A Casual Co-Creative Dutch Poetry Creator using RoBERTa and GPT-2

Lennert Saerens
Vrije Universiteit Brussel
Brussel, Belgium
lennert.saerens@vub.be

Abstract

In this paper, a co-creative Dutch poetry generation system is presented. The system implements a simple co-creative design pattern that allows its users to create poems line by line, starting from a number of keywords. The system uses a neural approach, consisting of an encoder-decoder architecture that uses a specialised RoBERTa model for the encoder and a specialised GPT-2 model for the decoder. The evaluation shows that the system can create Dutch poetry successfully. Finally, an experiment enquiring into the satisfaction and fulfillment users might get from using a co-creative system to write poems, is proposed.

Introduction

Poetry is a rich and dynamic form of artistic expression that uses language to create a powerful emotional experience for the reader or listener. It is a literary genre that has been part of human culture for centuries, and its influence can be seen in many different fields. At its core, poetry is about using language to convey complex ideas and emotions in a way that is both condensed and heightened. This can involve the use of various techniques such as meter, rhyme, alliteration, and figurative language, which allow poets to create a sense of rhythm and musicality in their work.

There are many different forms of poetry, from the structured and traditional, such as sonnets and haikus, to the more free-form and experimental, such as free verse. Poets can use these different forms to explore a wide range of themes and subjects, including love, nature, politics, and the human experience. One of the unique features of poetry is its ability to capture the essence of a moment or an emotion, using imagery and metaphor to create a vivid and emotional experience for the audience.

In addition to its artistic value, poetry has also played an important role in shaping cultural and political movements throughout history. Poets such as the Romantics have used their work to challenge societal norms and inspire change.

Overall, poetry is a versatile and enduring art form that can move and transform us in profound ways. It has the power to deeply move us and connect us to the deeper aspects of the human experience. Because of this, poetry is an interesting art form for study within the field of computational creativity.

This paper explores the creative domain of Dutch poetry generation through the development of a co-creative system that assists users in the crafting of Dutch poems, one line at a time. The system, which is designed to be particularly beneficial for individuals who may find it challenging to write poetry, operates by prompting the user to input a set of keywords. These keywords then serve as the inspiration for the system to suggest potential opening lines, mimicking the thought process a poet might engage in when conceptualizing a new piece.

The co-creative approach that is employed necessitates a dynamic interaction between the user and the system. This interaction poses a unique challenge to the creative agent, which must generate lines that are semantically and topically coherent, diverse, and non-repetitive. In mirroring the cognitive processes poets undertake, the system is able to produce text with qualities that are both evocative and emotionally resonant.

The co-creative Dutch poetry generation system presented here employs a fine-tuned encoder-decoder architecture, utilizing specialized RoBERTa (Liu et al. 2019) and GPT-2 (Radford et al. 2019) models for the encoder and decoder respectively. The system functions within the conventional norms and rules of Dutch poetry, fostering creativity by generating a diverse array of lines from which users can choose. This not only allows for the production of meaningful and emotionally resonant poetry but also provides the user with a fulfilling creative experience in writing their own poems.

Background

As can be seen from a recent survey in the field, poetry generation is becoming popular among researchers of Natural Language Generation, Computational Creativity and, broadly, Artificial Intelligence (Gonçalo Oliveira 2017). From this survey, we also see that the majority of these poetry generators are targeting the English language.

However, many poetry generation systems for other languages also exist. Some recent examples of such a systems include a system for generating Vietnamese poetry using an enhanced version of GPT-2 that has less topic drift and semantic inconsistency (Nguyen et al. 2021). A very similar system that uses a fine-tuned GPT-2 model exists for the Arabic language (Beheitt and Hmida 2022). There, the authors also focused on on creating a system that has less

topic drift. Another interesting poetry generator is a generator for Finnish poetry that uses a computationally creative genetic algorithm to teach a BRNN model to generate poetry (Hämäläinen and Alnajjar 2019).

