Emulating Author Style: A Feasibility Study of Prompt-enabled Text Stylization with Off-the-Shelf LLMs

Avanti Bhandarkar **Ronald Wilson** **Anushka Swarup** **Damon Woodard**
Florida Institute for National Security
University of Florida, Gainesville, FL, USA
{avantibhandarkar, ronaldwilson, aswarup, dwoodard}@ufl.edu

Abstract

User-centric personalization of text opens many avenues of applications from stylized email composition to machine translation. Existing approaches in this domain often encounter limitations in data and resource requirements. Drawing inspiration from the success of prompt-enabled stylization in related fields, this work conducts the first feasibility study into 12 pre-trained SOTA LLMs for author style emulation. Although promising, the results suggest that current off-the-shelf LLMs fall short of achieving effective author style emulation.

1 Introduction

Driven by the trend of using Generative AI for on-demand user-centric personalization in recent years, the demand for personalized content has become increasingly pronounced. Personalized text, generated by capturing the style of an author, is sought-after in creative content writing, data-to-text generation, email composition as well as machine translation to provide user-specific "naturalness" to text. Prior works attempted at replicating an author's writing style by mapping the content to a particular style but remained a challenging task due to the indecipherable and individualistic nature of writing style.

LLMs, with their understanding of natural languages via high-level latent representations, serve as versatile tools for linguistic analysis and manipulation. Therefore, they are explored abundantly in various text-based applications such as information retrieval, sentiment analysis, etc. (Lu et al., 2023; Zhu et al., 2023). In the realm of controllable text-generation, input optimization or prompt engineering has enabled a resource-efficient alternative to modulate LLM-generated text by modifying the input prompts (Zhang et al., 2023). More recently, this approach has been used to produce valid explanations for various stylistic textual entailment tasks and produced promising results in related fields such as Authorship Verification and Personality Prediction (Hung et al., 2023; Ji et al., 2023). Motivated by these observations, this work performs the first feasibility study into using off-the-shelf pre-trained LLMs for controllable stylized text generation via prompting for author style emulation.

2 Related Works

Author Style Emulation, within personalized text generation, seeks to replicate the distinctive styles of individual human authors. It is often conceived in two ways - Text Stylization or Text Style Transfer (TST). The former equips a text generator to produce author-stylized text, while the latter independently extracts an author's linguistic preferences (style) and modulates a text generator's semantic content accordingly.

Early work in this field mainly investigated TST, by attempting to *Shakerspearize* texts using parallel corpora produced with and without Shakespearean style (Xu, 2017; Jhamtani et al., 2017). However, limited availability of parallel corpora for an average author's style hindered progress in this direction (Hu et al., 2020). Motivated by the need for non-parallel data, some research works exploring TST for sentiment and formality utilized generative models, such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAE) etc., to obtain latent representation corresponding to the style and semantic content separately (Shen et al., 2017; Li et al., 2018). However, this line of research was not as explored for author-stylized text generation as the intricate relationship between an author's linguistic style and semantic content would have made their separation a challenging task.

In the recent past, Syed et al. (2020) exploited the capabilities of LLMs such as GPT2 by fine-tuning the model on an author's corpus to incorpo-

Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024), pages 76–82

Table 1: Brief Description of the Text Generators

Type	Text Generator	Size	Description
AutoR	GPT3	175B	The OpenAI text completion API used with the *davinci* engine.
Chat	GPT4	N/A	OpenAI chat completion API used with the *Turbo* engine.
	OpenChat3.5	7B	Trained with Conditioned-RLFT on instruction data (Wang et al., 2023a).
	CausalLM	14B	Trained on manually curated SFT dataset from many open-source corpora.
	Zephyr	7B	Fine-tuned on synthetic datasets using DPO algorithm. β variant used(Tunstall et al., 2023)
	Llama-2	7B	Fine-tuned on instruction datasets using SFT and RLHF (Touvron et al., 2023)
	NeuralChat	7B	Fine-tuned on an open source dataset using DPO algorithm . Version v3-1 used.
	DeepSeek	7B	Trained on a corpus of 2 trillion tokens in English and Chinese.
Instruct	GPT3.5	N/A	OpenAI text completion API used with the *turbo − instruct* engine.
	Falcon	7B	Fine-tuned using the base model on chat and instruct datasets(Almazrouei et al., 2023).
	Mistral	7B	Instruction fine-tuned model trained on conversation datasets (Jiang et al., 2023).
	INCITE	7B	Instruction fine-tuned model on a collection of instruction datasets by RedPajama project.

