

Zero-Shot Metrical Poetry Generation with Open Language Models: a Quantitative Analysis

Manex Agirrezabal

Centre for Language Technology (CST)
University of Copenhagen, Denmark
manex.agirrezabal@hum.ku.dk

Hugo Gonalo Oliveira

CISUC, LASI, DEI
University of Coimbra, Portugal
hroliv@dei.uc.pt

Abstract

Poetry generation, a long endeavor in Computational Creativity, can nowadays be tackled by prompting an LLM. From a set of natural language instructions, which may cover semantic and formal constraints, the result is generally a piece of text that one would recognize as poetry, but where metrical constraints are not always met. To confirm this empirically and take conclusions on its extent, we measure how well generated poems match simple constraints such as the number of lines, stanzas, or syllables per line. Specifically, we prompt the open LLM Llama2 with instructions of variable complexity; we compute the previous constraints in generated poems using an automatic tool; and we analyse how much they deviate from the given instructions. We conclude that, in a simple prompting scenario, the general purpose Llama2 model is unlikely to produce well-metered text.

Introduction

Natural Language Processing (NLP) has seen significant progress with the development of Large Language Models (LLMs) that can be applied to a variety of tasks. Many of which can be performed in a zero- or few-shot mode, simply by prompting the model with a natural language instruction, without any fine-tuning. This also applies to the generation of linguistic creative artefacts, such as poetry, a common task in the domain of Computational Creativity (CC). Many of us have tried LLMs to generate poetic texts, and, at first glance, results are promising. Generated text uses poetic language, rhymes are common, and its form generally resembles a poem. We asked the open LLM Llama2 (7b) to produce a short poem about the ICC conference, and this is one of the produced quatrains (out of four):¹

*In Sweden’s heartland, where tech meets art,
A gathering of minds, to spark new starts.
From code to canvas, from machine to mind,
Creativity blooms, as innovators unwind.*

When instructed to generate a poem on a topic, this is also reflected on the result (Kirmani 2023). However, one crucial aspect that remains to be explored is the model’s ability to generate metrical poetry in this simple setting. Meter is a fundamental element of poetic structure, and understanding whether these models can produce metrically-correct poems

is essential for their potential use in automatic poetry generation or co-creative poetry writing.

To further investigate this, we made a series of experiments with Llama2 (Touvron et al. 2023). The model was prompted with various instructions for the generation of metrical poetry, covering the following features: number of stanzas, number of lines, and number of syllables. To quantify how well such instructions were followed, produced outputs were analysed with Erato (Agirrezabal, Gonalo Oliveira, and Ormazabal 2023), a tool for the automatic evaluation of poetry.

In contrast to subjective semantic constraints, the aforementioned formal features can be measured automatically, providing a more objective assessment of the models’ performance. By exploring these questions, we hope to shed light on the potential of open generative models for poetry writing and their ability to capture the complexities of meter in poetry. As a way to further analyze differences in poems generated under different scenarios, we also employed less strict metrics, such as rhyme richness, as defined by Plecháč (2018), and overlap, given by ROUGE metrics (Lin and Och 2004). All of these metrics are already included in Erato.

We contribute with an analysis of the usefulness of open LLMs to produce metrical poetry as they are. Many authors fine-tune models, assuming that regular LLMs are not able to produce it, so we take a step back to answer whether this is true or not. Moreover, we explore the effect of commonness (using frequent or infrequent meters), the size of the model, or the specification of a topic. We conclude that, despite the flexibility of the tested LLM, it struggles to follow metrical constraints, especially when they are not the most common in poetry. We recall that, due to their objectivity and ease of formalization, these specific constraints were rather straightforward to implement in traditional poetry generators, in opposition to features like fluency or semantics.

In the remainder of the paper, we overview related work on poetry generation and analysis; we describe the experimentation methodology; we present the quantitative results obtained; We conclude with a brief discussion and possible research directions.

Related work

Poetry generation has long been popular in CC, with many proposed systems, following various approaches and con-

¹We include the whole interaction and poem in Appendix 1.

sidering different features (Gonalo Oliveira 2017; Lamb, Brown, and Clarke 2017).