Even more recently, another generator that generates modern French poetry was created (Hämäläinen, Alnajjar, and Poibeau 2022). This system pioneers the use of a fine-tuned encoder-decoder architecture that uses a specialised RoBERTa and GPT-2 model as the encoder and decoder respectively.

Despite the growing interest in non-English poetry generation, currently, no work on the generation of Dutch poetry exists. Another area within which more research can be conducted is co-creative poetry generation. While there exist a number of co-creative poetry generators (Gonçalo Oliveira et al. 2019; Boggia et al. 2022b; Hämäläinen 2018; van Heerden and Bas 2021), there is still room for other systems to try out different approaches and evaluate the impact of co-creativity on the user's creative process. One particularly interesting example of a co-creative poetry generation system within this discussion is *Co-PoeTryMe* (Gonçalo Oliveira et al. 2019). This system was built on top of the *PoeTryMe* (Gonçalo Oliveira 2012) system for automatic poetry generation after its users had expressed their desire to be more involved in the poetry generation process. This demonstrates some of the virtues co-creative systems might have.

Casual Dutch Poetry Creator

Within the previously described context, the aim of this work is threefold: (1) Create the first Dutch poetry generation system, (2) create a co-creative system that helps users generates poetry one line at a time, and (3) advance research within the field of co-creative poetry generation. To achieve all off these goals, the casual Dutch poetry creator proposed here combines ideas from several papers (Hämäläinen, Alnajjar, and Poibeau 2022; Boggia et al. 2022b; 2022a) into a new co-creative system. The proposed system consists of two major parts: the user interface, and the underlying model. Each of these has its own responsibilities. The user interface is responsible for prompting the user for their initial input. Afterwards, the user interface shows the user each of the possible lines the system has generated. The underlying model is responsible for generating the actual lines of poetry based on the input the user provided, and the lines that were previously chosen by the user.

Design Pattern

The Dutch poetry generator presented here, implements the simple co-creative design pattern for casual poetry generators (Boggia et al. 2022b). The aim of this design pattern is to make writing poems more fun and playful for "people who are not versed with poetry but who could nevertheless have joy from writing it". The goal of a co-creative system implemented using the design pattern is thus much more oriented at making the user feel "the joy of creativity" rather than generating prize-winning poetry. This design pattern was chosen because it has very simple user interaction. This lessens the overall complexity of the system, which allows for focusing on the poetry generation, and simplifies the evaluation. This also allows the user to focus on their own creative thought process rather than fine-tuning parameters.

The actual design pattern follows the method of generating poetry one line at a time. First, the user can input a number of keywords. Based on these keywords, the system will generate a number of suggested first lines for the poem. The user can then choose one of these first lines to their own liking. After the user has chosen a first line for their poem, the system will use the chosen first line as inspiration to generate candidates for the second line. The user can, at any point, decide they are satisfied with the poem that has been generated by sequentially choosing lines proposed by the system. At this point the system outputs the now completed poem.

In this system, this design pattern was implemented using a very simple command-line interface. A complete graphical user interface could be implemented but this is beyond the scope of this work. An example of an interaction between the user and the system is shown in Listing 1.

Listing 1: Example output of the proposed co-creative Dutch poetry generator.

```
INSPIRATIE: zee avond

KANDIDAAT-REGELS:
0 De zee, in de avondschemering,
1 De zee, in het avondgloren,
2 De zee, als een avond met haar eigen mysterie,
3 En avondlijke charmes, door de zee gegeven,
4 De aard van de avond en de geheimen van de zee,
5 De zee, in de avondtijd, blijft mysterieus,
6 De adem van de avondzee waait,
7 En de zee, als een avondlijke nevel,

KIES EEN KANDIDAAT
([0,1,2,3,4,5,6,7], -1 om te stoppen.)

[...]

HUIDIGE STAAT VAN HET GEDICHT:
De adem van de avondzee waait,
Over de golven en door de getijden,
De zilte geur, als de adem van de nacht,

KANDIDAAT-REGELS:
0 Een schelp op het avondstrand,
1 De adem van zeewier in de avondbries.
2 Een vuurtoren in de donkere nacht,
3 Een schelp op het zandstrand,
4 Een schelp in de avondbries.
5 Een vuurtoren in de ochtendbries.

KIES EEN KANDIDAAT
([0,1,2,3,4,5], -1 om te stoppen.)
```

