LLM Output Compliance with Handcrafted Linguistic Features: An Experiment

Andrei Olar[a]

Mathematics and Computer Science Faculty, Babeș-Bolyai University, Cluj-Napoca, Romania

Keywords: Handcrafted Linguistic Features, Large Language Models, Controlled Text Generation, Text Style Transfer.

Abstract: Can we control the writing style of large language models (LLMs) by specifying desired linguistic features? We address this question by investigating the impact of handcrafted linguistic feature (HLF) instructions on LLM-generated text. Our experiment evaluates various state-of-the-art LLMs using prompts incorporating HLF statistics derived from corpora of CNN articles and Yelp reviews. We find that LLMs demonstrate sensitivity to these instructions, particularly when tasked with conforming to concrete features like word count. However, compliance with abstract features, such as lexical variation, proves more challenging, often resulting in negative impacts on compliance. Our findings highlight the potential and limitations of utilizing HLFs for guiding LLM text generation and underscore the need for further research into optimizing prompt design and feature selection.

1 INTRODUCTION

Large language models (LLMs) have become a commonplace research topic today. Because language models such as BERT (Devlin et al., 2019) or GPT (Radford et al., 2018; Radford et al., 2019; Brown et al., 2020) constantly advance the state of the art on many natural language processing tasks, it is interesting to evaluate them on more specialized tasks. Multiple surveys (Minaee et al., 2024; Zhao et al., 2023) and benchmarks (paperswithcode.com, 2024; Chiang et al., 2024) show that large language models are good at following instructions. We intuit that training LLMs on diverse data (for instance the Pile (Gao et al., 2020)) uniquely qualifies them to produce text in a wide variety of styles.

Text style transfer (TST) is defined as the "task of transforming the stylistic manner in which a sentence is written, while preserving the meaning of the original sentence" (Toshevska and Gievska, 2022). This definition can be extended to entire articles or corpora of text because these are also the object of linguistic style and stylistics (Lugea and Walker, 2023). The task of transferring text style has a certain maturity. The interest in this task is renewed by the advancements made with LLMs.

Handcrafted linguistic features (HLFs) are single numerical values produced by a uniquely identifiable method on any natural language (Lee and Lee, 2023). Examples of HLFs range from simple constructs such as counts or averages of words, sentences or specific parts of speech (adjectives, verbs, a.s.o) to more complex statistics based on heuristics. All HLFs share the characteristics of computational ease and idempotence. This makes HLFs very attractive because of their dual potential. On the one hand, an LLM might be instructed using a prompt containing the value of an HLF. An example prompt instruction which asks the LLM to conform to a maximum *sentence count* of 10 could be "write no more than 10 sentences". On the other hand, HLFs can be computed on the text output by an LLM, too. This dual nature of HLFs allows inspecting how closely does LLM generated text conform to the goal outlined in the input prompt HLF instruction.

Syntax differs from meaning (Chomsky, 2002) and the intuition is that writing style is defined in terms of both. The distinction is important in order to set boundaries. HLFs are largely focused on syntax and similarly orderly constructs. Therefore, we cover the influence of meaning on author style minimally and superficially at best.

This limitation should not deter readers because, traditionally, linguistic style is centered around morphological and syntactic arrangements (Lugea and Walker, 2023). Although recent descriptions find se-

[a] https://orcid.org/0009-0006-7913-9276

Olar, A.
LLM Output Compliance with Handcrafted Linguistic Features: An Experiment.
DOI: 10.5220/0013350400003890
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 2*, pages 767-775
ISBN: 978-989-758-737-5; ISSN: 2184-433X

mantics and pragmatics to be inextricably linked to author style (Verma and Srinivasan, 2019), they too argue that a great deal can still be inferred from lexicon, syntax and a ‘surface’ level which contains basic HLFs such as the number of words in a sentence. Some have observed that many characteristics of written text can be computed as HLFs (Hovy, 1987; Lugea and Walker, 2023). Combining multiple HLFs that are revelatory of linguistic style could provide us with an easily interpretable and verifiable stylistic profile.

These are the strands of thought that have inspired us to pursue two research questions in the age of LLMs.

Firstly, **is it possible to influence an LLM’s writing style by instructing it to generate text that complies to certain HLFs**? This problem of controlled text generation has implications for text style transfer and authorship verification. Answering this question might point to cheaper, less intrusive and more user friendly solutions than fine tuning large language models for these tasks. Fine tuning LLMs also incurs some loss of generality (Yang et al., 2024), going directly against the purpose of generic language models.

