Generating Academic Abstracts: Leveraging Stylistic Metrics in Numerical Format Improves Text Generation

Elena Callegari **Peter Vajdecka** **Desara Xhura**

Abstract

We present a novel approach for controlled text generation, focusing on controlling the stylistic properties of the generated text. We extract numerical metrics highlighting different stylistic properties from a reference text, and concatenate these metrics into the text input of our generation model. We test out this approach on the task of academic abstract generation given the article text as input; our goal is to generate abstracts that are more in line with the writing style of the author(s) of the paper. We show that our proposed method is successful in aligning the stylistic characteristics of the generation with those of the target text.

1 Introduction

While Natural Language Generation (NLG) has witnessed remarkable advancements, generating text outputs that align with specific writing styles or stylistic preferences remains a challenge. Unless specific choices to control the text generation are used, the generation style remains implicitly dependent on the type of data that was used for training. This can result in text generations that sound unnatural, always "sound the same", and are very different from the intended or imagined target text. As such, it would be desirable to have explicit control over certain stylistic aspects of text generation. For example, an author wishing to generate text for an academic article might wish the generated text to mirror their overall dislike for passive-voice structures, or their preference for parataxis over hypotaxis. At the same time, even if text generation could be controlled down to these fine-grained parameters, most individuals lack the linguistic background necessary to understand and select these parameters.

This paper presents a novel approach for incorporating stylistic metrics to control text generation. To ensure that the generated text comes closest to the intended style, we extract a series of stylistic metrics from a reference text that was also written by the target author. These stylistic metrics are in the form of numerical values with decimal places; we use these metrics as part of the input that we feed to our text-generation model.

The specific task on which we test this approach is academic abstract generation given the article as input. Our intent is to generate abstracts that align with the style of the rest of the article, ensuring that the generated abstract feels like a coherent extension of the rest of the paper.

1.1 Background

CGT: Controllable Text Generation (CTG) is a subfield of NLG. The advent of large language models (LLMs) such as GPT (Radford et al., 2019), T5 (Raffel et al., 2020) and BART (Lewis et al., 2019) has made it possible to generate text that is more diverse and natural. However, LLMs are essentially still black boxes, lacking interpretability: LLMs always generate text according to the latent representation of the context, making it difficult to control the generation. This has led to the rapid rise of CTG studies with transformer-based LLMs. Various methodologies have emerged in the last 3-4 years, with topic and sentiment control being particularly popular areas of application of CTG (Zhang et al., 2022)

Concerning stylistic control in particular, important mentions are Syed et al. (2020), who control for stylistic properties by fine-tuning on a target author's corpus using denoising autoencoder loss, Wang et al. (2019), who incorporate GPT-2 with a rule-based system for formality style transfer, and Singh et al. (2020), who, using reinforcement learning, attempt to induce certain lexical target-author attributes by incorporating continuous multi-dimensional lexical preferences of the target authors into the language model.

Abstract Generation: Abstract generation can be seen as a type of summarization problem, a chal-

lenge that can be approached using a variety of techniques (Altmami and Menai, 2022). Extractive summarization involves identifying key sentences or fragments in the original text and piecing them together to form a summary (Altmami and Menai, 2022). In contrast, abstractive summarization or generative summarization aims to generate novel sentences, potentially using new phrasing or condensation, to provide an overview of the content (Altmami and Menai, 2022). A hybrid approach can combine both techniques, leveraging their respective strengths (Xiao et al., 2020). Additionally, some researchers have proposed citation-based summarization, where the content of citations is used to help produce the summary (Yasunaga et al., 2019). The length and complexity of scientific articles have historically made it difficult to train abstractive summarization models only. There is a distinct lack of research on the generation of abstracts from scientific articles using abstractive summarization techniques directly.

To our knowledge, no one has yet attempted to incorporate text together with decimal numbers as the input of LLM-based summarization systems to enhance the quality of the final summary in form of an abstract. This current study seeks to bridge this gap by investigating the application and performance of such an approach.