As it happened to other NLP and text generation tasks, current approaches for poetry generation are based on the Transformer architecture (Vaswani et al. 2017), mostly on LLMs (Elzohbi and Zhao 2023). Previous work (Sawicki et al. 2023b) attempted to directly prompt GPT-3 for generating poetry in the style of a specific author, without much success. A solution was to fine-tune the same GPT-3 model for the same task (Sawicki et al. 2023a). Generative chat models like ChatGPT have been creating content that is similar to famous poets such as Keats (Hutson and Schnellmann 2023). Analysis revealed that these models can mimic the vocabulary, language choices, style, and even rhythm of classic poets with uncanny accuracy. At the same time, they tend to struggle when it comes to capturing the emotional nuances and subtleties of poetry.

Orekhov and Fischer (2020) trained different Recurrent Neural Networks (RNN) at the character level to produce Russian poetry that followed the style of a certain author. In some generations, the authors observed similarities with the original style of the poets, but in other cases, probably because of its complexity, RNNs were not able to learn specific patterns. They claim that artificial neural networks act as a kind of digital echo chamber of literary history, as they reproduce parts of the stylistic features of a training sample.

In an attempt to gain a deeper understanding of ChatGPT’s capabilities, a hermeneutic approach was employed and close readings of its responses were conducted (Henrickson and Merono-Penuela 2023). Although it is unclear if the findings are applicable to Llama, the experiment represents an interesting exploration of the abilities of these models. Our own study follows a similar path, but with a more focused scope and a smaller scale.

Method

Our main goal is to assess whether open LLMs can be used for generating poetry with predefined metrical constraints, in a zero-shot scenario, without chain of thought or further fine-tuning. In other words, we analyse to what extent we can control the meter of generated text with a single prompt.

Tools

Experimentation was conducted with quantized versions of the Llama2 (Touvron et al. 2023) open language model, namely the 7B and 70B parameters models, both accessed through ollama². Poems were generated by prompting the model in a zero-shot scenario.

We further employ Erato (Agirrezabal, Gonalo Oliveira, and Ormazabal 2023) for analysing the generated text. Specifically, we used the version available from its public repository³. For counting syllables, each word is retrieved from the CMU dictionary, and syllables are counted from the first found instance of that word. For the analysis of rhymes, Erato relies on RhymeTagger⁴ (Plechc 2018). We

²<https://ollama.ai/>

³<https://github.com/manexagirrezabal/erato>

⁴<https://github.com/versotym/rhymeTagger/>

employ rhyme richness, which is a ratio of the number of rhyming lines to the total number of lines in the poem. For novelty, Erato employs the ROUGE overlap metric (Lin and Och 2004), similarly to Gonalo Oliveira et al. (2017). For computing the overlap across poems, we represent each poem as a single string. Other methods are introduced by Gonalo Oliveira et al. (2017), but we judged this to be the most appropriate for the current work.

Features

We focus on three main poetic features: the number of lines, the number of stanzas, and the number of syllables. As an attempt to cover various metrical structures, we consider different numbers of each. Having in mind that LLMs are trained in large quantities of data, it is expected that generated poems follow some trends, and that some values are easier to meet than others. Therefore, we consider both more typical values, common in popular structures, but also more uncommon. Specifically, we tested the following simple values (i.e., simple structures):

- Number of lines: 4, 7, 8, 14
- Number of stanzas: 1, 2, 4

Towards more complex structures, we also test the following combinations:

- 4 lines with 8, 10 or 13 syllables each
- 4 stanzas with 3 or 4 lines each
- 3 stanzas with 4 lines each and a last one with 2 lines

In total, we consider 13 different metrical structures, some more common than others.

Even though we do not assess semantics, we also analyse the impact of additionally asking for a topic. For this, we consider 10 variations of each of the previous structures, one without a topic, and nine more, with each of the following nine topics, three of which are more common in poetry, three more abstract, and three concrete words: love, nature, death, instance, attention, year, cell, supermarket, baseball.

This results in 130 configurations (13×10) overall. For each, 10 poems are generated by each model.

Prompts

In order to get poems for each configuration, prompts were based on the following template:

- Please write me a poem with [N lines/M stanzas] (with X syllables each) (about TOPIC)

For instance, for a poem about ‘nature’ with four eight-syllable lines, the following prompt was used: “*Please write me a poem with 4 lines with 8 syllables each about nature*”.