Secondly, we ask **if it’s possible to quantify how closely an LLM is able to follow prompts concerning the compliance of the generated text to HLFs?** The answer to this question could hint at a relatively simple and accurate way of evaluating the quality of (partial) text style transfer using a large language model.

2 RELATED WORK

Text style transfer reviews and surveys (Toshevska and Gievska, 2022; Jin et al., 2022) were helpful for informing our selection of HLFs. Studies on attaining fine-grained text style transfer using language models have served as inspiration (Lyu et al., 2023).

Linguistic style is the sum of identifiable language choices manifest in a text made from the language system by the text producer (Lugea and Walker, 2023). Another way to think about it is that style is the form used for delivering meaning (Hu et al., 2022). Opinions seem to converge that style depends on context and author choice vis-a-vis a communication goal (McDonald and Pustejovsky, 1985; Hovy, 1987). The author’s choice is available at all linguistic levels, including the morphological, lexical, syntactic, semantic and pragmatic levels (DiMarco and Mah, 1994; Lugea and Walker, 2023). The influence of syntactic structure over text style is apparent from a formal perspective (Chomsky, 2002). Our selection of HLFs is informed by the various aspects of linguistic style.

Constructing style from fine-grained aspects is not new. The StylePTB authors explain this in detail with respect to lexical, syntactic, semantic and thematic aspects (Lyu et al., 2021). The dataset was useful for understanding HLFs that function at the sentence level and the interplay of HLFs.

The body of work on HLFs is extensive and well referenced and synthesised by Lee and Lee (Lee and Lee, 2023). To our knowledge there is no work that connects HLFs with LLMs in the manner described in this paper. HLFs are used in the context of other, connected tasks such as assessing text readability (Lee et al., 2021).

Our approach stands out through its simplicity and focus. We propose engineering instruction prompts for LLMs, a straightforward strategy distinct from complex alternatives. With discussions on style transfer or author verification beyond this paper’s scope, we present a method for controlling LLM text generation through the use of HLFs.

The presented experiment is one of controllable text generation (CTG) (Zhang et al., 2023). Hightened recent interest in CTG and especially in benchmarking LLMs in the context of CTG (Chen et al., 2024) is particularly relevant for this paper. This lessens the burden of demonstrating how effectively LLMs respond to varied instructions for general controlled text generation. This paper differs from the existing CTG work in its use of HLFs for writing the text generation instructions. It additionally uses HLFs to measure the performance of LLMs in terms of their ability to generate text that conforms to some chosen target HLF values.

3 EXPERIMENT DESIGN

Our experiment aims to find out how well LLMs can follow prompts which have instructions derived from HLFs. Two scenarios are investigated.

The aim in the first scenario is to use an LLM to reword an input text so that its style resembles the text style of texts from a predetermined corpus of similarly styled texts. In this scenario the LLM rewords the input by following instructions which only contain HLFs. The input text used for this scenario is necessarily from outside the corpus.

The second scenario is designed to show how much bearing do instructions containing HLFs have on the way LLMs generate text. The task of the LLM remains the same as in the first scenario. However, examples from the chosen corpus are added to the LLM prompt which means the LLM rewords text aided by the examples. The second scenario has two variants because of this particularity. In the first variant, similarly to the first scenario, the LLM rewords input from outside the corpus. In the second variant of the second scenario the LLM uses input text from within the corpus.

3.1 Process

We start by choosing experiment parameter values. The following parameters have stable values throughout the experiment process: a selection of HLFs, a corpus containing texts, a target LLM, examples from the chosen corpus, one input text from the chosen corpus and one input text from outside the chosen corpus.

Once the parameters have been determined, HLF statistics are computed on the text corpus. Remembering that HLFs are just floating point numbers, the following statistics are computed for each HLF: **min** (the minimum value), **max** (the maximum value) and **avg** (the mean).

For each HLF, an instruction is constructed in natural language, adhering to the previously calculated minimum, maximum, and mean values. For instance, an instruction corresponding to the total word count HLF might be expressed as follows:

```
  - Total Words:  ensure the text
contains 14 words (min=10, max=50).
```

The HLF instructions are used to build system prompts for the LLM. The prompts are assembled using Jinja2 (Projects, 2024) templates located in the `templates` directory of the experiment's source code.