2 Our Approach

2.1 Dataset

We downloaded the Huggingface ArXiv dataset `https://huggingface.co/datasets/scientific_papers`), which contains 215'913 scientific articles together with their respective abstract, and removed the bibliography section from each article. We then counted word-piece token length (Vaswani et al., 2017) and word length for each article, and discarded all articles with lengths above 2800 words/3650 words-piece tokens. This was to comply with restrictions on the maximum token length allowed as input given our Nvidia H100 GPUs.

The resulting dataset consists of 18'175 scientific articles and their corresponding abstract. We partitioned this dataset into a 60:20:20 split for unbiased generative model development and evaluation.

2.2 Stylistic Metrics

The style of academic papers can vary significantly depending on the author, reflecting their unique perspectives, writing habits, and disciplinary backgrounds. These stylistic differences manifest in various aspects, including sentence structure, vocabulary choices, level of formality, use of technical jargon, use of punctuation. Additionally, disciplinary variations further contribute to the diversity of academic writing styles, as different fields may have specific conventions and norms regarding writing practices.

To perfectly reproduce a given author's style, one should control for all of these factors. However, we limit ourselves to attempting to control a number of parameters, which are listed below:

1. Average, mean, max, min number of words per sentence, and st. dev. value;
2. Average, mean, max, min word length, and st. dev. value;
3. Average, mean, max, min paragraph length, and st. dev. value;
4. Frequencies of different PoS categories (nouns, verbs, adverbs, adjectives, articles);
5. Presence or absence of Oxford commas;
6. Lexical diversity;
7. Number of colons and semicolons for every 500 words;
8. Percentage of words that appear in the 2000-most-frequent English words list;
9. Proportion of sentences containing a subordinate;
10. Max number of subordinates per sentence;
11. Proportion of passive verbs over total number of verbs.

While these metrics constitute only a fraction of all possible dimensions that can be mapped, they represent a good mix of syntactic (e.g. max number of subordinates), morphological (average word length), lexical (lexical diversity) and purely stylistic (use of Oxford commas) parameters.

These metrics were extracted for the texts in our dataset, generating stylistic reports of the likes of Figure 1). As one can see, these reports included not just text, but also numbers with decimal places.

2.3 Model Selection

After careful consideration of various large language models, we ultimately settled on T5. This

Minimum words per sentence: 3.
Maximum words per sentence: 65.
Average words per sentence: 9.592105263157896.
Mean words per sentence: 9.592105263157896.
Standard deviation of words per sentence: 12.03381288254714.
NOUN Proportion: 0.2.
VERB Proportion: 0.03.
DET Proportion: 0.05.
ADJ Proportion: 0.05.
ADV Proportion: 0.02.
AUX Proportion: 0.03.
CONJ Proportion: 0.0.
Oxford commas are not used.
Lexical Diversity Index: 0.2843915343915344.
Average n. of commas per sentence: 0.7171052631578947,
Average n. of semicolons per sentence: 0.0,
Predicted n. of commas per 500 words: 36.044973544973544,
Predicted n. of semicolons per 500 words: 0.0.
The proportion of common words in the text is: 0.31746031746031744.
Average n. of sentences with subordinate clause: 0.1.
Highest number of subordinates in a sentence: 2.
Average Word Length: 3.77.
Mean Word Length: 3.77.
Maximum Word Length: 27.
Min Word Length: 1.
Standard Deviation of Word Length: 3.03.
Passive Verb Proportion: 0.0.

Figure 1: Example of Stylistic Report

is because T5 has demonstrated excellent performance on numerical reasoning tasks (Yang et al., 2021) and has shown proficiency in learning numeracy with integer numbers (Pal and Baral, 2021). However, the original T5 model was not trained to handle inputs consisting of text together with numbers with decimal places (Raffel et al., 2020). To address this limitation, we explored the possibility of using a modified version of T5 known as Flan T5, which excels in few-shot learning (Chung et al., 2022). We hypothesized that Flan T5 might offer better control over the incorporation of stylistic metrics within decimal numbers and text.

