the text) using the FKGL score (Flesch, 1948).

**Morpho-Syntactic Features** We consider the frequencies of the part-of-speech (POS) tags and the dependency relations to capture the morpho-syntactic information for each author’s text. For the POS tags, we consider 14 core tags as defined in the Universal POS tagset, which are internally subdivided into open-class words (i.e., adjective, adverb, interjection, noun, proper noun, verb), closed-class words (i.e., adposition, auxiliary, coordinating conjunction, determiner, numeral, particle, pronoun, subordinating conjunction), and the class of ‘other’ which includes punctuation and symbols. For the dependency relations, we use 32 dependency relations as defined by ClearNLP.<sup>2</sup> We use the POS tagger and the dependency parser available in spaCy to extract the morpho-syntactic information.

**Discourse Features** Discourse coherence, which refers to how well a text or speech is organized to convey information, captures important stylistic aspects of one’s writings. Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) is one of the most influential approaches for document-level discourse analysis. It can represent a document’s discourse structure using a hierarchical tree in which nodes are recursively linked with rhetorical relations and labeled with nucleus or satellite tags to depict the importance of the child nodes in a relation. In our work, we consider the frequency of the RST relations for each author’s text to capture discourse coherence. We use the publicly available RST parser<sup>3</sup> introduced by Heilman and Sagae (2015) to obtain RST relations. We model 3 RST relations in total.

Once we extract the above linguistic features for each textual example written by a particular author, we average the values of the features for all examples to obtain the author’s vector. We consider the above features since they succinctly capture the author’s linguistic style in an interpretable manner (i.e., each vector dimension has a clear interpretation), and further, such vector representation serves as a lever for controlled modification of specific linguistic attributes. Table 4 in Appendix B provides the complete list of all the attributes we consider for this benchmark.

<sup>2</sup><https://github.com/clir/clearnlp-guidelines>

<sup>3</sup><https://github.com/EducationalTestingService/rstfinder>

## 2.4 Linguistic Attributes Representation

After extracting the linguistic attributes, each attribute will be represented by a continuous value indicating the average frequency of that attribute in texts written by a specific author. To reduce sparsity and enhance the models’ abilities to learn contextual representations for each attribute value, we discretize the values for all attributes. To do so, we group the values for each attribute into deciles based on the training data. After that, each value is represented by a specific discrete bin corresponding to the range it falls within. Representing each attribute using discrete bins reduces the vocabulary size the model has to learn to reflect the specified linguistic attributes in the generated text.

## 2.5 Evaluation Metrics

We evaluate models on their effectiveness of generating (or rewriting) input text that corresponds to a given style (as operationalized by the attributes discussed in §2.3). To quantify the effectiveness of various models at responding correctly to controlled changes in style, we report the success rate as measured by the fraction of times the model generated text whose style vector corresponds to the specific style provided. We compute the success rate for each attribute as well as the relative improvement over a random baseline (i.e., choosing an attribute bin at random). We report the **mean success rate** and the **median of the relative improvements** across all attributes as a summary of model performance. Naturally, the mean success rate ranges from 0 to 100. However, the relative improvements depend on the number of bins for each attribute; generally, it ranges from  $-100$  to  $(\# \text{ of bins}) \times 100$  for each attribute. Moreover, we use the distribution of grammatical errors to compare the **fluency** of the generated text against the gold data for each author. We use an in-house grammatical error detection model to get the distributions of grammatical errors. For each author, we compare the error distributions by running a t-test (Yuen, 1974) and report the fraction of authors whose distributions are not statistically significant ( $p > 0.05$ ). Hence, the fluency scores will range from 0 to 100.

## 3 Models

To evaluate our proposed benchmark and given the ubiquitous adoption of causal language models (CLMs) for various NLP tasks recently, we con-

sider the models from the **Pythia** Scaling Suite (Biderman et al., 2023). The suite comprises eight models with sizes ranging from 70M to 1B parameters. All the models are based on the GPT-Neo architecture (Black et al., 2021), which is an open-source replication of GPT-3 (Brown et al., 2020). The models were pretrained using the Pile dataset (Gao et al., 2020), an 825GB English dataset containing texts from 22 diverse sources, roughly broken down into five categories: academic writing, internet, prose, dialogue, and miscellaneous. All the Pythia models were pretrained on the same data. In this work, we use the 70M, 160M, and 1B Pythia models.

### 3.1 Baselines

For our baselines, we fine-tune the Pythia models on our benchmark dataset without feeding any attributes to the models. During inference, we feed the first sentence of every textual example in the Dev or Test sets to get the generated predictions. The intuition behind this baseline is to investigate if the models are able to pick up on the linguistic attributes from the text itself. We also consider GPT-3.5 (ChatGPT)<sup>4</sup>, given the good performance it has shown on various NLP benchmarks. We use the gpt-3.5-turbo0613 model from the OpenAI API.<sup>5</sup> We describe the training settings and the model’s hyperparameters in Appendix A. The prompts we use for the GPT-3.5 experiments are also presented in the same appendix.

