NLG AI – Natural Language Generation in the style of Asimov Isaac

Luca Corvitto
Sapienza University of Rome
Student ID 1835668
corvitto.1835668@studenti.uniroma1.it

Abstract

Generative language models like the GPT systems by OpenAI have taken the attention of the masses all over the world. Following the trend, this paper focuses on analysing if is possible to use this tool, in particular the open source model GPT2 available on HuggingFace, to generate text in the personal writing style of the famous sci-fi author Isaac Asimov.

1 Introduction

The evolution of natural language processing (NLP) have lead to numerous advancement in text generation. One of these is the ability to mimic and transfer different **writing styles**. *A personal writing style refers to the distinctive manner in which the semantics is expressed* (McDonald and Pustejovsky, 1985). Style originates as the characteristics inherent to every person's utterance, which can be expressed through the use of certain stylistic devices such as metaphors, as well as choice of words, syntactic structures, and so on. Understanding and replicating personal writing styles can have diverse applications, ranging from literary endeavors to personalized content creation (Jin et al., 2022).

Text style transfer is one of the NLP research area that aims to automatically transform the writing style of a given text while preserving its semantic content. Researchers have explored various style transfer techniques, such as attempting to emulate renowned literary styles like Shakespearean prose (Jhamtani et al., 2017). These attempts have shown the potential for automated text generation to replicate and transfer different writing styles.

In recent studies, researchers are paying increasing attention to modeling and manipulating the style of the text within its generation; this approach is called **stylized text generation** (Mou and Vechtomova, 2020). The goal is to control both the style, like the *persona* of a speaker in a dialogue, and the content of the text.

Stylized text generation is related to various machine learning techniques, for example, embedding learning techniques to represent style (Fu et al., 2018), adversarial learning and reinforcement learning with cycle consistency to match "content" but to distinguish different styles (Hu et al., 2017; Xu et al., 2018; John et al., 2019).

The presented approach is based on the concept of style-conditioned text generation by utilizing a fine-tuned pre-trained generative model, GPT2 in this case, to generate stylized text. Fine-tuning the model on a dataset comprising texts written in the desired style, Asimov's works in this case, enables the model to learn the underlying patterns and nuances of the specific writing style while maintaining coherence and semantic fidelity.

2 Text Generation

In this section it will be presented and described the main steps done in order to perform the generation of the text in the style of Isaac Asimov.

2.1 Data Collection

The first step consists in actually collecting Isaac Asimov's works in a format that could be converted into a easily accessible format, like `.txt`.

A first attempt was to use the `.pdf` file format, but most of the pdf files have header and footer for each page that would lead to a too long preprocessing of the text, that would inevitably lead to a large number of unwanted errors in the final version of the text file. The final solution was to use the `.epub` format, that has in general lesser unnecessary text.

The final number of Asimov's books collected was equal to 28. The complete list can be found in the appendix A in Table 4.

2.2 Data Preprocessing

After the collection of the needed Asimov's works, each of them was "cleaned", in order to obtain, as a final result, a long sequence (max length equal

Baseline Output
Alan woke up the next morning and found her in a car with her brother and three other friends. "I was absolutely horrified and it's just really upsetting," she said. "It's a huge shame because I've had a great life. "I'm going to be on a mission to help those kids."
GAI Output
Alan woke up to find himself face to face with a wall of roboticized debris. "I don't know what you are thinking, Mr. R. Daneel," he said, "but you know what you are thinking. The other side of the planet is still being habitable. You know it.
Enlarged GAI Output
Alan woke up with a vague feeling of euphoria. The First Minister was, of course, a bachelor. His hair was dark and thin, his eyes were fixed, his face uncertain, his eyes still looking through the window. He was wearing a plain white suit, a dark red shirt, a dark gray trousers, a dark blue shirt, a dark gray tie.

Table 1: Cherry picked examples of the outputs generated by the different models given the initial prompt as: "Alan woke up". GAI, in general, results more sci-fi oriented, while the enlarged version has a wider and more heterogeneous kind of contents.

to 512) of text in the writing style of the author. The "cleaning", or denoising, procedure consisted in the manual inspection of each `.txt` file in order to remove publishing information, prefaces written by other authors and chapter numbers or titles and all the superfluous information that could affect negatively the fine-tuning.

After that, each file was divided into sentences using the sentence tokenizer from the nltk package (Bird and Ewan, 2009) and then merged into a single `.txt` file.

The final step consisted in the tokenization of the sequence of sentences. For the GPT2 model small version, the maximum length of tokens in input is equal to 768, so the sentences were first grouped in a sequence structured as it follows:

"<|startoftext|>Lije Baley had just reached his desk when...<|endoftext|>".

then this sequence was tokenized and finally, to cope with the size of the GPT2 model, the tensors were packed to the maximum possible length.