The prompt templates correspond to each of the two experiment scenarios: '`prompt_1`' and '`prompt_2`', respectively. The static instruction in '`prompt_1`' to "*Write text that conveys the meaning of the first user prompt*" ensures that input text meaning is preserved. In addition to the primary directive from '`prompt_1`', '`prompt_2`' contains instructions concerning the interpretation of the provided examples, as well as logic to incorporate them. Both prompt templates contain logic to distinguish between when HLF instructions were given or not.

Baseline HLFs for the LLM output are computed after the preliminary steps are completed. For the first scenario, the LLM is instructed to reword the input text 10 times by using '`prompt_1`' without HLF instructions. The HLFs in our selection are computed for each LLM output, resulting in a 10-dimensional vector for each HLF. These vectors are the baseline HLFs for the LLM output in the first scenario.

The second scenario baseline HLFs are computed similarly to the first scenario baselines, using the '`prompt_2`' template and the examples provided as an experiment parameter. Remembering that the second scenario has two variants, we obtain two sets of baseline vectors, one for each variant. Note that the input from outside the corpus is the same as the input for the first scenario, while the input from inside the corpus is selected along with the corpus so that its HLFs are close to the average HLFs of the corpus.

From here on we refer to the HLFs computed using the above process as *baselines*. We have 3 sets of baselines: 1 for the first scenario and 2 for the second scenario.

For each set of baselines, we run 10 text generations using the corresponding prompt augmented with the HLF instructions computed earlier in the process. Doing this results in 3 sets of 10-dimensional vectors, each corresponding to the baselines. We refer to these vectors as *HLF results*.

Lastly, we compare baselines with their corresponding HLF results for each HLF in our selection, for each scenario and variant.

The comparison result for the first scenario indicates whether LLMs acknowledge HLF instructions at all. If they do, the HLF results should be significantly different from their baseline. Both variants of the second scenario quantify how much HLF instructions influence text generation. The underlying assumption is that LLMs are able to mimic examples when they are provided in the system prompt. In such a case, the presence of HLF instructions would remain among the few possible explanations for deviations from the baseline, especially when the baseline is already statistically close to the desired HLF values.

3.2 LLM Selection

The LLMs selected for the experiment must represent the state of the art. The Chatbot Arena (Chiang et al., 2024) leaderboard is essential in our selection based on this criterion. To cover more ground we encourage vendor diversity in our selection. The final selection is available in Table 1.

Throughout the remainder of this paper, we will reference individual LLMs by the ID assigned to each one in Table 1.

3.3 HLF Selection

Our main focus is to understand how susceptible LLMs are to prompts containing instructions to conform to HLFs. In this experiment we use LFTK (Lee and Lee, 2023), a framework built on top of spaCy (Honnibal et al., 2020) which provides implementations for computing many HLFs. LFTK categorizes HLFs by domain and family. We select HLFs based on their domain and family from the catalog implemented by LFTK, with the interest to make a diverse selection

Table 1: Large Language Model Selection.

ID	Name	Parameters	Context Size	Elo Score	License
gpt	OpenAI GPT-4o	undisclosed	128000	1287	Proprietary
gemini	Google Gemini 1.5 Pro	undisclosed	1048576	1268	Proprietary
claude3	Anthropic Claude 3 Opus	undisclosed	200000	1248	Proprietary
llama3_70b	Meta Llama 3-70B	70 Billion	8000	1208	Llama 3 Community
command_r	Cohere Command R+	104 Billion	128000	1189	CC-BY-NC 4.0 with Acceptable Use Addendum

with respect to both these attributes. Besides the domain and family, we also choose features based on the level[1] at which they influence text style as recognized in literature (Verma and Srinivasan, 2019; Lugea and Walker, 2023).

The instructions for each HLF are specifically crafted using natural language. The resulting instructions can be catalogued on the concrete-abstract spectrum. For example, ‘write 100 words’ is a more concrete instruction than ‘write as if you were in junior highschool’. We divide HLFs in five categories ranging from very concrete to highly abstract and choose 2 features from each category.

Table 2 showcases our HLF selection. The LFTK ID is used throughout tables and figures to refer to a specific HLF.

3.4 Corpus Selection

A corpus of texts is required to perform the experiment. We have conducted the experiment using corpora derived idempotently from the datasets listed in Table 3.