### 3.2 Multi-Attribute Controlled Models

Given an input sentence along with an author linguistic vector, the goal of the models is to generate text that reflects the provided linguistic attributes (§2.1). To incorporate features context into the models during training, we treat each author’s linguistic attribute vector as a sequence of tokens and prepend it to the textual examples written by that particular author. We extend the input embedding layer of all models by adding the attributes and their values as additional tokens to the models. This enables the model to learn the semantics of the attributes. During inference, we prepend the author’s linguistic attribute vector to the first sentence of every textual example in the Dev or Test sets to get the generated predictions. We refer to these models as **Prefix** throughout the paper.

<sup>4</sup><https://openai.com/blog/chatgpt>

<sup>5</sup><https://api.openai.com>

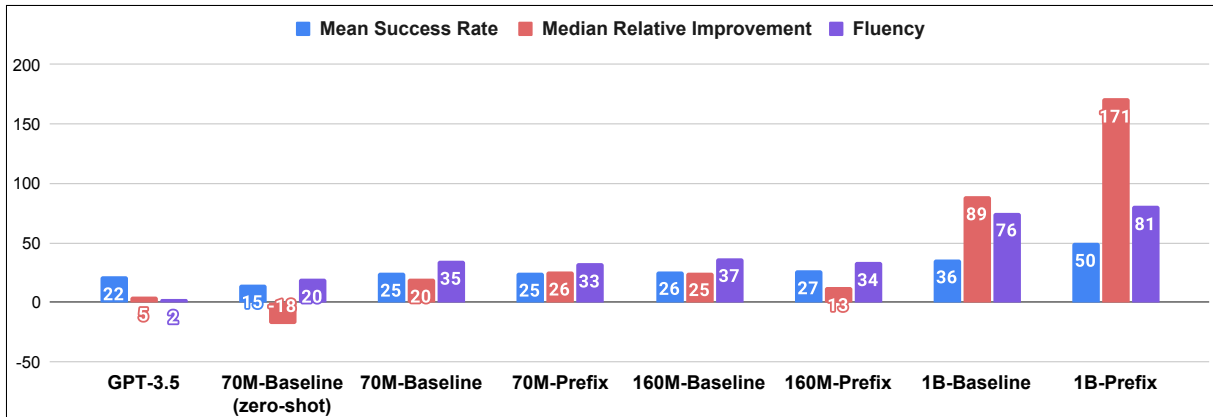


Figure 2: Results on the Test set of our benchmark using GPT-3.5 and the 70M, 160M, and 1B Pythia models. The Mean Success Rate and Fluency vary from 0 to 100. The Median Relative Improvement can vary from -100 to  $(\# \text{ of bins}) \times 100$  for each attribute. Details on the evaluation metrics are provided in §2.5.

## 4 Experiments and Results

In this section, we present the results of various models along with an investigation into the model’s sensitivity to the attributes. We also explore how the training sample size influences the model’s performance.

### 4.1 Overall Results

The results of the Test set are presented in Figure 2.

**Baselines** Although GPT-3.5 achieves a mean success rate of 22, it did not do well in terms of the median relative improvement and fluency. It is worth noting that GPT-3.5’s low score in fluency reflects the difference between the distribution of grammatical errors in the output of GPT-3.5 and the gold data. A sample of GPT-3.5 outputs reveals that the generated text is acceptable but contains different error proportions compared to the gold data. When it comes to the Pythia models, the untrained 70M Pythia baseline model (zero-shot) has the lowest scores in terms of the mean success rate and the median relative improvement. We observe an interesting pattern for the trained baselines: the larger the models, the better the results are across all metrics, with the 1B Pythia model being the best performer among the baselines.

**Multi-Attribute Controlled Models** The 70M and 1B Pythia *Prefix* models achieve a better performance than their baseline counterparts. This highlights that, generally, the learned representations of the linguistic attributes provide the models with better control. Overall, the 1B Pythia *Prefix* model was the best performer across all metrics. In terms of attribute-specific performance, we present

the results of the 1B baseline and the 1B *Prefix* models in Appendix C. Across almost all attributes, the *Prefix* model was the better performer in terms of the success rate and relative improvements.

### 4.2 Attribute Sensitivity Analysis

We investigate the model’s sensitivity to the attributes it has been conditioned on. To do so, we sample examples randomly from the Dev set for each author with at least 1000 data examples in Train. Then, for each attribute in each example, we change the attribute’s respective bin according to the possible bins it could have while keeping all other attributes the same. This is done in a controlled setup where each attribute would be assigned to a range of possible bins that reflect an increase or a decrease from its gold value. During inference, we prepend the updated author’s attribute vector to the first sentence of the textual example written by that particular author and perform greedy decoding to generate the model outputs. It is worth noting that we do this generation step for each attribute change. This resulted in 14,206 generated examples.