When using the system, the user will only come into contact with the user interface. This will prompt the user to either input keywords, or select on of the generated lines. I would argue that this process of first entering a number of keywords, and then generating lines one by one is quite close to what an actual poet might do when writing a poem.

At first, they might have an idea around which they might want to write a poem. This is analogous with entering a number of keywords. Then, the poem slowly forms line by line.

Finally, the chosen design pattern also influences the different interactions that occur within the system. The user interface interacts with the poetry generator by first asking it to generate a number of candidates for the poem's first line. The generator then interacts with the user interface by sending it the generated lines. The user will then interact with the user interface to either quit, or select a line, at which point the system either prints out the complete poem, or the generator is once again queried by the user interface. The user interface and the poetry generation model keep interacting with one another until the user is satisfied with the poem and quits the application.

Data

Although the training and validation data will not be an actual software component of the system, it still plays a vital role in the development of this kind off neural-based system. In order to train the encoder-decoder models that generate the actual poetry, a large amount of data is needed. This data is gathered by scraping all poems found on *gedichten.nl* as well as all Dutch poetry found on *poetryinternational.com*. Together, the data gathered from these two web sources form a large corpus of Dutch poetry that is used to train the model. 5573 poems were pulled from gedichten.nl and 1846 poems from poetryinternational.com. This brings the total number of poems in the corpus to 7419. Given the variance in the lengths of these poems and sonnets, each of them was segmented into individual stanzas. Subsequently, each stanza was treated as a separate poem to achieve a uniform length across the corpus. This process resulted in a total of 22.257 Dutch poems contained within the training corpus.

Architecture

The creative system's backend consists of two separate models: the *first-line model*, and the *next-line model*. This two-model system was first introduced in an mBart based implementation of the co-creative design pattern for casual poetry generators (Boggia et al. 2022a). The first-line model is responsible for generating possible first lines for the poem based on the keywords that the user entered via the command-line interface. The next-line model is responsible for generating consequent lines, based on the lines that were previously selected by the user. Both models use the same encoder-decoder architecture for sequence-to-sequence text generation.

The encoder and decoder models were not trained from scratch, but instead pretrained models were used for both. These models are then fine-tuned using transfer learning. It is important to pick suitable pretrained models.

RobBERT[1] (Delobelle, Winters, and Berendt 2020), a Dutch RoBERTa based language model was chosen for the encoder. This is a robust pre-trained model specifically designed for the Dutch language, which can be tailored to any given dataset for a variety of tasks including text classification, regression, or token-tagging (Delobelle, Winters, and Berendt 2020). Its effectiveness has been shown by numerous researchers, who have utilized it to attain cutting-edge performance across a broad spectrum of Dutch natural language processing tasks. For the decoder *gpt2-medium-dutch-embeddings*[2] (de Vries and Nissim 2020) is used. The Dutch model has the same Transformer layer weights as the English GPT-2 model, but its lexical layer has been fine-tuned for the Dutch language.

Using a specialised RoBERTa based model for the encoder, and a specialised GPT-2 based model for the decoder, allows the system to leverage both the natural language comprehension capabilities of RoBERTa, as well as the natural language generation capabilities of GPT-2 (Hämäläinen, Alnajjar, and Poibeau 2022).

Both these models were pretrained for dutch language generation tasks and were fine-tuned using the collected Dutch poetry corpus to excel at generating poetry within this architecture. Of course, the training process of the first- and next-line models will determine which lines are generated by the system, and thus suggested to the user. This means the knowledge of the system is captured in these pretrained fine-tuned models.