Note: The texts are generated using respective model's HuggingFace model repository unless otherwise specified.
Abbrev.: AutoR - Auto-regressive, SFT - Supervised Fine-Tuning, RLFT - Reinforcement Learning Fine-Tuning, RLHF - Reinforcement Learning with Human Feedback, DPO - Direct Preference Optimization

rate their style characteristics without the need for a parallel corpus. The idea was to utilize the LLMs for *stylistic rewriting* than simple text generation (i.e., writing). However, this method requires large amounts of labeled data with author identities to train a resource-heavy LLM such as GPT2.

More recently, due to the influx of large number of LLMs performing competitively on natural language generation and understanding tasks, investigations have been made to utilize off-the-shelf pre-trained LLMs for related applications such as Authorship Verification and Automatic Personality Prediction through prompt modulation with promising results (Hung et al., 2023; Ji et al., 2023). These results emphasize the capability of the LLMs to understand and follow instructions through various prompting strategies and guided instructions. Specifically, these works instruct the LLM to find entailment between a text and a label (for personality prediction) or between two texts (for authorship verification), while also providing explanations. However, to the best of the author's knowledge, no work has been done to extend this concept to generate text with an objective of emulating an author's writing style.

3 Methodology

Problem Statement: Given human-authored text T_A by an author A, generate T_G using a text generator G that faithfully mirrors the style of the author in T_A. The success criterion is for the generator to produce T_G such that a proficient author discriminator D attributes it accurately to the original author A.

To meet this objective, three key elements come into play: Author (A), Generator (G) and Discriminator (D). A signifies the identity label associated with a human-authored text sample T_A containing the author's linguistic preferences. Additionally, it is assumed that all text samples by A have consistent linguistic preferences that form the author's style signature. G modifies its language generation for different A by learning the author's stylistic preferences from T_A independently or by virtue of instructions. Finally, D must be capable of accurately differentiating between several authors based on their style signatures.

Authorship Attribution (AA) research focuses on creating robust algorithms to differentiate authors based on their distinct writing styles. Therefore, AA can serve as a source for text data with author identity labels, while effective AA algorithms become relevant tools for serving as author discriminators. For the purpose of this study text samples with author identity labels are obtained from 100 authors randomly chosen from a widely recognized AA corpus, the Blogs Authorship Corpus (Schler et al., 2006). The selected authors are ensured at least 100 text samples and a minimum of 500 words in each sample.

Next, off-the-shelf pre-trained text generators, the LLMs, are selected. This selection encompasses 12 state-of-the-art (SOTA) LLMs, as described in Table 1, capable of diverse text production including auto-regressive, chat, and instruction-tuned models. While the auto-regressive models are trained to predict the subsequent word based on the preceding text, chat and instruction-tuned models offer greater flexibility in text production having been trained on conversational interactions or task-specific instructions.

For consistency in text generation across LLMs,

Table 2: Summary Statistics of Generated Text Data

Text Generator	Num. words	Num. sentences
GPT3	468±98	25±13
GPT4	520±17	25±5
OpenChat	427±24	24±8
CausalLM	466±27	23±6
Zephyr	420±22	21±5
Llama-2	396±32	21±8
NeuralChat	413±25	22±13
DeepSeek	461±14	19±9
GPT3.5	522±28	28±5
Falcon	446±61	22±8
Mistral	433±28	24±12
INCITE	476±29	28±9

Note: Statistics are reported as mean±std. NLTK (https://www.nltk.org/) was used for word and sentence tokenization.

Table 3: Brief Description of the AA Algorithms

Algorithm	Description
Writeprints	Random Forest Classifier trained on Writeprints features (Mahmood et al., 2019)
LIWC	Random Forest Classifier trained on LIWC psycholinguistic features (Boyd et al., 2022)
Char-3-grams	SVM classifier trained on character 3-gram features using one-vs-rest classification strategy. (Kestemont et al., 2019)
BertAA	Cascaded architecture integrating fine-tuned BERT classifier, stylistic features (e.g., text length, word count), and hybrid features (e.g., most frequent character 2-gram) trained using Logistic Regression. (Fabien et al., 2020)
Contra-X	Contrastive learning with DeBERTa for cross-entropy fine-tuning, followed by classification using a 2-layer MLP. (Ai et al., 2022)

top-K and top-p (nucleus) sampling with K=50 and p=0.95 is used as decoding strategy wherever applicable. The maximum generation length is set to 500 tokens with a minimum requirement of 350 tokens. For each author, 10 text samples are reserved to serve as examples of author's writing style. Every generator is equipped with a prompt and an example text with a specific identity. Therefore, for each author-generator pair, total 10 text samples are generated. Summary statistics for the generated data[1] are provided in Table 2.