To measure the sensitivity of the model to the changes in the attributes, we compute the attribute-specific success rate by checking if the attribute value in the generated text moved in the right direction in terms of an increase or a decrease based on its newly assigned bin. We report the average success rate for each attribute across all generated examples. Figure 3 shows model performance as a function of attribute value displacement for 12 representative attributes. From the figure, we can make the following two observations: (a) First, we

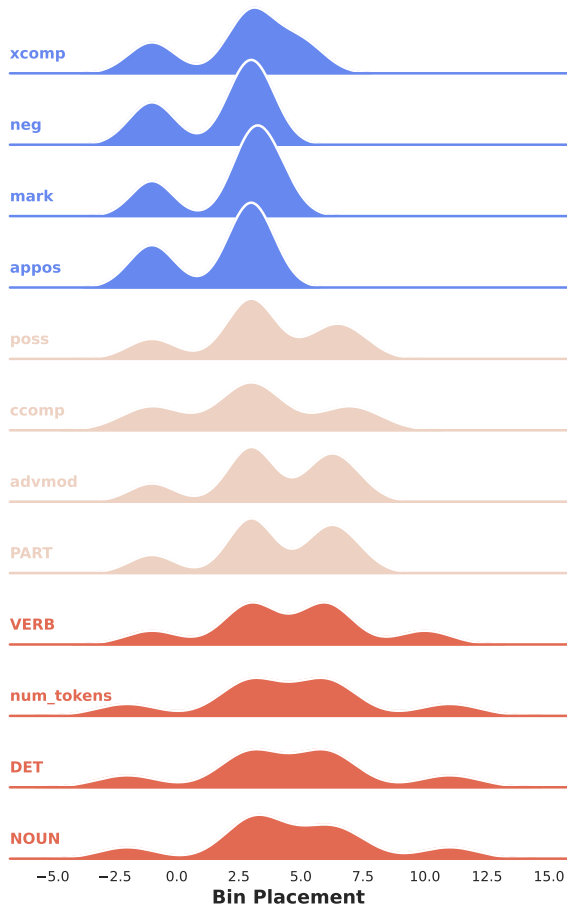


Figure 3: A collection of three different clusters of attributes, varying by their Bin Placement (x-axis) and their Mean Success Rate (%), (y-axis). Bin Placement refers to the number of bins skipped when assigning a new bin to each attribute (in sorted order), while the Mean Success Rate is calculated as explained in §2.5. We observe that the attributes display three distinct sensitivity patterns.

observe that the attributes display three distinct sensitivity patterns. (b) Second, performance responses tend to be bi-modal or even tri-modal, with performance peaking at certain values and degrading as one falls off the extremes. We believe a deeper investigation that could explain the emergence of the observed multi-modal performance is an interesting one, which we leave to future work.

Table 3 presents examples of model-generated outputs. The examples illustrate how the outputs of different models vary due to changing specific attributes for a fixed input sentence, highlighting the models’ attribute sensitivity.

### 4.3 Training Sample Efficacy

To gain insights into the scaling behavior of personalized LLMs in terms of data, we examine the

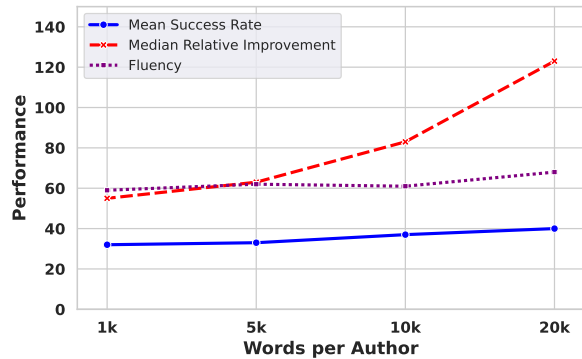


Figure 4: The performance of the 1B Pythia Prefix model on the Dev set at different data sizes in terms of the mean success rate, the median relative improvement, and fluency.

impact of the number of training samples per author on model performance by limiting the training data per author to 1k, 5k, 10k, and 20k words. It is worth noting that we repeat the discretization process described in §2.4 for all attributes based on the scaled training data before training and evaluating the models. Figure 4 presents the performance of the 1B Pythia Prefix model on the Dev set after being trained on different data sizes. As expected, we observe performance decrease across all three metrics as we reduce the number of words per author. We see the most degradation in performance in the median relative improvement. This is because it reports the median of the relative improvement to a random baseline and, therefore, has the highest sensitivity to the model performance. We also observe a degradation in the mean success rate; however, it is gradual as this metric is dependent on the bins, which makes it less sensitive to change. For fluency, we initially see a slight decrease in performance as we decrease the words per author to 10k, followed by a slight decrease for lower data proportions. For all three metrics, we see the biggest decrease in performance as we lower the training instances per author beyond 20k words.