When fine-tuning sequence-to-sequence text generation models, source- and target text data is needed. Since the first-line model uses the keywords as its input, the source data for this model consists of sets of keywords. As no predetermined keywords were available, substitute keywords were derived from the initial lines by randomly selecting two or more content tokens from the nouns, adjectives, and verbs present. The source text for each fine-tuning instance is created by randomly reordering and joining the tokens. The target data consists of first lines from the collected poem corpus. Using these keyword proxies as source text, and the first lines as target text, the model can be fine-tuned to generate first lines of poems based on keywords that appear in them. This approach is also used in the previously mentioned mBart based co-creative poetry generation system (Boggia et al. 2022a). It is also discussed there that multiple fine-tuning strategies exist for the next-line model. The approach that was found the best in (Boggia et al. 2022a) is called the *Next-Line Multi* approach. In this approach, the next-line model is fine-tuned by using up to three consecutive lines from a poem as the source data, and a fourth line, which follows after the first three in the poem, as the target data. The model resulting from this approach generated lines that were diverse and coherent, two qualities that are very desirable for our goal. Because of this, the next-line model that is used in the the proposed co-creative Dutch poetry system is also fine-tuned using the Next-Line Multi approach.

The final important part of the proposed system's architecture is the decoding strategy. The same decoding strategy as in the mBart based implementation of the co-creative design pattern for casual poetry generators (Boggia et al.

[1] https://huggingface.co/pdelobelle/robbert-v2-dutch-base

[2] https://huggingface.co/GroNLP/gpt2-medium-dutch-embeddings

2022a) is used. An effective selection of candidate lines is characterized by its diversity, providing the user with a genuine variety of options. Neural-based models like the one discussed here stochastically generate output sequences, allowing the creation of multiple line candidates from the same model.

These candidates are selected by sampling several sequences from the probabilities predicted by the fine-tuned models. This approach ensures the output sequences do not simply adhere to a distribution of high probability succeeding tokens, but rather they are less predictable and capable of surprising the user (Holtzman et al. 2019).

Evaluation

The process of evaluation is crucial to the success of any co-creative poetry generation system. It serves not only as a means to measure and validate the system's effectiveness in generating coherent, diverse, and original poetry, but also as a valuable tool to identify areas for potential improvement. This dual function of evaluation is particularly significant considering the intricacies and nuances of the Dutch language and the intricate nature of poetry generation. There are essentially two main aspects of evaluation: internal and external. The internal evaluation is related to the system's self-assessment, where it examines the generated poetic lines based on predefined criteria. On the other hand, external evaluation aims to involve human perception, to understand how well the system's output meets the expectations of the end users. In the following subsections, we will delve into the methodologies adopted for both these forms of evaluation, drawing upon successful models and frameworks established in previous research.

Internal Evaluation

Since the architecture of the system presented here is based on the co-creative design pattern for casual poetry generators (Boggia et al. 2022b), their internal evaluation measures are used as the internal evaluation measures for the Dutch poetry creator. Evaluation is still a key part of their paper since an internal system needs to decide which lines to show to the user. This process is based on evaluating each of the generated lines with respect to four criteria: *semantic coherence*, *topic coherence*, *tautology*, and *diversity*. Here, semantic coherence refers to the fact that the presented lines should be coherent on a semantic level with the poem so far, topic coherence refers to the fact that presented lines should have the same topic as the poem so far, tautology refers to the fact that the presented lines should not be repetitive, and diversity refers to the fact that the system should generate a diverse array of lines for the user to choose from. The exact ways in which each of these criteria and their scores can be calculated is explained in the work by (Boggia et al. 2022a) but it is beyond the scope of this paper.