AA explores author discrimination at various levels resulting in over 1000 features, ranging from granular features like character n-grams to utilizing contextual features such as BERT (Tyo et al., 2022; Wilson et al., 2021). For optimal evaluation of the stylistic alignment between author and LLM generated text, it is imperative to consider diverse AA algorithms that capture various aspects of author's style. Table 3 describes the most popular and SOTA AA algorithms that are selected to serve as author discriminators.

For each algorithm, training is performed on 90 text samples per author. For the algorithms that require separate training and validation sets, 90% of the author training data is allocated for training while the remaining 10% is reserved for validation. Both BertAA and Contra-X algorithms are trained for 5 epochs each. The results are reported as average accuracy across 5 training-testing runs.

Ultimately, assessing the capability of G for emulating A's style necessitates maintaining A and D constant while varying the inputs of G. As prompts serve as inputs to each LLM, the distinct components of the prompt are adjusted, and the impact on D's performance for each A is evaluated. Detailed specifications of the prompt's individual components are discussed in the following section.

[1]The text data generated for this study can be accessed here

4 Prompting Protocol for Stylization

Following the prompt decomposition technique outlined in Giray (2023), a four-part prompt, namely, task, instructions, output indicator and example author text, is designed (see Table 4). Initially, similar to Wang et al. (2023b), a *Trivial Emulation Protocol* (TEP) is considered where the LLM is provided with a simple task definition along with short snippet of author text. This protocol relies on the LLM's capability of capturing author's unique style representation in an unguided scenario.

Some research works observed that furnishing the LLM with additional task-specific knowledge in the form of guided instructions greatly enhanced its ability to consider relevant textual characteristics (Hung et al., 2023). Therefore, a *Complex Emulation Protocol* (CEP) is developed where the LLM is provided with additional author data and/or guided instructions in the form of few key linguistic features that potentially demonstrate the author's unique linguistic preferences (Boenninghoff et al., 2019). Thus, the *prompting strategy* and *length of example author text* serve as the control parameters for evaluation of the two protocols.

For the purpose of testing, an example author text is categorized as either short or long text, representing the first 50 and 300 words from the original author example, respectively. Additionally, two variations of the prompting strategy involves exclusion and inclusion of guided instructions, identi-

Table 4: Elements of Prompt used for Author Emulation

Prompt Element	Prompt Text
Task	<SYS> You are an **emulator** designed to replicate the writing style of a human author.<\SYS> Your task is to generate a 500-word continuation that *seamlessly integrates with the provided human-authored snippet*. Strive to make the continuation *indistinguishable from the human-authored text*.
Instructions	The goal of this task is to **mimic the author's writing style** while paying meticulous attention to *lexical richness and diversity*, sentence structure, *punctuation style, special character style*, expressions and idioms, *overall tone, emotion* and mood, or any other relevant aspect of writing style established by the author.
Output Indicator	As output, exclusively return the text completion without any accompanying explanations or comments.
Example author text	Text snippet : [50 or 300-word human authored text]

Note: Text enclosed in <SYS> and <\SYS> indicates the system prompt provided to chat models; features emphasized in instructions are linguistically verifiable.

fied as simple and directed prompting, respectively. Across both emulation protocols, the task and output indicator remain consistent.

Evaluation of the author emulation protocols involves testing the trained AA algorithms on both the example author texts and the text generated by each LLM. Results are presented as average accuracy across authors for each LLM in the form of box-and-whisker plot. Assuming that the example author texts contain sufficient stylistic information about their respective authors, *LLMs capable of emulating an author's writing style are expected to demonstrate comparable AA performance on the synthetic LLM-generated text to that observed on the original author texts*.