## 5 Related Work

### 5.1 Personalized Language Modeling

Numerous works have explored learning user embeddings as a way to capture syntactic and semantic properties of the language of individuals (Hovy, 2015; Ould Amer et al., 2016; Zeng et al., 2017; Welch et al., 2020b,a; Rocca and Yarkoni, 2022). These are then typically used to provide personalized inductive priors to downstream tasks. Recent

Model	Attributes	Output
Original	VERB: 8-11, num_sents: 5-6, ...	Today’s lunch <u>was</u> a <u>layered</u> entree, <u>consisting</u> of, from bottom to top, corn chips, <u>shredded</u> lettuce and popcorn chicken. What <u>was</u> this culinary delight? A popcorn chicken salad, of course. I <u>have</u> yet to <u>figure</u> out exactly why there <u>were</u> corn chips <u>involved</u> . And why the salad bar then <u>had</u> taco-plate fixin’s, rather than salad fixin’s.
GPT-3.5	VERB: 8-11, num_sents: 5-6, ...	Today’s lunch <u>was</u> a <u>layered</u> entree, <u>consisting</u> of, from bottom to top, corn chips, <u>shredded</u> lettuce and popcorn chicken. The dish <u>was</u> a perfect combination of crunchy and savory flavors, with the corn chips <u>providing</u> a satisfying crunch and the popcorn chicken <u>adding</u> a deliciously salty taste. The shredded lettuce <u>added</u> a refreshing touch to the dish, <u>making</u> it a perfect meal for a hot summer day. Overall, the dish <u>was</u> a great choice for lunch and I would definitely <u>recommend</u> it to anyone <u>looking</u> for a tasty and satisfying meal.
	<b>VERB: 27-32,</b> num_sents: 5-6, ...	Today’s lunch <u>was</u> a <u>layered</u> entree, <u>consisting</u> of, from bottom to top, corn chips, <u>shredded</u> lettuce and popcorn chicken. The chicken <u>was</u> <u>seasoned</u> and <u>layered</u> with chips. I <u>had</u> one bag of popcorn chicken and the other <u>had</u> regular chicken breast. It <u>was</u> quite tasty and guilt-free. Of course, there <u>was</u> the inevitable over-eating comment from my hubby about how I don’t <u>like</u> eating chicken when there’s so much stuff in it now. He <u>said</u> he wouldn’t <u>eat</u> another chicken entree today. Shh, he’s <u>talking</u> shop. (He’s not wrong!)
1B-Prefix	VERB: 8-11, <b>num_sents: &gt;=21,</b> ...	Today’s lunch <u>was</u> a <u>layered</u> entree, <u>consisting</u> of, from bottom to top, corn chips, <u>shredded</u> lettuce and popcorn chicken. The chicken, which I’ve <u>had</u> in the past, <u>was</u> really great – very moist and flavorful. There <u>were</u> also some other good flavors <u>going</u> on, but the entrees <u>were</u> strong enough to <u>keep</u> me from <u>feeling</u> overwhelmed. I <u>had</u> about a tablespoon of corn and a cup of popcorn, and it <u>was</u> just right. Both flavors <u>were</u> good, though the lettuce <u>was</u> overbearing and I <u>was</u> having a hard time <u>telling</u> what <u>was</u> real and what wasn’t. Sharky’s <u>has</u> <u>convinced</u> me that it’s not always the food I <u>know</u> in my mouth. That there’s more to be <u>learned</u> about what we <u>eat</u> .

Table 3: Illustrative examples of how the generated outputs of different models vary due to changing specific attributes for a fixed input sentence. We present generated outputs using GPT-3.5 and the 1B Pythia Prefix model (our best model). Text in purple indicates the input prompt. Attributes in bold (i.e., VERB or num\_sents) indicate a change in the bin for this attribute. Underlined and bold words refer to verbs.

studies (Miresghallah et al., 2022; Zhong et al., 2021; Oba et al., 2023) have explored leveraging techniques such as adapters for adapting LLMs for various personalized NLP and NLG tasks.

Our work is closely related to King and Cook (2020), who analyzed various methods, such as interpolation, fine-tuning, and priming language models for the personalization of general-purpose language models. They also analyzed model adaptation for models trained on users with similar demographics. This was inspired by Lynn et al. (2017), who showed that demographic factors could help model a variety of classification tasks, and found that personalized models perform better than those adapted from similar demographics. Shao et al. (2020) have also explored language models for personalization but focused on handling OOV tokens.

Finally, our work is also related to Salemi et al. (2023) in that they also propose a benchmark for training and evaluating language models for per-

sonalized text generation tasks. However, their work is focused on specific tasks such as headline and email subject generation, whereas our work is focused on personalized language modeling.

## 5.2 Multi-Attribute Controlled Generation

Controlling text generation using control codes is an active area of research where, given a code  $c$ , the learning problem is formulated to generate text  $x$  by calculating the probability of  $p(x|c)$ . Numerous works have been proposed to train attribute-conditional models by fine-tuning pretrained models with attribute-specific corpora or training conditional generative networks (Keskar et al., 2019; Dathathri et al., 2020; Lample et al., 2019; Logeswaran et al., 2018; Krause et al., 2021; Russo et al., 2020; Yu et al., 2021; Kulkarni et al., 2021). Some of the noteworthy lines of research have proposed techniques such as Prefix-Tuning (Qian et al., 2022), Adapters (Houlsby et al., 2019), and Prompt-

tuning (Yang et al., 2023; Chen et al., 2023) to approach the problem. However, most of the prior works are limited by the number of attributes, as well as the amount of variation in those attributes that they are able to model, typically focusing on high-level or coarse-grained attributes such as sentiment or domain.