Since the system architecture uses a neural approach, the scores of the generated lines with respect to these criteria is completely based on the way the model is fine-tuned. The authors of the mBart based implementation of the co-creative design pattern for casual poetry generators (Boggia et al. 2022a) tests a number of training methods. The Next Line Multi fine-tuning approach came out on top. In this approach, the next-line model is fine-tuned by using up to three consecutive lines from a poem as the source data, and a fourth line, which follows after the first three in the poem, as the target data. The model resulting from this approach generated lines that were diverse and coherent, two qualities that are very desirable for our goal. Because of this result, the Next Line Multi fine-tuning approach is also used for the co-creative Dutch poetry creator presented here.

External Evaluation

The basic idea of this work is to generate poems line by line. One way of evaluating how good a line proposed by the system is, is to have humans evaluate it. This consist of a qualitative evaluation. The external evaluation for this work is very similar to the approach taken by the authors of the modern French poetry generator (Hämäläinen, Alnajjar, and Poibeau 2022), since the idea of generating poetry with an encoder-decoder model for this Dutch poetry creator was heavily inspired by their approach to automatic poetry generation. In their work they generate poems based on keywords that were not seen during training. These poems were then used to conduct a crowd-sourced evaluation on Appen[3].

Because of this, a crowd-sourced evaluation on Appen was also conducted for this work. It was made sure that the language requirement for the audience on Appen was set to Dutch so that the audience can evaluate the quality of Dutch poetry, especially with regards to various linguistic aspects.

There is however an important difference between the work conducted in (Hämäläinen, Alnajjar, and Poibeau 2022) and the work presented here. The authors of the modern French poetry generator (Hämäläinen, Alnajjar, and Poibeau 2022) generate entire poems while the Dutch poetry creator presented here consists of (1) generating possible lines to start a poem with, or (2) generating possible lines to continue an existing poem. Because of this, both these aspects were evaluated individually.

The evaluation of the first-line model will be discussed first. When the user first starts using the system to generate a new poem, they are prompted to input a number of keywords. These keywords are then used as inspiration by the system to generate a number of possible first lines. To evaluate the quality of the possible first lines that the system generated, both the keywords that served as input, as well as the possible first lines suggested by the system are shown to the audience. They were then asked to evaluate six statements about the generated line in a five point Likert scale, where 1 represents the worst and 5 the best grade. This approach is the same as the one taken by the authors of the modern French poetry generator (Hämäläinen, Alnajjar, and Poibeau 2022). The statements are the following:

1. De gesuggereerde eerste lijnen passen bij de gegeven de sleutelwoorden.
2. De gesuggereerde eerste lijnen zijn verstaanbaar gegeven de sleutelwoorden.

[3]https://appen.com

3. De gesuggereerde eerste lijnen zijn grammaticaal correct.
4. De gesuggereerde eerste lijnen spreken tot de verbeelding.
5. De gesuggereerde eerste lijnen roepen emoties op.
6. Ik vind de gesuggereerde eerste lijnen mooi.

These statements are Dutch translations of statements that seem largely agreed upon since the authors of the modern French poetry generator state that they are used by several other authors for evaluating poetry (Hämäläinen, Alnajjar, and Poibeau 2022).

Finally, the way the keywords and their corresponding possible first lines generated by the system, are selected is also of utmost importance for a fair and sound evaluation. For the generation of these tuples, random sets of four keywords were selected from the training corpus, making sure that that exact combination of keywords was never used during training. These sets of keywords and the lines generated by the system based on them should reflect the ability of the system to generate qualitative Dutch poetry in the most fair way.

20 different sets of possible first lines were generated using the system. Every single set of possible first lines is assessed by 20 different crowd-workers. A single worker has the flexibility to evaluate all 20 poems or just a subset of them. In instances where they evaluate only some, the rest of the unevaluated poems are presented to another crowd-worker for assessment. It's important to note that a specific poem cannot be evaluated multiple times by the same crowd-worker.