5 Discussion

Results of TEP are presented in Figure 1. In short texts, the expected low stylometric information corresponds to relatively low performance across all AA algorithms on author text. Notably, GPT-4 outperforms other LLMs, achieving maximum accuracy close to 72% of the maximum accuracy achievable on author text. Being one of the most advanced SOTA models amongst the chosen LLMs, in the absence of specific instructions, GPT4 may leverage the knowledge from its training data to identify and replicate relevant linguistic style representations. Interestingly, the top performing LLMs

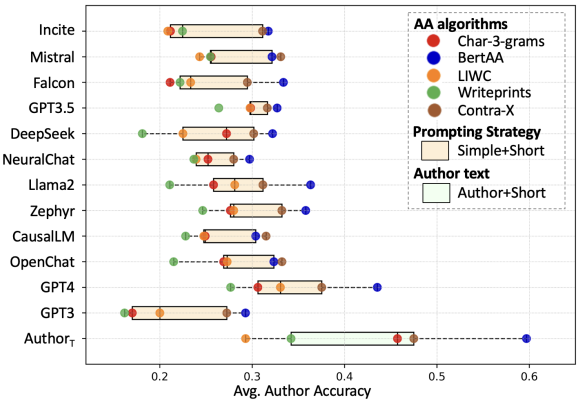

Figure 1: Results of Trivial Emulation Protocol (TEP)

in this protocol - GPT4, Llama2 and Zephyr - are chat models alluding to the potential of chat models for author style emulation in an unguided scenario.

For CEP, three scenarios are evaluated by providing progressively more information in the prompt. The results of this protocol are presented in Figure 2. DeepSeek, a chat model, consistently outperforms all LLMs across scenarios, showcasing strong potential for author style emulation, followed closely by GPT3 and Incite. No clear consensus emerges on the optimal model type for CEP.

With regards to the first control parameter - length of example author text - an expected significant performance improvement is observed across all AA algorithms between short and long author texts. Similarly, most LLMs benefit from addition of more author data, albeit with varying degrees.

Considering the second control parameter - prompting strategy - most LLMs exhibit reduced performance while transitioning from simple to directed prompting. This observation may be caused by one or more of the factors discussed below.

First, the LLM's intrinsic understanding of an author's writing style may be better. Each LLM is trained on large amounts of training data spanning diverse domains that allow the model to learn intricate linguistic structures of language. Given that the LLMs are trained with a probabilistic objective, aiming to generate the next word that most closely aligns with the preceding context, it is plausible that these models develop an understanding of the author's writing style from the example author text. Consequently, constraining the LLM to generate text by focusing on the linguistic features highlighted in the directed prompt might interfere with its intrinsic understanding, thereby leading to a degradation in performance.

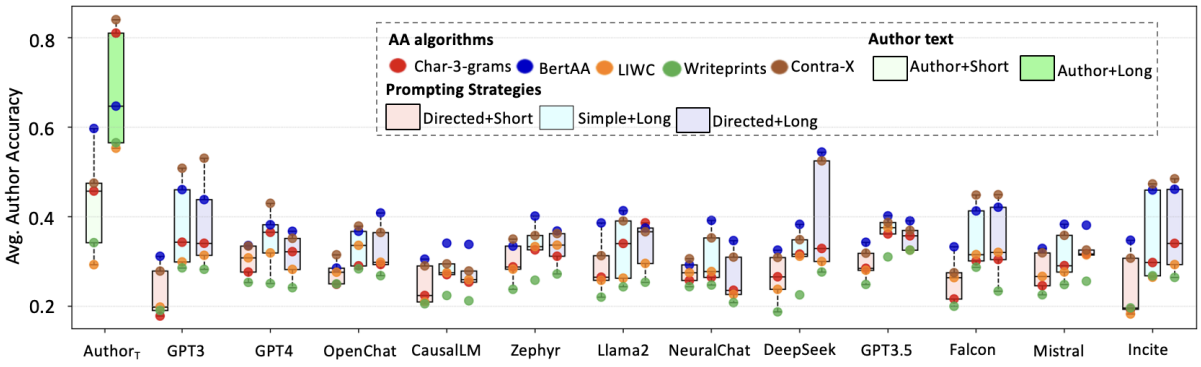

Figure 2: Results of Complex Emulation Protocol (CEP)

Second, limited efficacy of the linguistic features in the prompt in capturing author's style. The directed prompt's linguistic features form a subset of the LIWC and Writeprints feature sets. The lower performance of AA algorithms using these features suggests two possibilities: their possible absence in the author's text or limited influence on their writing style. However, the superior performance of these algorithms on the author's text samples, compared to the generated texts, suggests the influence of these features on the author's writing style to some extent. Nevertheless, considering the subjective nature of an author's linguistic choices, the assumption that a fixed set of linguistic features can universally influence all authors equally might be flawed. Therefore, instead of static directed prompting, a more effective approach could involve dynamically prompting LLMs by considering each author's individual linguistic preferences.