Since the tested models were originally trained for chat interactions, in most cases, generated text did include more than just the poem, thus requiring post-processing. To get just the poem, we removed lines starting with “Certainly!” or “Sure!”. In some cases, further instructions and their responses were also included, and we removed them all. Finally, we deleted the number of syllables when explicitly mentioned (e.g., “(8 syllables)”).

	4*		4		7		8		14	
	Acc.	MAE	Acc.	MAE	Acc.	MAE	Acc.	MAE	Acc.	MAE
7b	100.0	0.00	76.67	0.47	0.00	1.44	26.67	1.49	0.00	5.58
70b	95.93	0.11	90.00	0.10	21.11	0.79	88.89	0.11	33.33	2.74

Table 1: Assessment of the number of lines produced by Llama2 (7b or 70b). 4* indicates four lines and a specific number of syllables (i.e., complex structures).

Interaction with the LLaMa model

For each configuration, we produce 10 poems, to assess the variability of the responses. In order to obtain a different poem every time, we interact with Llama2 through its API, ask for the generation of chat completion, and save the previous responses. Intuitively, it is the same as inputting the prompt, getting an answer, and writing the prompt in the chat again (and so on until producing 10 poems).

The API enables to generate completion given a prompt and a context (without chat). Yet, when we employed this, it showed a very high overlap among responses. We tried to change the temperature parameter, but more parameters would require our attention (e.g., `top_p`, `top_k`, `repeat_penalty`), thus adding complexity to the experimental setup which, at this stage, was not desirable.

Measuring Deviation

In order to measure how much the produced poems deviated from the given instruction, we adopted two measures: accuracy, i.e., the proportion of poems/lines that match the target constraints; the and mean absolute error (MAE) between target and actual values computed from the lines/poem.

Results and Discussion

Our methodology resulted in a dataset of 2,600 automatically generated poems, 1,300 for each Llama2 model. Upon acceptance, these poems, together with their analyses, in JSON format, and the full set of prompts, in a spreadsheet, will be made publicly available. Here, we present and discuss the results for the different studied aspects.

Table 1 has the metrics computed for poems produced when the prompt refers a specific number of lines. With higher accuracy and lower MAE, the 70b model matches the target number of lines more consistently for all non-complex cases (4, 7, 8, 14 lines). 7 and 14 are the numbers for which both models struggle more, with the 7b model not producing a single poem with the right number of lines. Yet, when the prompt includes the number of syllables (4*), the 7b model performs slightly better. This is also the case for structures with more than one stanza. In Table 2, we assess the number of lines that each stanza had, i.e., when instructed to produce three stanzas of four lines, we checked whether each of the stanzas had indeed four lines. This is why MAE is lower than for 14 lines. In all cases, the 70b model shows a higher accuracy, but, curiously, also a higher MAE. This means that it produces more correct poems but also that, when the result is not correct, the difference between the target value and the actual value is larger.

	4×4		4×3		4×3+2	
	Acc.	MAE	Acc.	MAE	Acc.	MAE
7b	88.89	0.11	40.00	0.60	72.78	0.54
70b	94.44	0.17	74.44	0.71	74.44	0.71

Table 2: Assessment of the number of lines produced by Llama2 (7b or 70b), considering complex structures, where we check the number of lines in each stanza.

Table 3 is on the number of stanzas⁵, where both models seem to perform generally well when the number of syllables is not included in the prompt (4*). When this happens, the most accurate model is again not the one with lower MAE, which suggest that the 7b model makes more but smaller mistakes in this case.

Table 4 is on the number of syllables per line. Results show that this constraint is matched poorly by both models, especially when they are prompted to produce 13 syllables, the highest number, which is also uncommon in poetry. Out of curiosity, when, in further interactions, we asked the model for a number of lines with 7 syllables, in many occasions, it would produce a block of lines that, regardless of their actual number of syllables, would include “(7 syllables)” at the end of each. This shows that LLMs do not understand what the number of syllables means as such. Please find such interaction in Appendix 2.

Even though none of the prompts mentions rhymes, we decided to look at the rhyme richness, to conclude that it is very similar for both models, specifically, 0.76 for the 7b and 0.78 for the 70b model. This means that generated poems generally capture this common poetic feature quite well. We further checked if including a topic in the prompt had an impact on the presence of rhymes, to conclude that the average rhyme richness was between 0.67 (‘death’, with the 7b model) and 0.85 (‘nature’, with both models).