	Q1	Q2	Q3	Q4	Q5	Q6
Avg	3.65	3.77	3.55	3.59	3.57	3.69
STD	0.79	0.88	0.77	0.81	0.88	0.75

Table 1: Evaluation results of the creation of first lines.

The evaluation of the first-line model presents encouraging results, which suggest that the co-creative Dutch poetry generation system can generate acceptable and, to a certain extent, engaging first lines for a poem. From the results in Table 1, it can be observed that on a Likert scale of 1 to 5, all the average scores for the different questions are above the midpoint of 3. This indicates a general positive response from the crowd-workers who evaluated the generated lines.

Looking specifically at the criteria, the highest average score was for Q6 (3.69), indicating that crowd-workers liked the suggested first lines to substantial degree. This is a crucial aspect of poetry generation, since an engaging and pleasing first line can draw the reader into the poem. Moreover, the scores for Q1 (3.65) and Q2 (3.77) suggest that the system is successful in generating lines that are both appropriate and understandable based on the provided keywords.

However, there is some room for improvement. Q3, which assesses the grammatical correctness of the generated lines, received a slightly lower average score (3.55), pointing out that the system might occasionally generate lines with grammatical issues. This issue was also observed during the testing and the fine-tuning of the models. The standard deviations suggest that the responses were reasonably consistent, with no question exhibiting excessive variance.

Next, the evaluation of the next-line model will be discussed. Once the user has selected their preferred first line, the system will use the selected first line to generate a number of possible next lines. The way the quality of these next lines will be evaluated will be very similar to the way the generated first lines are evaluated. The same methodology will be used: crowd-workers on Appen will evaluate the generated lines based on modified versions of the statements that were listed previously for the evaluation of the first-line model:

1. De gesuggereerde volgende lijnen passen bij de gegeven de sleutelwoorden.
2. De gesuggereerde volgende lijnen zijn verstaanbaar gegeven de sleutelwoorden.
3. De gesuggereerde volgende lijnen zijn grammaticaal correct.
4. De gesuggereerde volgende lijnen spreken tot de verbeelding.
5. De gesuggereerde volgende lijnen roepen emoties op.
6. Ik vind de gesuggereerde volgende lijnen mooi.

Just like was the case for the first lines, the lines suggested by the system cannot simply be given to the audience for evaluation. Instead, the audience was given both the keywords and the lines that were previously chosen by the user, together with the lines that the system suggests at that point. This is necessary because a number of the statements mentioned, such as understandability, rely on this extra context.

In order to ensure a fair evaluation, the same method as for the evaluation of the first-line model for choosing which keywords to start from, was used. From there on out, random lines were selected to come to the situation that will be evaluated by the audience. The number of lines chosen before arriving at the one under evaluation also varied. This was done because the system only takes into account the last three selected lines.

	Q1	Q2	Q3	Q4	Q5	Q6
Avg	3.63	3.75	3.58	3.62	3.59	3.67
STD	0.80	0.89	0.78	0.82	0.87	0.76

Table 2: Evaluation results of the creation of next lines.

The evaluation of the next-line model reveals promising results, illustrating the ability of the Dutch poetry generation system to effectively generate engaging subsequent lines. The results in Table 2 display an overall positive reception from the crowd-workers evaluating the proposed lines. All the average scores for each question surpass the midpoint of 3 on a 5-point Likert scale, suggesting an overall satisfaction with the lines proposed by the system.

Examining the specific criteria, the highest average score is obtained for Q6 (3.67), implying that the evaluators found the suggested next lines aesthetically pleasant. This is an important observation, since an engaging follow-up line is

crucial in maintaining the reader's interest in the poem. Furthermore, the scores for Q1 (3.63) and Q2 (3.75) indicate that the system is effectively generating lines that align with and are comprehensible based on the provided keywords and previously selected lines.