The main criteria for choosing these datasets were the difference in style between the texts contained in one corpus versus the other and the similarity in style between the texts from the same corpus.

The datasets are available in the Huggingface ecosystem (Lhoest et al., 2021). Yelp reviews are loaded from `Yelp/yelp_review_full`, while CNN stories are loaded from `abisee/cnn_dailymail`.

The Yelp review data set provides a diverse assortment of reviews from online users. The authors employ a casual writing style, very similar to the writing style of an average person. We construct a corpus containing the first 1000 reviews from the test split of this dataset.

CNN/DailyMail contains stories and news with diverse styles. The style follows the publisher’s guidelines, but, within those guidelines, it varies slightly for each author. The articles are more formal, longer and more complex in structure than Yelp reviews which makes this dataset a good alternative. The corpus that is based on this dataset contains the first 1000 CNN articles from the test split of the dataset.

3.5 Input Text Choice

The experiment involves using input text from both outside the corpus and from within it.

An input text from within each corpus is chosen so that it exhibits the closest HLFs to the averages computed on the corpus. The inputs are chosen from the corpora when the corpora are selected and are stable throughout all experimental evaluations.

External input text is chosen in relation to each corpus. The chosen texts are located in the `data` folder of the experiment’s source code.

The text external to the CNN article corpus is an extract from the classic *The Life and Opinions of Tristram Shandy, Gentleman*, by Laurence Sterne (Sterne et al., 2003). The text style is not similar to the style of news articles at all. Our selected HLFs are also very different from the average values computed for the CNN article corpus. A text that is similarly divorced from the style and HLFs of most Yelp reviews is Oscar Wilde’s *Sonnet to Liberty* (Wilde, 1909). Both texts are in the public domain and were obtained from Project Gutenberg (Hart, nd).

Choosing input texts that are vastly different from their corresponding corpus is based in the assumption that LLMs would generate baseline results that do not conform to the HLF statistics on that corpus.

4 EXPERIMENT RESULTS

4.1 Interpretation Conventions

HLFs are real numbers derived from text. Plot figures show the HLF on the vertical axis and the text generation trial number on the horizontal axis. We plot the baseline using a continuous grey line and the HLF

[1] the notion of style level is described in (Lugea and Walker, 2023), chapter 1, page 7

Table 2: Handcrafted Linguistic Features.

LFTK ID	Name	Family	Domain	Abstraction	Style Level
t_word	total words	wordsent	surface	very concrete	lexical
t_sent	total sentences	wordsent	surface	very concrete	syntactic
n_uverb	total number of unique verbs	partofspeech	syntax	concrete	syntactic
n_uadj	total number of unique adjectives	partofspeech	syntax	concrete	syntactic
simp_ttr	simple type token ratio	typetokenratio	lexico-semantics	regular	pragmatic
a_verb_pw	number of verbs per word	partofspeech	surface	regular	syntactic
corr_adj_var	corrected adjective variation	lexicalvariation	lexico-semantics	abstract	lexical
corr_verb_var	corrected verb variation	lexicalvariation	lexico-semantics	abstract	lexical
fkgl	Flesch-Kincaid grade level	readformula	surface	highly abstract	pragmatic
a_kup_pw	Kuperman age of acquisition	worddiff	lexico-semantics	highly abstract	lexical

Table 3: Source Datasets.

Name	Task	Number of Samples	Content Type	Reference	License
Yelp Reviews (test split)	Sentiment Classification Text Style Transfer Text Classification	50000	Reviews	(Zhang et al., 2015)	Yelp Dataset License Agreement
CNN / DailyMail (test split)	Summarization Question Answering Text Generation	11490	News Stories	(Hermann et al., 2015) (See et al., 2017)	Apache 2.0

result using a continuous grenat line. The minimum and maximum HLF on the corpus are plotted using dotted black lines. A continuous black line designates the average HLF on the corpus. We refer to the corpus average as the *target* because the LLM is instructed to generate text that primarily conforms to this value.

Result tables present an overview over an experiment scenario variant. Firstly, they display significant differences between baselines and HLF results. Secondly, they reveal the relative closeness of the baseline and HLF results to the target average. **Bold** text represents significant differences between the baseline and HLF results. Underlined text marks HLF results that are closer to the target than the baseline. *Italic* text marks HLF results that are further away than the baseline to the target.