2.4 Logic & Experiments

Our goal was to assess the impact of incorporating stylistic reports, as calculated on different types of target texts (either the original abstract or the article text), on the performance of our model.

Our approach consisted of three phases. The first phase involved training the model using raw article text as input, without any stylistic report. This step served as the baseline for our experiment, allowing us to assess the performance of the model without any stylistic metrics. The model in this case relied solely on the patterns within the input text to make inferences.

After establishing a baseline, we moved to the second phase of the approach, which involved incorporating stylistic reports calculated on the original abstract of each given paper. The stylistic report and the raw text of each article were concatenated to form a new input.

The final phase of the experiment expanded on the second phase by using stylistic reports calculated on the raw text of each article (i.e. the text of the article minus abstract and bibliography) rather than on the original abstract. We hypothesized that including the stylistic report from the entire article would provide a more comprehensive representation of the article's style, as opposed to a report derived solely from the original abstract. Similar to the previous phase, the stylistic report was concatenated with the raw article text to form the input to the Flan T5 model.

2.5 Fine-tuning Flan T5 models

We employed PyTorch as the framework used for fine-tuning in parallel on three Nvidia H100 GPUs. We trained all models for 3 epochs with a learning rate of 1e-5, a batch size of 3, and using the Adam optimizer (Kingma and Ba, 2014). We set the maximum input sequence length to 4000 word-piece tokens (the stylistic report had a length of 350 tokens, leaving a maximum length of 3650 tokens for the article input) and the maximum output sequence length to 400 word-piece tokens for generated abstract. To promote diversity and exploration during training, we employed a sampling parameter set to True. To ensure reproducibility and control the randomization during training, we set the random seed to 42.

3 Results

To evaluate our experiments, we first use Rouge metrics (Lin, 2004) to determine the quality of the generated abstract by establishing a word overlap with the original abstract. We see the results in Table 1. Incorporating in the input a stylistic report calculated on the original abstract clearly improves the quality of the generations compared to the baseline, on all Rouge metrics. On the other hand, incorporating a stylistic report calculated on the article text will not improve the results over the baseline model.

We also calculated the cosine similarity between the numerical values in the stylistic reports of the generated abstracts vs. those calculated on the orig-

Model	Rouge 1 F-score	Rouge 2 F-score	Rouge l F-score	Rouge 1 P	Rouge 2 P	Rouge l P	Rouge 1 R	Rouge 2 R	Rouge l R
Abstract stylistic report + Flan T5	**0.342**	**0.128**	**0.302**	**0.430**	**0.162**	**0.379**	**0.313**	**0.120**	**0.276**
Article stylistic report + Flan T5	0.333	0.121	0.293	0.416	0.152	0.366	0.307	0.114	0.271
Baseline: Flan T5	0.335	0.123	0.294	0.420	0.155	0.369	0.306	0.115	0.269

Table 1: Rouge Metrics

Model	min similarity	max similarity	mean similarity	median similarity	std_ev similarity
Abstract stylistic report + Flan T5	0.028	**0.999**	**0.901**	**0.939**	**0.116**
Article stylistic report + Flan T5	0.042	0.999	0.885	0.934	0.138
Baseline: Flan T5	**0.045**	0.999	0.890	0.935	0.129

Table 2: Cosine similarity statistics of generated abstracts and original abstracts

stylistic vector =

$$\begin{bmatrix} 1 \\ 65 \\ 9.592105263157896 \\ 9.592105263157896 \\ 12.03381288254714 \\ 0.20 \\ 0.03 \\ 0.05 \\ 0.05 \\ 0.02 \\ 0.03 \\ 0.00 \\ 0.25 \\ 0.2843915343915344 \\ 0.7171052631578947 \\ 0.0 \\ 36.044973544973544 \\ 0.0 \\ 0.31746031746031744 \\ 0.10 \\ 2 \\ 3.77 \\ 3.77 \\ 27 \\ 1 \\ 3.03 \\ 0.00 \end{bmatrix}$$