However, similar to the first-line model, there are areas where the system can improve. Q3, evaluating the grammatical correctness of the generated lines, received a slightly lower average score (3.58). This highlights occasional grammatical issues in the generated lines, marking an opportunity for refinement to enhance the perceived quality of the generated poetry. The standard deviations show reasonably consistent responses across the crowd-workers, without overly drastic variances in their ratings.

Finally, since the work presented here is a co-creative system, it is aimed to be used by humans in real time. Because of this, the system should be responsive to the user's input and should not take too long to generate first and next lines. Things that might influence the speed at which the system operates include: number of keywords, number of already selected lines and thus length of the poem ... To evaluate the responsiveness of the system, several benchmarks were conducted. The application stayed reactive at all times and this quantitative evaluation of the process served as an import test for the system.

Another interesting implication of the co-creative nature of the work presented here is the possible impact the system has on its users' creative process. Because of this, it is very interesting to study whether using a co-creative system would provide a more fulfilling and valuable experience for users rather then using a system that is not co-creative. To accomplish this, an experimental design was implemented involving a crossover study where participants interacted with both systems. The main hypothesis was that the co-creative system would provide a more fulfilling and valuable experience for users. However, since at this point in time no non-co-creative system equivalent to the co-creative Dutch poetry creation systems presented here exist, the experiment could online be designed but not actually conducted. If the conducted were to be conducted, it would go as follows.

Participants would be recruited from a variety of backgrounds. All participants would provide provide informed consent before beginning the experiment. Each participant would be randomly assigned to start with either the co-creative or the non-co-creative poetry generation system. They would then be asked to select a set of keywords that they would like to use as inspiration for a poem. Using the assigned system, participants would generate a poem based on the chosen keywords. Upon completion, they would fill out a survey evaluating their experience and the resultant poem.

The same process would be repeated using the other system, ensuring that each participant has an opportunity to create poems with both the co-creative and the non-co-creative system. The order of system use would be randomized to control for potential order effects.

The survey would include a combination of Likert-scale and open-ended questions to elicit both quantitative and qualitative data. Participants would be asked to rate their level of satisfaction, enjoyment, and perceived creativity from their experience using each system. Additional questions would evaluate the ease of use and any frustrations encountered during the process. Open-ended questions would provide an opportunity for participants to express their thoughts and feelings about each system in their own words, and to articulate any perceived differences between their experiences with the two systems. Quantitative survey data would be analyzed using statistical tests to determine whether there were significant differences in satisfaction, enjoyment, perceived creativity, and ease of use between the co-creative and non-co-creative systems. Qualitative responses would be thematically analyzed to identify common patterns and trends in user experience. By combining quantitative and qualitative data, the aim would be to provide a holistic view of user experiences with co-creative and non-co-creative poetry generation systems. This method would allow us to explore not only whether the co-creative system led to more fulfilling experiences, but also why this might be the case, paving the way for future enhancements to the design of creative systems.

Conclusion

In summary, the research presented in this paper successfully achieved its threefold objectives: the establishment of the inaugural Dutch poetry generation system, the development of a co-creative system facilitating users to generate poetry incrementally, and the contribution to the burgeoning field of co-creative poetry generation research.

The novel system manifested its accomplishment of the first two objectives by pioneering Dutch poetry generation and integrating a co-creative feature that guides users in creating poems line by line. The external evaluation performed lends credence to these claims, substantiating the system's capacity to produce a diverse range of contextually fitting and semantically coherent lines, in response to keywords and previous lines. However, the constructive insights gleaned from these evaluations underscore opportunities for future improvements, offering critical direction for subsequent endeavors aimed at refining and enhancing the system's performance.

In furtherance of the third objective, a proposed experiment exploring the potential influence of the co-creative system on user creativity offers exciting prospects for future research. This experiment provides a platform to investigate whether user satisfaction or creative fulfillment is significantly heightened when engaging with a co-creative system compared to non-co-creative alternatives, marking another step forward in the field of co-creative poetry generation.