Finally, some LLMs may lack the capability to incorporate the specified linguistic features highlighted in the instructions during text generation. To evaluate the similarity of the linguistically verifiable features from the directed instructions between the LLM generated texts and author's example texts, a test was performed as described in A.1. As anticipated, the LLMs exhibit superior alignment to the said linguistic features in longer texts due to increased data availability. Further it is noted that LLMs are most capable of producing higher alignment with linguistic features that reflect the tone, authenticity, analytical aspects, and lexical richness from author texts. However, replicating the lexical diversity and punctuation style proves more challenging. One of the notable observations is the lack of overall linguistic alignment between the author's texts and the best-performing model, DeepSeek, emphasizing the potential significance of linguistically unverifiable aspects in the instructions for successful emulation.

6 Conclusion

This study explored the use of pre-trained LLMs to generate author-stylized text through prompted inputs, controlling parameters such as example text length and directed instructions with stylometric information. Evaluated against five AA algorithms, the two author emulation protocols (trivial and complex) assessed the feasibility of 12 SOTA LLMs for *Author Style Emulation*. Control parameters significantly impacted the LLM's emulation capacity, emphasizing the need for user-specific personalized instruction generation. Overall, the maximum author emulation performance is only two-thirds that of original texts, highlighting the LLM's current limitations for plug-and-play author style emulation. As this task involves a complex interplay between an author identity, text generator and author discriminator, an extended future work will explore how individual author identities and the properties of author discriminator impact the subsequent author style emulation capabilities of a text generator, specifically the LLMs.

7 Limitations

The feasibility study performed in this work has a few potential limitations with respect to subjectivity, the choice of control parameters, scalability and generalizability. In this work, the results are averaged across authors which may conceal subjective preferences of individual authors. The preliminary choice of control parameter - length of author text - is specific to the author corpus utilized for this work and assumes one example with maximum 300 words sufficiently represents author's unique linguistic choices. Due to the absence of previous

work performing prompt-enabled author-stylized text generation, the control parameter - prompting strategy - is limited by two choices - unguided and guided with static instructions. The generalizability and scalability of this work is limited by the choice of a small author set size with 100 authors.

References

Bo Ai, Yuchen Wang, Yugin Tan, and Samson Tan. 2022. Whodunit? learning to contrast for authorship attribution. *arXiv preprint arXiv:2209.11887*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Benedikt Boenninghoff, Steffen Hessler, Dorothea Kolossa, and Robert M Nickel. 2019. Explainable authorship verification in social media via attention-based similarity learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 36–45. IEEE.

Ryan L Boyd, Ashwini Ashokkumar, Sarah Seraj, and James W Pennebaker. 2022. The development and psychometric properties of liwc-22. Technical report, University of Texas at Austin.

Maël Fabien, Esaú Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. Bertaa: Bert fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 127–137.

Louie Giray. 2023. Prompt engineering with chatgpt: A guide for academic writers. *Annals of Biomedical Engineering*, pages 1–5.

Zhiqiang Hu, Roy Ka-Wei Lee, Charu C Aggarwal, and Aston Zhang. 2020. Text style transfer: A review and experimental evaluation. *arXiv preprint arXiv:2010.12742*.

Chia-Yu Hung, Zhiqiang Hu, Yujia Hu, and Roy Lee. 2023. Who wrote it and why? prompting large-language models for authorship verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14078–14084.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *EMNLP 2017*, 6:10.

Yu Ji, Wen Wu, Hong Zheng, Yi Hu, Xi Chen, and Liang He. 2023. Is chatgpt a good personality recognizer? a preliminary study. *arXiv preprint arXiv:2307.03952*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Mike Kestemont, Efstathios Stamatatos, Enrique Manjavacas, Walter Daelemans, Martin Potthast, and Benno Stein. 2019. Overview of the cross-domain authorship attribution task at {PAN} 2019. In *Working Notes of CLEF 2019-Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019*, pages 1–15.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*.

Qiang Lu, Xia Sun, Yunfei Long, Zhizezhang Gao, Jun Feng, and Tao Sun. 2023. Sentiment analysis: Comprehensive reviews, recent advances, and open challenges. *IEEE Transactions on Neural Networks and Learning Systems*.

Asad Mahmood, Faizan Ahmad, Zubair Shafiq, Padmini Srinivasan, and Fareed Zaffar. 2019. A girl has no name: Automated authorship obfuscation using mutant-x. *Proc. Priv. Enhancing Technol.*, 2019(4):54–71.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. *arXiv preprint arXiv:1705.09655*.