2.3 Model Specifications

In this work it is used the GPT2 model (Radford et al., 2019) from Huggingface by OpenAI. The list of the hyper-parameters used is shown in the appendix A in Table 5.

For the generation task different approaches were tried and so they will be briefly described and presented here.

Temperature Sampling (Ackley et al., 1985) consists in performing *Random Sampling* (Singh, 2003) using temperature T as a value to increase the probability of probable tokens while reducing the one of tokens less probable. $T = 0$ is equivalent to $\frac{argmax}{maxlikelihood}$, while $T = \infty$ corresponds to a Random Sampling. This approach was not directly tasted but was implemented in many of the following ones.

Top-p Sampling, or Nucleus Sampling (Holtzman et al., 2020), compute and sort in descending order all the probabilities that are then summed up until this total sum, called Cumulative Distribution Function (CDF), is above an adjustable hyperparameter p. Once the CDF is formed, everything below p is eliminated by setting it to $-\infty$.

Temperature Top-p Sampling is a variation of *Top-p Sampling* in which the probabilities, before being sorted, are modified by the Temperature value T.

Sample-and-Rank (Adiwardana et al., 2020) is another approach composed by two main steps: First, it samples N independent candidate responses using *Temperature Sampling*. Then it selects the candidate response with the highest probability to use as the final output.
The final attempt was to try both the approaches together.

Top-p Sample-and-Rank shares the first step with *Sample-and-Rank*, but then uses *Top-p sampling* to selects the final output.

After different trials *Temperature Top-p Sampling* appeared to be the most efficient approach in this case, since it was faster and the quality (human evaluated) of the chosen samples was similar and sometimes higher than the other ones.

Performance	Test Set	Generated Text
Accuracy	0.996	0.799
Precision	0.992	0.963
Recall	1.000	0.733
F1 score	0.996	0.832

Table 2: Performance of the classifier on real sentences extracted from different books and on the generated text in the First Trial

Performance	Test Set	Generated Text
Accuracy	0.976	0.693
Precision	0.956	0.849
Recall	0.999	0.696
F1 score	0.977	0.765

Table 3: Performance of the classifier on real sentences extracted from different books and on the generated text in the Second Trial

3 Evaluation and Results

In this section it will be shown how the produced output was evaluated and the results obtained.

3.1 Metric Selection

To verify if the underlying patterns and nuances of the specific writing style were apprehended by the model, the metrics that are commonly used in NLG, like BLEU (Papineni et al., 2002) or ROUGE (Lin, 2004), can not be used here since they measure just the content. NLG metrics that are also used in TST are Perplexity or Cosine Similarity (Jin et al., 2022), but novel metrics that try to disentangle content and style were created for TST tasks like Style Transfer Intensity (STI) and Content Preservation (CP) (Mir et al., 2019). Unfortunately, since these brand new metrics are designed for TST tasks, that take as input both the target style attribute a and a source sentence x that constrains the content (Jin et al., 2022), they can not be used in this case. So, as it is commonly done in the literature (Toshevska and Gievska, 2022), it is performed a *fine-tuning* of a **pre-trained transformer** based classifier to distinguish between two classes: Asimov and Non Asimov.

3.2 First Trial: GAI - Sci-fi Fictions only

The first training data used for the GPT fine-tuning contained 19 books, resulting in a number of words equal to 1894032, all of them concerning science fiction. An example of the outputs obtained in this trial are shown in Table 1

In order to evaluate the generated texts the Asimov dataset was expanded adding the same number of books from other authors, with a low percentage about science fiction to mitigate the bias, and it was used to fine-tune the transformer classifier. The obtained results are shown in Table 2.

As it can be seen, the model's outputs, being generated from a sci-fi fine-tuning, are sci-fi oriented, and in the same way the classifier is biased to detect science fiction more than the Asimov's writing style itself. Due to these limitations, another trial was made.

3.3 Second Trial: eGAI - enlarged Dataset

In this second trial the model was fine-tuned using a larger dataset, containing, in addition to the Asimov sci-fi books, also some oh his other works, like science essays and novels of other genres, leading to a total of 28 books, equal to 2376380 words. In parallel, also the *non Asimov* dataset was updated adding more books in order to have a complete balanced dataset for the classifier.

The new outputs and evaluation results are shown respectively in Tables 1 and 3. Here we can see that the performances of the new model (3) seem overall lower than the first one (2). This is due to the fact that the new one is more heterogeneous and it can no more focus just on the content, in fact also the performance on the test set are lower. Emulate, and also detect, the writing style is a difficult task and so both Precision and Recall decreased, due to the similar content in the outputs.