4.2 Measurements

Because HLFs are just real numbers highlighting characteristics of text, they can be used for setting goals to aim for (‘write a 100 word paragraph’) and for describing text attributes (‘this paragraph has 100 words’). In our experiments we set goals for the chosen LLMs using instruction prompts and compute the corresponding HLFs on the generated text in order to measure LLM efficacy.

As outlined in Subsection 3.1, our experiment compares a baseline with the corresponding HLF results. Remember that in the context of a scenario variant, both the baseline and the HLF results for a specific HLF are vectors containing 10 real numbers.

The Kolmogorov-Smirnov test (Kolmogorov, 1933; Smirnov, 1939) is useful for determining whether the baseline and HLF results vectors differ significantly. We use the Scipy (Virtanen et al., 2020) implementation of the test. In our experiment, the vectors are significantly different for a $p\_value \leq 0.05$.

For each HLF, the target HLF value is repeated 10 times to construct a vector which is compatible with the baselines and HLF results. This allows us to compute the Euclidian norm of the vectorial difference between the baselines and HLF results on one hand and the target vector, on the other. A similar computation can be performed based on the area between each of the vectors and the target. Because the area measurements are correlated with the Euclidian norm, we present results using only the Euclidian norm.

Note that each HLF can have a different range of values than other HLFs. While computing the Euclidian norm using Numpy (Harris et al., 2020), we use the raw values of each HLF, without preprocessing. Therefore the norm computed for one HLF does not imply anything in relation to another HLF or LLM.

Let us denote with N_b and N_h the Euclidian norms for the baseline and the HLF results, respectively. We denote the difference between the norms with $Diff_N = N_b - N_h$.

The result tables show $Diff_N$ for each LLM in the context of a prompt and a text corpus. The paper is focused on some key results observed on the CNN article corpus. Additional results are available online.

4.3 HLF Instruction Impact

Table 4 shows that LLMs take into account HLF instructions when running the experiment. On the CNN article corpus, the HLF instructions lead to HLF results that are closer than the baseline to the target HLF. On the Yelp dataset, the HLF instructions generate results that are further away than the baseline from the target HLF.

Figure 1 shows a positive result, where the result obtained with the aid of HLF instructions is closer to the target. Figure 2 shows a negative result, where the baseline is closer to the target. Figure 3 shows an inconclusive result.

4.4 HLF Instructions and Context Awareness

The second scenario of our experiment uses examples to allow the LLM to generate better baselines. As pointed out at the end of Subsection 3.1, we assume that the difference we measure between baselines and HLF results is due to the HLF instructions in the system prompt.

The first variant of the second scenario involves using input text from outside the corpus. Table 5 contains the measurements we obtained in this context. We notice the HLF results tend to be closer to the target than the baseline results. In terms of significance and efficacy of the HLF instructions, the results are similar to the ones obtained in the first scenario.

We should note that there are fewer significant differences between baseline and HLF results. This seems to validate our assumption that using examples in the system prompt causes the baselines to be closer to the target HLF value.

There are exceptions to the general trend observed for the first scenario. One such exception is highlighted in Figure 4. In figs. 5 and 6 we even observe regressions in performance with significant differences between baseline and result. Even so, these are exceptions and the results obtained in our first scenario are largely confirmed on the CNN article corpus.

Finally, the second variant of the second scenario investigates the suggestion power of HLF instructions. By using input from the corpus, the LLM should produce baselines that comply even more to the target HLFs statistics computed on the corpus. We expect fewer significant differences between baselines and HLF results. Additionally, HLF results should be closer to the target than in our previous observations.

The results shown in Table 6 contradict these expectations. The most surprising behaviour is exhibited for HLFs which are derived into more concrete generation instructions, such as the total number of words (Figure 8). Additionally, we notice regressive behaviour for some models compared to the previously explored results in Figure 7. Finally, we notice that the abstractness of the text generation instructions for certain HLFs leads to non-compliance in Figure 9.

The overall sentiment, taking into account the additional results obtained on the Yelp dataset, is that LLMs do not take much advantage of the examples from text corpora, nor of the input from the corpora for the tasks performed in this experiment.

5 DISCUSSION

In terms of our first question — whether HLF instructions impact an LLM's writing, the response is affirmative. Not all LLMs are equally susceptible to HLF instructions, though. The experiment design involves choices that rely on the presence of a corpus and the usage of a fixed external input text. This is relevant as evidentiated by the difference in HLF results obtained using the Yelp reviews corpus when compared to the HLF results obtained on the CNN article corpus. While similarly significant, the HLF results on Yelp reviews are mostly worse than their corresponding baselines failing to confirm the positive impact of HLF instructions.