Looking ahead, future iterations of the system may consider implementing an option for users to introduce additional keywords at any point during their poetic composition. Such a feature could significantly bolster the system's creative capabilities by enabling users to incorporate new concepts as they deem fit, potentially introducing unexpected narrative twists. Further enhancement may also involve allowing users to manipulate parameters such as the system's temperature, thereby modulating the diversity of suggested lines. Such advancements could severely augment

the creative synergy between users and the system, allowing for new possibilities for co-creative poetry generation.

Acknowledgments

This Dutch poetry generator and accompanying paper were created within the context of the *Computational Creativity* course taught at the Vrije Universiteit Brussel by Professor Geraint A Wiggins and teaching assistant Nicholas Harley, both of which provided me with extremely helpful feedback and guidance throughout the entire research process.

References

Beheitt, M. E. G., and Hmida, M. B. H. 2022. Automatic arabic poem generation with gpt-2. In *ICAART (2)*, 366–374.

Boggia, M.; Ivanova, S.; Linkola, S.; Kantosalo, A.; and Toivonen, H. 2022a. One line at a time - generation and internal evaluation of interactive poetry. In Hedblom, M. M.; Kantosalo, A. A.; Confalonieri, R.; Kutz, O.; and Veale, T., eds., *Proceedings of the 13th International Conference on Computational Creativity, Bozen-Bolzano, Italy, June 27 - July 1, 2022*, 7–11. Association for Computational Creativity (ACC).

Boggia, M.; Ivanova, S.; Linkola, S.; Toivonen, H.; and Kantosalo, A. 2022b. Casual poetry creators: A design pattern and internal evaluation measures. In Hedblom, M. M.; Kantosalo, A. A.; Confalonieri, R.; Kutz, O.; and Veale, T., eds., *Proceedings of the 13th International Conference on Computational Creativity, Bozen-Bolzano, Italy, June 27 - July 1, 2022*, 2–6. Association for Computational Creativity (ACC).

de Vries, W., and Nissim, M. 2020. As good as new. how to successfully recycle english gpt-2 to make models for other languages.

Delobelle, P.; Winters, T.; and Berendt, B. 2020. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 3255–3265. Online: Association for Computational Linguistics.

Gonçalo Oliveira, H. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, 11–20. Santiago de Compostela, Spain: Association for Computational Linguistics.

Gonçalo Oliveira, H.; Mendes, T.; Boavida, A.; Nakamura, A.; and Ackerman, M. 2019. Co-poetryme: Interactive poetry generation. *Cognitive Systems Research* 54:199–216.

Gonçalo Oliveira, H. 2012. Poetryme: a versatile platform for poetry generation. volume 1, article 21.

Hämäläinen, M.; Alnajjar, K.; and Poibeau, T. 2022. Modern french poetry generation with roberta and GPT-2. In Hedblom, M. M.; Kantosalo, A. A.; Confalonieri, R.; Kutz, O.; and Veale, T., eds., *Proceedings of the 13th International Conference on Computational Creativity, Bozen-Bolzano, Italy, June 27 - July 1, 2022*, 12–16. Association for Computational Creativity (ACC).

Hämäläinen, M., and Alnajjar, K. 2019. Let's FACE it. Finnish poetry generation with aesthetics and framing. In *Proceedings of the 12th International Conference on Natural Language Generation*, 290–300. Tokyo, Japan: Association for Computational Linguistics.

Hämäläinen, M. 2018. Poem machine - a co-creative NLG web application for poem writing. In *Proceedings of the 11th International Conference on Natural Language Generation*, 195–196. Tilburg University, The Netherlands: Association for Computational Linguistics.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nguyen, T.; Nguyen, P.; Pham, H.; Bui, T.; Nguyen, T.; and Luong, D. 2021. Sp-gpt2: Semantics improvement in vietnamese poetry generation. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1576–1581.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8):9.

van Heerden, I., and Bas, A. 2021. Afriki: Machine-in-the-loop afrikaans poetry generation.