Bakhtiyar Syed, Gaurav Verma, Balaji Vasan Srinivasan, Anandhavelu Natarajan, and Vasudeva Varma. 2020. Adapting language models for non-parallel author-stylized rewriting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9008–9015.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Jacob Tyo, Bhuwan Dhingra, and Zachary C Lipton. 2022. On the state of the art in authorship attribution and authorship verification. *arXiv preprint arXiv:2209.06869*.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023a. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Yaqing Wang, Jiepu Jiang, Mingyang Zhang, Cheng Li, Yi Liang, Qiaozhu Mei, and Michael Bendersky. 2023b. Automated evaluation of personalized text generation using large language models. *arXiv preprint arXiv:2310.11593*.

Ronald Wilson, Avanti Bhandarkar, Princess Lyons, and Damon L Woodard. 2021. Sqse: a measure to assess sample quality of authorial style as a cognitive biometric trait. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(4):583–596.

Wei Xu. 2017. From shakespeare to twitter: What are language styles all about? In *Proceedings of the Workshop on Stylistic Variation*, pages 1–9.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):1–37.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

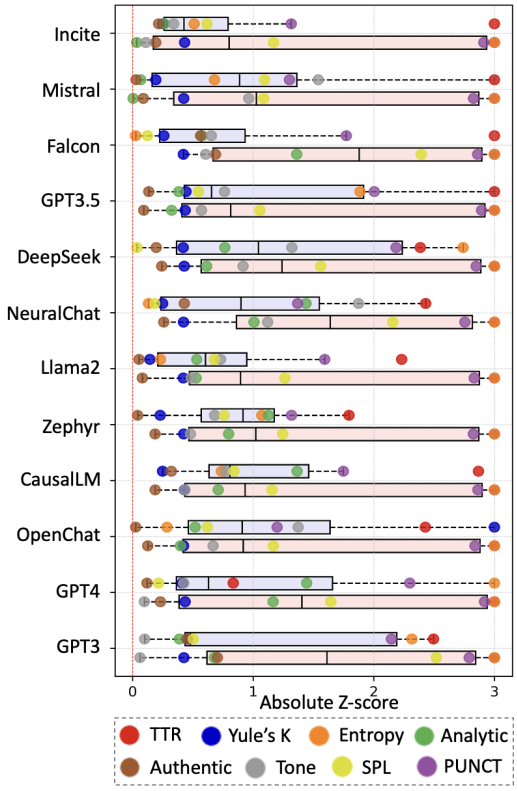

Figure 3: Results of linguistic feature alignment test [SPL refers to special character frequencies; PUNCT refers to frequencies of punctuation characters]

A Appendix

A.1 Linguistic Feature Analysis

A test is designed to evaluate the adherence of each LLM to the linguistically verifiable features highlighted in the instructions. Success criteria of an LLM modulating its generation by incorporating the linguistic features is measured by a Z-score. Initially, for each feature the mean (μ) and standard deviation (σ) for every author is computed from the author's short and long texts. Further, the absolute Z-score is calculated from each LLM generated text by comparing the feature (x) to respective author's mean and standard deviation as described in equation 1. The range for Z is set to be [0,3], where a $Z \geq 3$ is capped at 3. Finally, an average Z-score is computed for every feature-LLM combination, where a near-zero score indicates better alignment between the author text and the LLM generated text.

$$Z = \left| \frac{x - \mu}{\sigma} \right| \tag{1}$$

To assess lexical richness and diversity, three metrics are examined: Yule's K measure, Type-Token Ratio (TTR) and Shannon's Entropy. TTR evaluates lexical diversity, Shannon's Entropy quantifies the unpredictability or diversity of word usage in a text, and Yule's K measure evaluates the richness and evenness of word frequency distribution. The psycho-linguistic feature set, LIWC, offers three key linguistic dimensions, namely, Tone, Authentic and Analytic that capture the overall emotional tone, degree of authenticity and formal or structured thinking respectively in text. Finally, Writeprints feature set offers two categories of features - punctuation style and special character style. Independent category averages result in two subsequent linguistic features, bringing the total evaluated linguistic features to eight. The results of this test are presented in Figure 3.

The LIWC features and Yule's K measure are found to be consistently less than a Z-score of 1 indicating most similarity amongst all features. However, TTR and Shannon's entropy consistently show highest variance.