The impact HLF instructions have on LLM output is limited. Our metrics do not show a consistent desired impact on the LLM output either. There isn't conclusive evidence that if we improve the LLM's chances of producing a better baseline, this will result in closer HLF results to the desired target values. In fact, there is some evidence that HLF instructions have an adverse effect when examining the HLF results obtained on the Yelp reviews dataset. This is especially true of highly abstract HLF instructions.

The choice of input text in relation to the target HLF values is consequential. LLMs don't yet seem able to cover the gulf between Oscar Wilde poems and the average Yelp review in terms of linguistic features. We surmise that even if LLMs are able to generate text that complies to certain target HLFs, there is a limit to this ability.

6 LIMITATIONS AND FUTURE WORK

Other experiments are required to better understand the capabilities of large language models with regard to HLFs. The selection of language models, the reliance

Table 4: $Diff_N$ on the CNN-DailyMail corpus. Scenario 1.

model	t_word	t_sent	n_uverb	n_uadj	simp_ttr	a_verb_pw	corr_adj_var	corr_verb_var	fkgl	a_kup_pw
claude3	**623.16**	**24.74**	**51.18**	**26.68**	0.06	-0.0	**0.46**	**1.4**	1.44	***-1.66***
gemini	**985.86**	**34.52**	**100.36**	**53.39**	**0.52**	0.05	**2.62**	**4.68**	-4.17	***-2.43***
gpt	**125.87**	1.46	10.95	10.77	0.02	-0.01	0.71	0.81	***-1.72***	0.43
command_r	**1024.78**	**39.67**	**81.58**	**41.46**	**0.46**	**0.03**	**2.67**	**3.78**	1.4	**0.14**
llama3_70b	**459.45**	**17.78**	**36.5**	**19.75**	**0.12**	**0.03**	**1.43**	**2.18**	1.5	***-0.06***

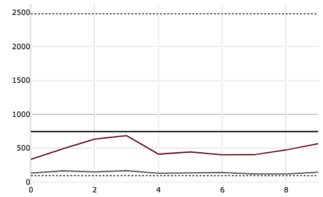

Figure 1: Command-R+ `t_word`. CNN-DailyMail, Scenario 1.

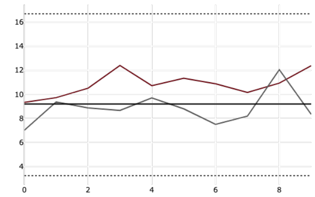

Figure 2: GPT `fkgl`. CNN-DailyMail, Scenario 1.

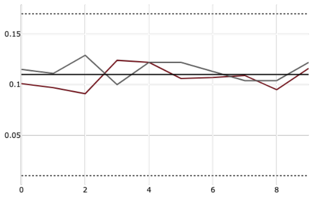

Figure 3: Claude3 `a_verb_pw`. CNN-DailyMail, Scenario 1.

Table 5: $Diff_N$ on the CNN-DailyMail corpus. Scenario 2, Input Outside Corpus.

model	t_word	t_sent	n_uverb	n_uadj	simp_ttr	a_verb_pw	corr_adj_var	corr_verb_var	fkgl	a_kup_pw
claude3	**718.49**	**24.13**	**44.43**	**23.43**	**0.4**	0.03	0.94	**0.77**	1.2	0.07
gemini	**785.6**	**25.8**	**60.5**	**41.6**	**0.41**	**0.05**	**1.95**	**2.67**	-4.62	***-1.94***
gpt	47.8	5.18	6.13	-0.13	0.02	0.0	0.05	0.3	3.06	-0.27
command_r	**263.51**	5.35	**25.08**	**17.37**	***-0.09***	-0.01	**1.74**	**1.42**	**10.46**	**1.56**
llama3_70b	**376.0**	**15.63**	**38.63**	15.74	**0.24**	**0.06**	1.69	**2.91**	0.31	-0.3

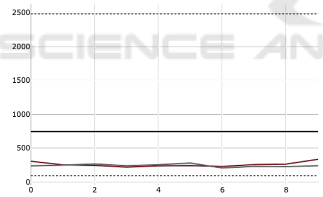

Figure 4: GPT `t_word`. CNN-DailyMail, Scenario 2, Input Outside Corpus.

