

# Casual Poetry Creators: A Design Pattern and Internal Evaluation Measures

Michele Boggia<sup>♡</sup> Sardana Ivanova<sup>♠</sup> Simo Linkola<sup>♠</sup> Hannu Toivonen<sup>♠</sup> Anna Kantosalo<sup>♠</sup>

<sup>♡</sup> Department of Digital Humanities <sup>♠</sup> Department of Computer Science  
University of Helsinki, Finland  
{first.last}@helsinki.fi

## Abstract

We explore the concept of Casual Poetry Creators with the aim of making poetry writing fun and entertaining for the user. We present a simple co-creative interaction design pattern based on constructing poems line by line, suggesting the user a set of line candidates at each step. We also propose objective measures by which a Casual Poetry Creator can evaluate and choose which line candidates to show to the user and sketch out a plan to evaluate the measures and pattern with users.

## Introduction

Writing poetry is a creative act. Poets do it for various reasons—to communicate a feeling or a viewpoint, for self-expression or for therapeutic reasons, for instance. In this paper, we address people who are not versed with poetry but who could nevertheless have joy from writing it—given that they had access to easy-to-use tools that make the threshold to try out poetry writing very low.

We explore the concept of *Casual Poetry Creators*, systems that use a simple interaction pattern with the aim of making poetry writing fun and entertaining for novices. Casual Creators, a term coined by Compton and Mateas (2015), refers to a class of co-creative tools characterized by playfulness and the lack of task-focus. The main goal for Casual Poetry Creator systems, then, is not to generate great poetry, but rather to help the user feel the joy of creativity.

We contribute two elements to Casual Poetry Creators:

First, the defining element of our Casual Poetry Creator is a simple interaction pattern where poems are generated line by line, with the user in control over which lines are used in the poem. Specific design advice on casual creators has been published in the form of design patterns (Compton and Mateas 2015; Compton 2019; Petrovskaya, Deterding, and Colton 2020) and case studies of designing suitable parameter spaces for casual creation e.g. in the domains of games (Colton et al. 2018) and visual arts (Colton et al. 2020). Several simple, interactive poetry generators have been proposed, too. However, as far as we know, this is the first time the task is considered within the casual creators framework. The actual poetry generation method is outside the scope of this paper; instead, we present methods in a separate paper (Boggia et al. 2022). Casual Poetry Creators can be implemented with different generation methods, e.g., sequence-

to-sequence linguistic models to generate lines. We hope that our work encourages researchers to contribute novel Casual Poetry Creators based on their models.

Second, we define objective evaluation measures for assessing candidate lines for poetry. These measures have several applications: (a) with these measures, a Casual Poetry Creator can internally evaluate its line candidates, so as to provide an appropriate set to the user; (b) when designing a new Casual Poetry Creator, the measures can be used to assess the suitability of different poetry generators for casual creation; (c) during the building of a Casual Poetry Creator, the measures can help fine-tuning linguistic models for this particular purpose.

This paper is structured as follows. In the next section, we briefly review background in casual creators and interactive poetry writing. We then introduce the interaction pattern for Casual Poetry Creators. Next, we give definitions of objective measures that can be used to implement Casual Poetry Creators. We wrap this paper up with concluding remarks. In a parallel paper (Boggia et al. 2022), we give poetry generation algorithms that are suitable for Casual Poetry Creators, we describe implementations of the objective measures, and we give empirical results.

## Background

### Casual creators

The concept of Casual Creators (Compton and Mateas 2015) gives a name for an old phenomenon covering both physical tools as well as software characterized by assistance, automation and limiting the domain space of possible creative outputs to support novice creators (Compton 2019, p. 20).

Compton and Mateas (2015) define a casual creator as an interactive system that encourages fast, confident, and pleasurable exploration of a possibility space, resulting in the creation or discovery of surprising new artifacts that bring feelings of pride, ownership, and creativity to the users that make them.

Casual creators offer an interesting platform for computational creativity developers to develop applications for use in the real world. Examples of casual creation emerge in physical toys, as part of other, more complex software, such as character creation tools within games, and as tools or games are re-used for casual creation instead of their original pur-

pose (Compton 2019, p. 6, 11, 14). Dedicated applications conforming with casual creation are also readily available on commercial app platforms, such as the Apple App Store (Petrovskaya, Deterding, and Colton 2020), further speaking to their role as a widely available form of pass-time for novice creators. In addition, Casual creators offer opportunities to create well-being for their users (Compton 2019, p. 3), making them a significant area to improve the outreach of computational creativity research.

The goal of our Casual Poetry Creator interaction pattern and the metrics are the same as with any casual creator systems: they focus on the users' enjoyment of the creative process itself above productivity and scaffold the creative process by enabling the rapid and fluent exploration of a restricted creative space (Compton 2019, p. 6–7).

## Interactive Poetry Generators

Poetry generation is a popular research topic in computational creativity and numerous methods have been proposed in the literature. A review of different techniques to generate poetry is out of the scope of this paper, however, and we refer the interested reader to Gonçalo Oliveira (2017).

Interactive poetry generation where the software acts as an intelligent or creative partner has also been addressed by several scholars. The following are representative examples of interactive poetry writing systems.