A major advantage of our approach is its ability to introduce a large number of discrete or continuous attributes, with a fair amount of variation in each of them. It focuses on a much greater number of fine-grained linguistic attributes that can be easily inferred from large amounts of data, irrespective of document lengths, domains, or other higher-order characteristics, which makes it better adaptable to a diverse range of applications. This adaptability is particularly beneficial in scenarios where the data varies significantly in style, structure, or content. Our approach leverages the inherent capabilities of pretrained models, enhancing them with attribute-specific features. This enables the model to not only understand but also generate text that accurately reflects the desired attributes, be they structural, syntactic, or stylistic. Moreover, our work allows for the dynamic adjustment of attributes in real-time, offering a level of control and precision that is not typically feasible with traditional models. As a result, users can tailor the output to meet specific requirements without the need for extensive retraining or manual intervention. This flexibility is highly essential and desirable for personalized text generation.

## 6 Conclusion and Future Work

We presented a novel benchmark to train and evaluate the ability of LLMs to generate personalized text based on multiple linguistic attributes. While existing research predominantly concentrates on content control or modeling specific writing style elements, our work stands out by focusing on nuanced stylistic attributes across diverse linguistic dimensions. We systematically investigate the performance of various LLMs and draw insights from the factors that drive such performance. We make our code, data, and pretrained models publicly available to encourage research on personalization and controlling LLMs. In future work, we plan to explore the use of other pretrained LLMs, and to extend the linguistic features that are covered in our benchmark.

## Limitations

In this work, while we present a benchmark to study the various linguistic phenomena in text, our study is limited to English. It would be interesting to observe how the patterns we observe for English extend to other languages. Although we investigate the performance of LLMs, we restrict ourselves to CLMs (GPT-3.5 and Pythia) for both the zero-shot and fine-tuning experiments. It would be interesting to study if this effect is replicated in other models, such as encoder-decoder-based models. We propose three automated metrics for our benchmark and observe interesting findings on models' capabilities to handle the fine-grained attributes. However, even though we do manually look at generated examples to ensure that our metrics make sense, a large-scale human study towards looking at generated outputs would help us study and better understand the current models' performance and capabilities. Finally, while building our dataset, we draw on multiple sources of author-grounded data that spans multiple domains; we could further extend this to more domains and authors.

## Acknowledgments

We thank our colleagues at Grammarly for the helpful discussions and constructive feedback.

## References

- Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. [Pythia: A suite for analyzing large language models across training and scaling](#).
- Joachim Bingel, Gustavo Paetzold, and Anders Søgaard. 2018. [Lexi: A tool for adaptive, personalized text simplification](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 245–258, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. [GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow](#). If you use this software, please cite it using these metadata.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child,



- Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#).
- Derek Chen, Celine Lee, Yunan Lu, Domenic Rosati, and Zhou Yu. 2023. [Mixture of soft prompts for controllable data generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14815–14833, Singapore. Association for Computational Linguistics.
- Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. [Plug and play language models: A simple approach to controlled text generation](#). In *International Conference on Learning Representations*.
- Rudolph Flesch. 1948. [A new readability yardstick](#). *Journal of Applied Psychology*, 32(3):p221 – 233.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.
- Carlos Gómez-Rodríguez and Paul Williams. 2023. [A confederacy of models: a comprehensive evaluation of LLMs on creative writing](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14504–14528, Singapore. Association for Computational Linguistics.
- Michael Heilman and Kenji Sagae. 2015. [Fast rhetorical structure theory discourse parsing](#).
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. [Parameter-efficient transfer learning for NLP](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.
- Dirk Hovy. 2015. [Demographic factors improve classification performance](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 752–762, Beijing, China. Association for Computational Linguistics.
- Zhiqiang Hu, Roy Ka-Wei Lee, and Nancy F. Chen. 2023. [Adapter-tst: A parameter efficient method for multiple-attribute text style transfer](#).
- Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.
- Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL - A Conditional Transformer Language Model for Controllable Generation. *arXiv preprint arXiv:1909.05858*.
- Milton King and Paul Cook. 2020. [Evaluating approaches to personalizing language models](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2461–2469, Marseille, France. European Language Resources Association.
- Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. [GeDi: Generative discriminator guided sequence generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Vivek Kulkarni, Shubhanshu Mishra, and Aria Haghighi. 2021. Lmsoc: An approach for socially sensitive pretraining. *arXiv preprint arXiv:2110.10319*.
- Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc’Aurelio Ranzato, and Y-Lan Boureau. 2019. [Multiple-attribute text rewriting](#). In *International Conference on Learning Representations*.
- John Lee and Chak Yan Yeung. 2018. [Personalizing lexical simplification](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 224–232, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS’18*, page 5108–5118, Red Hook, NY, USA. Curran Associates Inc.
- Veronica Lynn, Youngseo Son, Vivek Kulkarni, Niranjan Balasubramanian, and H. Andrew Schwartz. 2017. [Human centered NLP with user-factor adaptation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1146–1155, Copenhagen, Denmark. Association for Computational Linguistics.
- Zhengyi Ma, Zhicheng Dou, Yutao Zhu, Hanxun Zhong, and Ji-Rong Wen. 2021. [One chatbot per person: Creating personalized chatbots based on implicit user profiles](#). In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’21*, page 555–564, New York, NY, USA. Association for Computing Machinery.

- Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. 2019. [Personalizing dialogue agents via meta-learning](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5459, Florence, Italy. Association for Computational Linguistics.
- William Mann and Sandra Thompson. 1988. [Rhetorical structure theory: Toward a functional theory of text organization](#). *Text*, 8:243–281.
- Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. [Training millions of personalized dialogue agents](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779, Brussels, Belgium. Association for Computational Linguistics.
- Fatemehsadat Mireshghallah, Vaishnavi Shrivastava, Milad Shokouhi, Taylor Berg-Kirkpatrick, Robert Sim, and Dimitrios Dimitriadis. 2022. [UserIdentifier: Implicit user representations for simple and effective personalized sentiment analysis](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3449–3456, Seattle, United States. Association for Computational Linguistics.
- Maria Nadejde and Joel Tetreault. 2019. [Personalizing grammatical error correction: Adaptation to proficiency level and L1](#). In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 27–33, Hong Kong, China. Association for Computational Linguistics.
- Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. [Justifying recommendations using distantly-labeled reviews and fine-grained aspects](#). 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 188–197, Hong Kong, China. Association for Computational Linguistics.
- Daisuke Oba, Naoki Yoshinaga, and Masashi Toyoda. 2023. [Perplm: Personalized fine-tuning of pre-trained language models via writer-specific intermediate learning and prompts](#).
- Nawal Ould Amer, Philippe Mulhem, and Mathias Géry. 2016. [Toward Word Embedding for Personalized Information Retrieval](#). In *Neu-IR: The SIGIR 2016 Workshop on Neural Information Retrieval*, volume abs/1606.06991, Pisa, Italy.
- Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. [Controllable natural language generation with contrastive prefixes](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2912–2924, Dublin, Ireland. Association for Computational Linguistics.
- Ella Rabinovich, Raj Nath Patel, Shachar Mirkin, Lucia Specia, and Shuly Wintner. 2017. [Personalized machine translation: Preserving original author traits](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1074–1084, Valencia, Spain. Association for Computational Linguistics.
- Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. [CoEdIT: Text editing by task-specific instruction tuning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5274–5291, Singapore. Association for Computational Linguistics.
- Roberta Rocca and Tal Yarkoni. 2022. [Language as a fingerprint: Self-supervised learning of user encodings using transformers](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1701–1714, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Giuseppe Russo, Nora Hollenstein, Claudiu Cristian Musat, and Ce Zhang. 2020. [Control, generate, augment: A scalable framework for multi-attribute text generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 351–366, Online. Association for Computational Linguistics.
- Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. [Lamp: When large language models meet personalization](#).
- Jonathan Schler, Moshe Koppel, Shlomo Engelson Argamon, and James W. Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*.
- Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2014. [Authorship attribution with topic models](#). *Computational Linguistics*, 40(2):269–310.
- Liqun Shao, Sahitya Mantravadi, Tom Manzini, Alejandro Buendia, Manon Knoertzer, Soundar Srinivasan, and Chris Quirk. 2020. [Examination and extension of strategies for improving personalized language modeling via interpolation](#). In *Proceedings of the First Workshop on Natural Language Interfaces*, pages 20–26, Online. Association for Computational Linguistics.
- Hiroaki Takatsu, Takahiro Kashikawa, Koichi Kimura, Ryota Ando, and Yoichi Matsuyama. 2021. [Personalized extractive summarization using an ising machine towards real-time generation of efficient and coherent dialogue scenarios](#). In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 16–29, Online. Association for Computational Linguistics.
- Charles Welch, Jonathan K. Kummerfeld, Verónica Pérez-Rosas, and Rada Mihalcea. 2020a. [Compositional demographic word embeddings](#). In *Proceedings of the 2020 Conference on Empirical Methods*