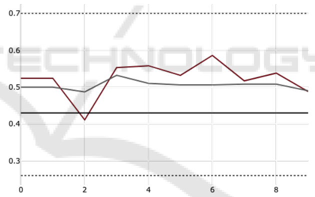

Figure 5: Command-R+ `simp_ttr`. CNN-DailyMail, Scenario 2, Input Outside Corpus.

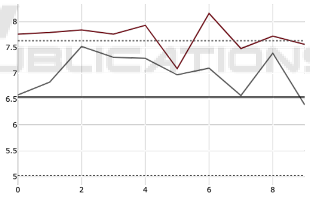

Figure 6: Gemini `a_kup_pw`. CNN-DailyMail, Scenario 2, Input Outside Corpus.

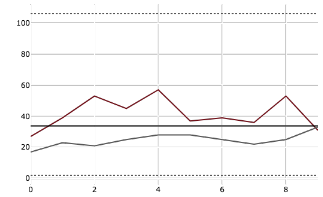

Figure 7: Claude3 `n_uadj`. CNN-DailyMail, Scenario 2, Input Inside Corpus.

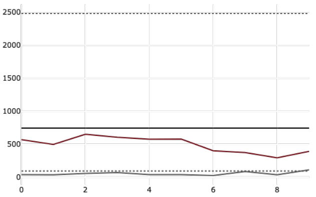

Figure 8: Gemini `t_word`. CNN-DailyMail, Scenario 2, Input Inside Corpus.

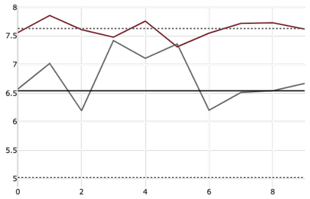

Figure 9: Llama3 `a_kup_pw`. CNN-DailyMail, Scenario 2, Input Inside Corpus.

on a corpus for computing target linguistic features, the fixed choice of input text and not least of all the wording of the system prompts are all limitations of the current approach.

Making different experiment design choices in all these respects might yield more positive and more powerful results. Using a selection of HLFs and not individually analysing the behaviour of the LLM un-

Table 6: $Diff_N$ on CNN-DailyMail. Scenario 2, Input Inside Corpus.

model	t_word	t_sent	n_uverb	n_uadj	simp_ttr	a_verb_pw	corr_adj_var	corr_verb_var	fkgl	a_kup_pw
claude3	**871.02**	**34.2**	**34.48**	***-7.38***	**0.27**	0.01	***-0.93***	***-0.72***	1.16	-0.14
gemini	**1315.41**	**46.19**	**114.47**	**72.38**	**1.17**	0.03	**5.25**	**6.64**	-1.6	***-2.1***
gpt	**521.29**	**23.3**	**43.13**	19.85	**0.2**	0.02	1.08	**1.04**	5.09	0.67
command_r	**371.08**	**13.31**	22.06	**17.19**	**0.17**	**0.02**	0.88	1.09	**4.05**	-0.15
llama3_70b	**1040.85**	**41.99**	**80.58**	**49.12**	**0.55**	**0.04**	**3.99**	**5.05**	7.18	***-1.94***

der the influence of each individual HLF is another limiting choice that invites to future work on the alternative.

7 CONCLUSIONS

We designed an experiment that tries to understand whether it is possible to generate text that exhibits certain linguistic features by instructing a large language model. It turns out state of the art large language models are receptive to instructions regarding the linguistic features of the output. This is especially true for concrete instructions.

However, the outcomes are not always good. From a pragmatic standpoint, prompt engineering and a careful choice of language features and input text seem like the way to obtain desirable results. Providing examples in the input prompt does not seem to influence the HLFs of the LLM output in the expected manner. Rewording text which already exhibits the desired linguistic features can have adverse effects, too.

Finally, we've seen how we might use handcrafted linguistic features to assess LLM output. Setting up "before and after" scenarios to evaluate relative improvements of the LLM outcome in relation to the selected HLFs is one way to achieve this. With enough measurements, the relative difference between baselines and target HLFs can be quantified using geometric or statistical means.

REFERENCES

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chen, Y., Xu, B., Wang, Q., Liu, Y., and Mao, Z. (2024). Benchmarking large language models on controllable generation under diversified instructions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38:17808–17816.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J. E., and Stoica, I. (2024). Chatbot arena: An open platform for evaluating llms by human preference.