So, getting rid of the sci-fi bias, the model meet its limitation in the emulation and the detection of the writing style, lowering the performances.

3.4 Human Evaluation

Due to the limitations of the metrics a human evaluation was also performed.

This evaluation was performed reading some extracts both from the baseline and the fine-tuned model, first and second version.

The GAI and eGAI outputs resulted in more concise sentences, in line with Asimov's writing style. The limitation of the GPT2 model, however, lead sometimes to nonsense due to the change of the subject's name or due to repeated sentences, both in the baseline and in the fine-tuned outputs. This could have negatively affected the success of the task.

4 Conclusion and Future Works

In this work was presented and analyzed a simple and novel approach in the stylized text generation by the fine-tuning of an existing large language model (GPT2) on the writing style of a specific author (Isaac Asimov). The paper analyzed the overall process from the data collection and preprocessing to the text generation and showed the potential and the limitations of this approach using a transformer classifier for the evaluation of the task.

From the analyses resulted that this kind of task is complex to perform for LLMs such as GPT2 (more powerful LLMs like GPT4, LLaMa or Bard could reach better results) but the real complexity is in the detection of the *generated* writing style since there is no suitable metrics to use. The Transformer Classifier used is limited since it should be fine-tuned as well and there is no guarantee that, even if the LLM should manage to emulate perfectly the writing style, the classifier could detect it.

References

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169.

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot.

Edward Loper Bird, Steven and Klein O'Reilly Media Inc Ewan. 2009. Natural language processing with python.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

Harsh Jhamtani, Varun Gangal, Eduard H. Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *CoRR*, abs/1707.01161.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep Learning for Text Style Transfer: A Survey. *Computational Linguistics*, 48(1):155–205.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Florence, Italy. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

David D. McDonald and James D. Pustejovsky. 1985. A computational theory of prose style for natural language generation. In *Second Conference of the European Chapter of the Association for Computational Linguistics*, Geneva, Switzerland. Association for Computational Linguistics.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating style transfer for text. *CoRR*, abs/1904.02295.

Lili Mou and Olga Vechtomova. 2020. Stylized text generation: Approaches and applications. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 19–22, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Sarjinder Singh. 2003. *Simple Random Sampling*, pages 71–136. Springer Netherlands, Dordrecht.

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

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988, Melbourne, Australia. Association for Computational Linguistics.

A Appendix

Asimov	Non Asimov
The Caves of Steel	*The Count of Monte Cristo*
The Naked Sun	*A clockwork Orange*
The Robots of Dawn	*A Space Odissey*
Robots and Empire	*Count Zero*
Pebble in the Sky	*The Da Vinci Code*
The Stars, Like Dust	*Cop Hater*
The Currents of Space	*1984*
Prelude to Foundation	*Dune*
Forward the Foundation	*Mission of Gravity*
Foundation	*Norwegian Wood*
Foundation and Empire	*Ulysses*
Second Foundation	*Harry Potter and the Philosopher's Stone*
Foundation's Edge	*The Lord of the Rings*
Foundation and Earth	*Animal Farm*
The End of Eternity	*The Picture of Dorian Gray*
The Gods Themselves	*Ubik*
Fantastic Voyage	*Do Android Dreams of Electric Sheep*
Fantastic Voyage II: Destination Brain	*Neuromancer*
Nemesis	*Mona Lisa Overdrive*
Of Time and Space and Other things	*Hyperion*
Puzzles of the Black Widowers	*Astrophysics for People in a Hurry*
More Tales of the Black Widowers	*American Prison: A Reporter Undercover*
Murder at the ABA: A Puzzle in Four	*Theory of Everything*
Atom: A Journey Across the Subatomic Cosmos	—
The Planet That Wasn't	—
Whiff of Death	—
Isaac Asimov's Guide to Earth and Space	—
View From a Eight	—

Table 4: Complete list of the books collected in the dataset. Before the horizzontal line there are the books used for the first version of the dataset, after that there are the other non sci-fi books added in the enlarged version.

Hyper-parameter	Value
Fine-Tune	
Model	GPT2
Batch Size	16
Learning Rate	$2e^{-5}$
Epochs	20
Optimizer	AdamW

Hyper-parameter	Value
Classifier	
Model	bert-base-case
Batch Size	2
Learning Rate	$1e^{-6}$
Epochs	10
Optimizer	Adam

Hyper-parameter	Value
Generation	
Entry Count	10
Entry Length	512
Top P	$[0.7, 0.8, 0.9]$
Temperature	$[0.88, 1.0]$
Number of Samples	$[20, 50]$

Table 5: Values of the hyper-parameters used for both training and generation of text. For the generation task all the values were used in combination.