Figure 2: Example of Stylistic Vector

inal abstracts. This was done to determine whether incorporating a stylistic report in the input to Flan T5 as a CTG technique would be effective in aligning the stylistic properties of the generated abstract to those of the original abstract. To calculate cosine similarity, we converted each stylistic report (example shown in Fig. 1) into a vector (example shown in Fig. 2). We then calculated the cosine similarity between the stylistic vectors of the generated abstract versus those of the original abstract. In Table 2 we can see the cosine similarity statistics calculated on the test dataset. The most important values are obviously the mean and the standard deviation. Here we see that the model including stylistic reports from the abstracts as input generates the most stylistically similar abstracts to the original abstracts. This model also achieves the lowest variability around the mean via standard deviation, which guarantees that the stylistic similarities are the least different from each other compared to the other models.

4 Conclusions

We proposed an approach to control the stylistic properties of text generation that consists in incorporating decimal-number stylistic metrics of a target text in the generations of Transformer-based LMs. Our results indicate that extracting stylistic metrics from the original abstract and using these as control mechanisms for the abstract generation is successful in aligning the stylistic characteristics of the generation with the target text.

We noticed a slight decrease in the Rouge and similarity scores if we calculate stylistic metrics on the full article instead. This might be due to multiple reasons, one of them being the fact that STEM articles (the ArXiv dataset consists of STEM articles) tend to have multiple authors, each of them possibly exhibiting a unique writing style. It would thus be interesting to replicate this study by only selecting single-authored articles as input text, and see whether that nullifies the difference noticed when using the original abstract vs. the article text to calculate the stylistic report. A second reason might be the fact that different sections might require a different style of writing. For instance, *Methodology* sections explaining often differ in style from *Introduction* or *Conclusions* sections. It would be interesting to delve deeper into the examination of variations in writing styles across article sections and explore whether specific stylistic patterns more effectively capture distinct sections of the text. Overall, our findings suggest that the CTG technique we have introduced in this study can assist researchers in refining specific sections of an article to align with a desired writing style. This addresses the challenge posed by articles with multiple authors who have distinct writing styles. Additionally, our methods offer the ability to finely adjust certain sections to match a target writing style.

5 Limitations

A limitation of this study is the lack of a qualitative human evaluation of the generated abstracts and their alignment to the original abstract. We have not resorted to human evaluation due to the high-level expertise required to evaluate abstracts of highly technical papers. This is an often-cited issue in NLG studies (Syed et al., 2020; Singh et al., 2020). Despite the lack of such evaluation, our results are promising and offer a plausible line of research for controlling text generation using a target text to extract the control parameters.

References

Nouf Ibrahim Altmami and Mohamed El Bachir Menai. 2022. Automatic summarization of scientific articles: A survey. *Journal of King Saud University-Computer and Information Sciences*, 34(4):1011–1028.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Kuntal Kumar Pal and Chitta Baral. 2021. Investigating numeracy learning ability of a text-to-text transfer model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3095–3101, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Hrituraj Singh, Gaurav Verma, and Balaji Vasan Srinivasan. 2020. Incorporating stylistic lexical preferences in generative language models. *arXiv preprint arXiv:2010.11553*.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2019. Harnessing pre-trained neural networks with rules for formality style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3573–3578.

Liqiang Xiao, Lu Wang, Hao He, and Yaohui Jin. 2020. Copy or rewrite: Hybrid summarization with hierarchical reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9306–9313.

Peng-Jian Yang, Ying Ting Chen, Yuechan Chen, and Daniel Cer. 2021. Nt5?! training t5 to perform numerical reasoning. *arXiv preprint arXiv:2104.07307*.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7386–7393.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. A survey of controllable text generation using transformer-based pre-trained language models. *arXiv preprint arXiv:2201.05337*.