Chomsky, N. (2002). *Syntactic structures*. Mouton de Gruyter.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

DiMarco, C. and Mah, K. (1994). A model of comparative stylistics for machine translation. *Machine translation*, 9(1):21–59.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., and Leahy, C. (2020). The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Hart, M. (n.d.). Project gutenberg.

Hermann, K. M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701.

Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spacy: Industrial-strength natural language processing in python. Version: 3.7.5; Last Accessed: 2024-06-14.

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

Hu, Z., Lee, R. K.-W., Aggarwal, C. C., and Zhang, A. (2022). Text style transfer: A review and experimental evaluation. *SIGKDD Explorations Newsletter*, 24(1):14–45.

Jin, D., Jin, Z., Hu, Z., Vechtomova, O., and Mihalcea, R. (2022). Deep learning for text style transfer: A survey. *Computational Linguistics*, 48(1):155–205.

Kolmogorov, A. (1933). Sulla determinazione empirica di una legge didistribuzione. *Giorn Dell'inst Ital Degli Att*, 4:89–91.

Lee, B. W., Jang, Y. S., and Lee, J. (2021). Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lee, B. W. and Lee, J. (2023). LFTK: Handcrafted features in computational linguistics. In Kochmar, E., Burstein, J., Horbach, A., Laarmann-Quante, R., Madnani, N., Tack, A., Yaneva, V., Yuan, Z., and Zesch, T., editors, *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 1–19, Toronto, Canada. Association for Computational Linguistics. Version: 1.0.9.

Lhoest, Q., Villanova del Moral, A., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Le Scao, T., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A., and Wolf, T. (2021). Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lugea, J. and Walker, B. (2023). *Stylistics: Text, Cognition and Corpora*. Palgrave Macmillan Cham.

Lyu, Y., Liang, P. P., Pham, H., Hovy, E., Póczos, B., Salakhutdinov, R., and Morency, L.-P. (2021). StylePTB: A compositional benchmark for fine-grained controllable text style transfer. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2116–2138, Online. Association for Computational Linguistics.

Lyu, Y., Luo, T., Shi, J., Hollon, T., and Lee, H. (2023). Fine-grained text style transfer with diffusion-based language models. In Can, B., Mozes, M., Cahyawijaya, S., Saphra, N., Kassner, N., Ravfogel, S., Ravichander, A., Zhao, C., Augenstein, I., Rogers, A., Cho, K., Grefenstette, E., and Voita, L., editors, *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 65–74, Toronto, Canada. Association for Computational Linguistics.

McDonald, D. D. and Pustejovsky, J. (1985). A computational theory of prose style for natural language generation. In *Second Conference of the European Chapter of the Association for Computational Linguistics*.

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., and Gao, J. (2024). Large language models: A survey.

paperswithcode.com (2024). Sentence completion on hellaswag. accessed on 2024-05-29.

Projects, T. P. (2024). Jinja - jinja documentation (3.1.x).

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training. accessed on 2024-05-29.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. accessed on 2024-05-29.

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

Smirnov, N. V. (1939). On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Math. Univ. Moscou*, 2(2):3–14.

Sterne, L., New, J., New, M., and Ricks, C. (2003). *The Life and Opinions of Tristram Shandy, Gentleman*. Penguin classics. Penguin Books Limited.

Toshevska, M. and Gievska, S. (2022). A review of text style transfer using deep learning. *IEEE Transactions on Artificial Intelligence*, 3(5):669–684.

Verma, G. and Srinivasan, B. V. (2019). A lexical, syntactic, and semantic perspective for understanding style in text. *arXiv preprint arXiv:1909.08349*.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272. Version: 1.13.1.

Wilde, O. (1909). *Poems: With the Ballad of Reading Gaol*. Methuen & Company.

Yang, H., Zhang, Y., Xu, J., Lu, H., Heng, P.-A., and Lam, W. (2024). Unveiling the generalization power of fine-tuned large language models. In Duh, K., Gomez, H., and Bethard, S., editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 884–899, Mexico City, Mexico. Association for Computational Linguistics.

Zhang, H., Song, H., Li, S., Zhou, M., and Song, D. (2023). A survey of controllable text generation using transformer-based pre-trained language models. *ACM Comput. Surv.*, 56(3).

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2023). A survey of large language models.