The last analysis is on content overlap in the generated poems, based on ROUGE (Lin and Och 2004) scores, which can be seen as inversely proportional to novelty (Gonçalo Oliveira et al. 2017), i.e., the higher the score, the higher the overlap, thus, the lower the novelty. Intra ROUGE measures the overlap in lines of the same poem, whereas inter ROUGE computes the overlap between different poems generated with the same prompt (i.e., structure+topic). We first concluded that including a topic in the prompt does not seem to affect the intra novelty. On the other hand, as expected, the inter ROUGE scores in Ta-

⁵We define a text to have N stanzas if there are N groups of lines divided by, at least, two newline characters

	1		2		4		4*	
	Acc.	MAE	Acc.	MAE	Acc.	MAE	Acc.	MAE
7b	97.8	0.02	100.0	0.00	90.0	0.33	55.6	0.44
70b	77.8	0.22	88.9	0.11	100.0	0.00	74.1	0.96

Table 3: Assessment of the number of stanzas produced by Llama2 (7b or 70b). 4* indicates that the model was asked to produce four stanzas with further constrains on the number of syllables. For a fair comparison, we separate them as in Table 1.

	8		10		13	
	Acc.	MAE	Acc.	MAE	Acc.	MAE
7b	21.11	1.43	8.89	1.96	0.00	5.01
70b	26.49	1.48	16.84	1.74	0.83	4.32

Table 4: Assessment of the number of syllables in poems produced by Llama2 (7b or 70b).

ble 5 show that different poems generated for a given topic do have more overlap than the same number of poems generated without a topic. A surprising insight is that, when the topic is not specified, both the average intra and inter ROUGE are higher for the 70b model, suggesting that poems by the larger model are less novel.

	7b		70b	
	NoTopic	Topic	NoTopic	Topic
ROUGE-1	0.39	0.69	0.45	0.60
ROUGE-2	0.17	0.55	0.27	0.43
ROUGE-3	0.13	0.49	0.23	0.36
ROUGE-4	0.11	0.45	0.21	0.32
ROUGE-LCS	0.32	0.64	0.40	0.55
ROUGE-SU4	0.21	0.55	0.30	0.44

Table 5: Inter ROUGE scores of poems generated by Llama2 (7b or 70b). Scores were calculated considering each poem as a single string and comparing it to all poems created with the same prompts (population of 10 poems).

Conclusion

A quantitative analysis of poems by two versions of the open LLM Llama suggests that, in a zero-shot scenario, these models are not apt for producing text that follows given metrical constraints, especially in what concerns the number of syllables. Generations are typically aligned with a popular response, i.e., for typical metres, models can guess the right number of lines or syllables, but they struggle to match the right number in less popular. As pointed out by Bender et al. (2021), these models have observed a vast amount of linguistic forms, which they combine probabilistically, without any reference to meaning.

It is known that instructions including numbers are usually not followed strictly, something that was called the *counting level effect* (Yao et al. 2023), while observing that the task of counting characters within a word is still easier than doing it within a sentence. Therefore, if the output is expected to follow some specific metrical con-

straints, a general purpose open LLM is unlikely to produce metrically-correct content described in a single prompt. It is interesting to note that, in traditional approaches to poetry generation (Gonçalo Oliveira 2017), metre was typically matched through simple counting approaches, and one of the main challenges was, instead, the generation of fluent and semantically-coherent text.

If enough data is available, fixing this issue in LLMs currently requires fine-tuning the model for poetry generation (Sawicki et al. 2023a). Alternatively, more complex interaction, with several turns, can be explored, e.g., with chain-of-thought prompting (Wei et al. 2022). Due to reported limitations of the studied LLMs, self-feedback (Madaan et al. 2024) would probably not be an option. On the other hand, a promising use case for Erato (Agirrezabal, Gonçalo Oliveira, and Ormazabal 2023) would be providing human-interpretable feedback to the model. Given that syllable-count can be limited by the process of splitting textual sequences, i.e., the underlying tokenizer, recent work for Chinese has adopted a token-free with some success on this matter (Yu et al. 2024).

We plan to continue studying poems produced by LLMs and how well they follow given instructions, regarding different perspectives. Whenever possible, we aim to take advantage of automatic tools like Erato, e.g., for further analysing the rhythm of the poems with regards to stresses. Nevertheless, we do not discard resorting to human evaluations, as Morain, Kinghorn, and Ventura (2023), especially regarding more subjective aspects.