- in *Natural Language Processing (EMNLP)*, pages 4076–4089, Online. Association for Computational Linguistics.
- Charles Welch, Jonathan K. Kummerfeld, Verónica Pérez-Rosas, and Rada Mihalcea. 2020b. [Exploring the value of personalized word embeddings](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6856–6862, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Mingfeng Xue, Boxing Chen, and Jun Xie. 2023. [Tailor: A soft-prompt-based approach to attribute-based controlled text generation](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 410–427, Toronto, Canada. Association for Computational Linguistics.
- Dian Yu, Zhou Yu, and Kenji Sagae. 2021. [Attribute alignment: Controlling text generation from pre-trained language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2251–2268, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Karen K. Yuen. 1974. [The two-sample trimmed t for unequal population variances](#). *Biometrika*, 61(1):165–170.
- Ziqian Zeng, Yichun Yin, Yangqiu Song, and Ming Zhang. 2017. [Socialized word embeddings](#). In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3915–3921.
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. [Personalizing dialogue agents: I have a dog, do you have pets too?](#) In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.
- Hanxun Zhong, Zhicheng Dou, Yutao Zhu, Hongjin Qian, and Ji-Rong Wen. 2022. [Less is more: Learning to refine dialogue history for personalized dialogue generation](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human*
- Language Technologies*, pages 5808–5820, Seattle, United States. Association for Computational Linguistics.
- Wanjun Zhong, Duyu Tang, Jiahai Wang, Jian Yin, and Nan Duan. 2021. [UserAdapter: Few-shot user learning in sentiment analysis](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1484–1488, Online. Association for Computational Linguistics.

## A Experimental Setup

**Hyperparameters** We use Hugging Face’s transformers (Wolf et al., 2020) to fine-tune our models. We fine-tune the Pythia models for 10 epochs on 8 GPUs by using a learning rate of 5e-5, a batch size of 4 with 4 gradient accumulation steps, a seed of 42, and a maximum sequence length of 1024. At the end of the fine-tuning, we pick the best checkpoint based on the performance of the Dev set.

**GPT-3.5** For the GPT-3.5 experiments, we used the prompt described in Figure 5.

```
Complete the given input sentence so that the stylometric
attributes of the completed text are close to the provided
stylometric attributes. The length of the auto-completed
text should be about 1024 tokens.
<stylometric vector> Attributes </stylometric
vector> <input> Input Text </input>
```

Figure 5: Prompt template we used for the GPT-3.5 experiments.

## B Stylistic Attributes

	Attribute	Min	Max	Avg	Std.	# Bins
<b>Lexical</b>	FKGL	2.7	15.6	7.8	2.3	8
	sents	4.7	67.6	12.6	6.9	10
	tokens	84.7	1044.4	241.2	136.8	10
<b>Discourse Relations</b>	attribution	0.3	10.5	1.8	1.3	4
	contrast	0.0	4.0	0.8	0.6	3
	elaboration	5.4	80.4	18.0	10.3	10
<b>POS Tags</b>	ADJ	4.8	66.3	16.8	9.8	10
	ADP	5.0	93.4	21.7	13.4	10
	ADV	1.8	59.1	12.5	8.0	10
	AUX	4.0	60.6	13.8	7.9	9
	CONJ	2.5	37.3	8.2	4.8	9
	DET	6.9	74.8	20.7	11.9	10
	NOUN	10.9	156.8	38.5	22.0	10
	NUM	0.6	15.4	2.6	1.9	4
	PART	1.9	30.1	6.8	4.2	8
	PRON	3.4	123.5	22.1	14.0	10
	PROPN	1.4	74.1	16.4	12.5	10
	PUNCT	6.2	165.1	30.0	18.9	10
	SCONJ	0.8	22.4	4.6	3.1	7
VERB	8.5	120.8	25.3	15.0	10	
<b>Dependency Relations</b>	ROOT	4.7	67.6	12.6	6.9	10
	acl	0.2	5.6	1.1	0.9	3
	acomp	0.6	12.5	3.2	1.9	5
	advcl	1.1	18.9	3.8	2.4	6
	advmod	2.2	63.0	13.6	8.7	9
	amod	3.0	56.1	12.2	7.8	9
	appos	0.3	10.3	2.1	1.9	5
	attr	0.8	14.2	3.4	1.8	6
	aux	2.6	42.8	8.9	5.8	9
	auxpass	0.2	7.2	1.6	1.2	4
	case	0.0	5.8	1.1	1.1	4
	cc	2.5	37.0	8.2	4.8	9
	ccomp	0.8	21.6	4.1	2.9	6
	compound	1.4	36.6	10.0	6.6	10
	conj	2.4	37.8	8.6	5.0	9
	det	6.7	73.9	20.4	11.8	9
	dobj	3.7	47.1	10.6	6.0	10
	mark	0.5	16.3	3.2	2.2	5
	neg	0.2	12.4	2.3	1.6	5
	nmod	0.1	17.5	0.9	1.7	3
	npadvmod	0.4	13.2	1.5	1.3	4
	nsubj	4.9	94.8	20.2	11.4	10
	nsubjpass	0.2	6.7	1.4	1.0	4
	nummod	0.3	12.0	1.6	1.2	4
	pcomp	0.2	8.0	1.2	0.9	3
	pobj	4.3	78.5	18.6	11.5	10
	poss	0.7	21.6	4.6	3.1	7
	prep	4.5	82.4	19.6	12.1	10
	prt	0.1	9.2	1.5	1.1	4
	punct	6.3	185.0	30.3	19.5	10
relcl	0.5	12.7	2.9	1.9	6	
xcomp	0.7	13.6	2.7	1.8	5	

Table 4: List of the linguistic attributes we model. The minimum, maximum, average, standard deviation and the # of bins are based on the training data.