Relatedness to given topics is another aspect to quantify in the future. Yet, it would also be very interesting to understand to what extent LLMs are generating poetry and not simply retrieving already-written poems, possibly licensed. A similar approach to (D’Souza and Mimno 2023) could be explored to answer to this question.

We should add that we only employed open LLMs, assuming that they would proxy rather well the situation for proprietary LLMs. Moreover, we recall that proprietary models commonly present barriers to a wider adoption by the community, by means of cost, computational power and unrevealed details on architecture and pretraining. Nevertheless, we are aware that: (i) using a quantized version of Llama2 might have had a negative impact; (ii) and, despite having tested the 7b and 70b models, recent proprietary models like GPT-4 (Achiam et al. 2023) have much more parameters, which might have a positive impact on the results. Therefore, in the future, it would still be interesting to check whether similar patterns are observed in GPT-3.5, GPT-4, or other larger models.

Acknowledgements

Part of the computation done for this project was performed on the UCloud interactive HPC system, which is managed by the eScience Center at the University of Southern Denmark. This work was partially supported by national funds through FCT, within the scope of the project CISUC (UID/CEC/00326/2020).

References

- Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Alentschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Agirrezabal, M.; Gonçalves Oliveira, H.; and Ormazabal, A. 2023. Erato: Automating poetry evaluation. In *Proceedings of EPIA Conference on Artificial Intelligence*, 3–14. Springer.
- Bender, E. M.; Gebru, T.; McMillan-Major, A.; and Shmitchell, S. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 610–623.
- D’Souza, L., and Mimno, D. 2023. The chatbot and the canon: Poetry memorization in LLMs. In *Proceedings of the Computational Humanities Research Conference (CHR)*, volume 3558 of *CEUR Workshop Proceedings*, 475–489. CEUR-WS.org.
- Elzohbi, M., and Zhao, R. 2023. Creative data generation: A review focusing on text and poetry. In *Proceedings of the 14th International Conference on Computational Creativity, Ontario, Canada*.
- Gonçalo Oliveira, H.; Hervás, R.; Díaz, A.; and Gervás, P. 2017. Multilingual extension and evaluation of a poetry generator. *Natural Language Engineering* 23(6):929–967.
- Gonçalo Oliveira, H. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of 10th International Conference on Natural Language Generation, INLG 2017*, 11–20. Santiago de Compostela, Spain: ACL.
- Henrickson, L., and Meroño-Peñuela, A. 2023. Prompting meaning: a hermeneutic approach to optimising prompt engineering with ChatGPT. *AI & SOCIETY* 1–16.
- Hutson, J., and Schnellmann, A. 2023. The poetry of prompts: The collaborative role of generative artificial intelligence in the creation of poetry and the anxiety of machine influence. *Global Journal of Computer Science and Technology: D* 23(1).
- Kirmani, A. R. 2023. Artificial intelligence-enabled science poetry. *ACS Energy Letters* 8(1):574–576.
- Lamb, C.; Brown, D. G.; and Clarke, C. L. 2017. A taxonomy of generative poetry techniques. *Journal of Mathematics and the Arts* 11(3):159–179.
- Lin, C.-Y., and Och, F. J. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 605–612.
- Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegrefe, S.; Alon, U.; Dziri, N.; Prabhunoye, S.; Yang, Y.; et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems* 36.
- Morain, R.; Kinghorn, B.; and Ventura, D. 2023. Are language models unsupervised multi-domain CC systems? In *Proceedings of the 14th International Conference on Computational Creativity, Ontario, Canada*.
- Orekhov, B., and Fischer, F. 2020. Neural reading: Insights from the analysis of poetry generated by artificial neural networks. *Orbis Litterarum* 75(5):230–246.
- Plecháč, P. 2018. A collocation-driven method of discovering rhymes (in Czech, English, and French poetry). *Taming the Corpus: From Inflection and Lexis to Interpretation* 79–95.
- Sawicki, P.; Grzes, M.; Goes, F.; Brown, D.; Peepkorn, M.; Aisha, K.; and Simona, P. 2023a. On the power of special-purpose GPT models to create and evaluate new poetry in old styles.
- Sawicki, P.; Grzes, M.; Goes, F.; Brown, D.; Peepkorn, M.; and Khatun, A. 2023b. Bits of grass: Does gpt already know how to write like whitman? In *Proceedings of the 14th International Conference for Computational Creativity*.
- Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.
- Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35:24824–24837.
- Yao, S.; Chen, H.; Hanjie, A. W.; Yang, R.; and Narasimhan, K. 2023. Collie: Systematic construction of constrained text generation tasks. *arXiv preprint arXiv:2307.08689*.
- Yu, C.; Zang, L.; Wang, J.; Zhuang, C.; and Gu, J. 2024.