## C Stylistic Attributes Performance

	Attribute	1B-Baseline		1B-Prefix	
		Success Rate	Relative Improvement	Success Rate	Relative Improvement
<b>Lexical</b>	FKGL	34.9	144.1	<b>46.5</b>	<b>225.7</b>
	sents	23.0	106.6	<b>29.3</b>	<b>163.2</b>
	tokens	13.9	25.4	<b>33.0</b>	<b>196.9</b>
<b>Discourse Relations</b>	attribution	57.5	72.4	<b>71.1</b>	<b>113.2</b>
	contrast	75.2	50.4	<b>85.1</b>	<b>70.1</b>
	elaboration	19.5	75.8	<b>30.4</b>	<b>173.2</b>
<b>POS Tags</b>	ADJ	21.8	96.1	<b>35.5</b>	<b>219.6</b>
	ADP	18.5	66.6	<b>41.0</b>	<b>269.0</b>
	ADV	25.1	125.9	<b>38.5</b>	<b>246.3</b>
	AUX	23.4	87.3	<b>43.1</b>	<b>244.8</b>
	CONJ	20.7	65.4	<b>28.8</b>	<b>130.4</b>
	DET	27.6	148.6	<b>48.2</b>	<b>333.8</b>
	NOUN	14.3	28.5	<b>28.7</b>	<b>158.7</b>
	NUM	<b>45.5</b>	<b>36.5</b>	42.4	27.3
	PART	29.9	109.3	<b>53.5</b>	<b>274.7</b>
	PRON	18.9	70.1	<b>38.8</b>	<b>249.1</b>
	PROPN	<b>24.0</b>	<b>116.2</b>	21.3	92.1
	PUNCT	23.0	106.7	<b>33.5</b>	<b>201.8</b>
	SCONJ	27.7	66.2	<b>53.0</b>	<b>218.1</b>
VERB	13.8	24.5	<b>33.1</b>	<b>197.8</b>	
<b>Dependency Relations</b>	ROOT	23.0	106.6	<b>29.3</b>	<b>163.2</b>
	acl	65.3	30.7	<b>79.9</b>	<b>59.8</b>
	acomp	39.7	58.9	<b>61.2</b>	<b>144.8</b>
	advcl	31.6	58.1	<b>64.5</b>	<b>222.4</b>
	advmod	30.9	147.5	<b>36.1</b>	<b>188.8</b>
	amod	27.0	116.2	<b>41.3</b>	<b>230.6</b>
	appos	56.3	125.1	<b>67.4</b>	<b>169.5</b>
	attr	35.7	78.5	<b>56.3</b>	<b>181.7</b>
	aux	23.4	87.3	<b>43.1</b>	<b>244.8</b>
	auxpass	58.9	76.7	<b>75.3</b>	<b>125.8</b>
	case	74.4	123.1	<b>74.9</b>	<b>124.6</b>
	cc	21.9	74.9	<b>33.1</b>	<b>165.0</b>
	ccomp	37.8	88.8	<b>55.9</b>	<b>179.7</b>
	compound	<b>21.3</b>	<b>91.3</b>	19.5	75.7
	conj	20.7	65.4	<b>28.8</b>	<b>130.4</b>
	det	27.6	148.6	<b>48.2</b>	<b>333.8</b>
	dobj	11.7	4.8	<b>22.0</b>	<b>98.3</b>
	mark	44.3	77.2	<b>65.4</b>	<b>161.5</b>
	neg	49.1	96.6	<b>66.9</b>	<b>167.7</b>
	nmod	<b>83.0</b>	<b>65.9</b>	82.2	64.5
	npadvmod	64.5	93.4	<b>70.3</b>	<b>110.9</b>
	nsubj	20.0	79.8	<b>41.6</b>	<b>274.8</b>
	nsubjpass	68.9	106.7	<b>75.9</b>	<b>127.6</b>
	nummod	63.2	89.5	<b>65.2</b>	<b>95.5</b>
	pcomp	67.7	35.4	<b>78.1</b>	<b>56.2</b>
	pobj	20.4	83.7	<b>41.2</b>	<b>270.9</b>
	poss	35.2	111.4	<b>55.5</b>	<b>232.8</b>
	prep	22.9	106.1	<b>45.5</b>	<b>309.7</b>
	prt	63.4	90.2	<b>72.8</b>	<b>118.5</b>
	punct	23.0	106.7	<b>33.5</b>	<b>201.8</b>
relcl	39.9	99.5	<b>67.8</b>	<b>239.0</b>	
xcomp	49.9	99.5	<b>65.8</b>	<b>163.3</b>	

Table 5: Attribute-specific performance of the 1B Pythia baseline and Prefix models. The best results are in bold.