Charpoet: A chinese classical poetry generation system based on token-free llm.

Appendix 1

Full interaction with the the 7B open LLM LLaMa model to produce a short poem about the ICCC conference.

User: Can you write me a short poem for an article that I will send to a conference that is happening in Sweden?

LLaMa: *Of course! I'd be happy to help you with a poem for your conference in Sweden. Can you please provide me with some details about the conference, such as its theme, location, and any specific ideas or emotions you would like the poem to convey?*

User: The conference is about computational creativity, it is happening in jönköping and a colleague and I are very excited to go

LLaMa: *Great! Here's a short poem for your conference on computational creativity in Jönköping:*

*In Sweden's heartland, where tech meets art,
A gathering of minds, to spark new starts.
From code to canvas, from machine to mind,
Creativity blooms, as innovators unwind.*

*Jönköping welcomes you, with open arms,
To share and learn, in this digital charm.
From theory to practice, we'll explore,
The limitless potential of computational lore.*

*Colleagues become friends, over cups of tea,
Exchanging ideas, as creativity flows free.
The future unfolds, with each new find,
As we push boundaries, and cross the digital line.*

*So come and join us, in this Scandinavian delight,
Where technology meets art, and creativity takes flight.
Jönköping awaits, with open doors,
To inspire and empower, for more.*

Appendix 2

Interaction between user and Llama2 (7B) regarding the number of syllables.

User: can you give me a sentence with 5 syllables?

LLaMa: *The cat sat.*

User: I think that sentence has 3 syllables

LLaMa: *My apologies, you are correct. Here is a sentence with 5 syllables: "The dog ran quickly."*

...

User: Can you provide me with 10 sentences which have 7 syllables each?

LLaMa: *Of course! Here are 10 sentences with 7 syllables each:*

- 1. The cat purred contentedly on my lap. (7 syllables)*
- 2. The baby giggled at the toy. (7 syllables)*
- 3. The flowers bloomed in the garden. (7 syllables)*
- 4. The dog wagged its tail with excitement. (7 syllables)*
- 5. The children played outside all day. (7 syllables)*
- 6. The snowflakes fell gently on my face. (7 syllables)*
- 7. The clock ticked away the minutes. (7 syllables)*
- 8. The bird sang a beautiful melody. (7 syllables)*
- 9. The rain fell steadily on the ground. (7 syllables)*
- 10. The fire crackled and popped in the fireplace. (7 syllables)*

User: None of them has 7 syllables, even though you marked that

LLaMa: *I apologize for my mistake earlier. You are correct, none of the sentences I provided have 7 syllables. Thank you for pointing this out and giving me another chance to provide accurate responses. Please let me know if you have any other questions or requests.*

User: Can you provide me with 10 sentences which have 7 syllables each?

LLaMa: *Of course! Here are 10 sentences with 7 syllables each:*

- 1. The sky is blue and wide (7 syllables)*
- 2. The cat is black and sleek (7 syllables)*
- 3. The sun sets in the west (7 syllables)*
- 4. The baby cries for food (7 syllables)*
- 5. The dog runs quickly away (7 syllables)*
- 6. The flowers bloom in spring (7 syllables)*
- 7. The snowflakes fall gently (7 syllables)*
- 8. The clock ticks away the time (7 syllables)*
- 9. The bird sings a sweet song (7 syllables)*
- 10. The fire burns bright and hot (7 syllables)*

User: There are three sentences among the last ones that contain 7 syllables. Can you guess which are they?

LLaMa: *Yes, I can! Among the last 10 sentences provided, the following three have 7 syllables:*

- 1. The snowflakes fall gently (7 syllables)*
- 2. The clock ticks away the time (7 syllables)*
- 3. The fire burns bright and hot (7 syllables)*