*The Poetry Machine* (Kantosalo et al. 2014; Kantosalo, Toivanen, and Toivonen 2015) uses a fridge magnet metaphor for interaction. The system starts with a sample poem generated by the system, and then the user can move words or lines around, write more text, or ask the system to suggest new words or lines.

*Hafez* (Ghazvininejad et al. 2017) produces sonnets based on given words and eight style parameters tuned by the user. The interaction model is based on the user adjusting the parameters and asking the system to regenerate the poem according to the new parameters.

*Machine in the Loop* (Clark et al. 2018) is an approach used for writing text line by line in other fields of creative writing such as stories and poetry. At every iteration, the system suggests a line which the user may then edit.

*Co-PoeTryMe* (Oliveira et al. 2019) includes user interface functions similar to the Poetry Machine, and additionally it allows constraints to be specified for new words (e.g. rhyme) and offers an extensive editing functionality.

We focus on poetry writing applications using a very simple exploration method for the creative conceptual space of poetry. The space is initialized based on user-given input keywords, and subsequent direction of poetry generation takes place through the one-touch and mutant-shopping interaction patterns for casual creators (Compton and Mateas 2015; Petrovskaya, Deterding, and Colton 2020).

The Casual Poetry Creator design pattern that we describe in the next section is simpler than in any of the above systems. Our aim is to make the use of Casual Poetry Creators as simple as possible: no parameters to tune, no different user interface functions to choose from—not necessarily even an option to edit lines produced by the system, which removes the need for a keyboard when writing poetry with a

Casual Poetry Creator. For the input of keywords that function as seeds for line generation, different keyboard-free alternatives can be considered such as pointing at words in a document, or offering the user a set of random words from which to select the seeds.

## A Design Pattern for Casual Poetry Creators

We consider a simple model for co-creative poetry generation. The poetry generator produces poetry one line at a time, in simple interaction with the user:

1. Before the generation of the first line, the user may give a couple of keywords;
2. Candidates for the first line are produced with the keywords as inspiration; if no keywords were provided, first line candidates are produced based on random keywords sampled from a dictionary;
3. The user selects one of the candidate lines suggested by the system;
4. Candidates for the next lines are produced based on the previous lines (and potentially the keywords);
5. The poem is constructed iteratively and incrementally by going back to step 3; the user may decide to stop the generation of new lines at any time, in which case the final poem is printed by the system.

This design allows very simple user interaction. At any time, the system provides a handful of candidate lines from which the user chooses one, and the system then generates candidates for the next line based on the previous selections of the user.

The generation of candidate lines should satisfy three criteria: (1) each candidate line is related to the previous lines in the poem, or to the possible keywords in the case of the first line; (2) each candidate line is poetic; and (3) the candidates for the  $n$ th line are diverse.

In this paper we focus on this simple interaction pattern for two reasons. First, we believe this makes a good Casual Creator, with extreme simplicity for the user but still a wide space of possible poems (assuming the poetry generator is successful in the three goals listed above). Second, the simple interaction pattern can be easily reused with other poetry generation methods, as well as in Casual Creators in other domains where incremental creation of artefacts is natural.

In Listing 1, we present an example implementation of the Casual Poetry Generator pattern as a command line interface.<sup>1</sup> The example shows the keyword input and a few iterations of the creation process including the line candidate examples and the current lines in the poem.

The line-by-line candidate generation model can be seen as an instance of several design patterns of Casual Creators (Compton and Mateas 2015). It gives instant feedback in the form of new lines based on the user's selection; it produces many candidates and overlays them; it avoids the blank canvas problem by writing lines for the user; it offers limited actions to encourage exploration; and it's mutant shopping

<sup>1</sup>A Python implementation is available at [https://github.com/bmichele/poetry\\_generation](https://github.com/bmichele/poetry_generation).

### Listing 1: Example outputs of the command line implementation of a casual poetry generator.

```
GIVE KEYWORDS: nature summer

LINE CANDIDATES:
0 Nature, in the summer's heat,
1 Nature, in summer's sunshine-bright,
2 Nature, like Summer's with her own decrees,
3 And summer's charms, by Nature given,
4 The summer's nature and the summer's love,
5 Nature, in summer time, is still the same,
6 The summer's breath of nature blows,
7 And Nature, like a Summer's flowery mist,

PLEASE CHOOSE CANDIDATE
(integer in [0,1,2,3,4,5,6,7], -1 to stop.)

[...]

CURRENT POEM STATE:
The summer's breath of Nature blows,
Across the fields and through the trees,
Its fragrance, like the breath of May,

LINE CANDIDATES:
0 A rose-leaf in the garden-bough,
1 The breath of flowers blown in the breeze.
2 A rose-bush in the midst of May,
3 A rose-bush in the garden-bough,
4 A rose-leaf in the garden breeze.
5 A rose-bush in the morning breeze.

PLEASE CHOOSE CANDIDATE
(integer in [0,1,2,3,4,5], -1 to stop.)
```

in the sense that it offers alternative lines ready to be picked by the user (however instead of changing the whole artifact, our pattern focuses on additive iteration). Saving and sharing is trivial since the approach only operates on and produces text.

Perhaps the closest parallels within existing casual creator patterns are 'Limiting Actions to Encourage Exploration' and 'Mutant Shopping' (Compton and Mateas 2015). Casual Poetry Creators could be implemented in a mobile interface with a 'One-touch creativity' pattern, which uses only one type of gesture for the interaction (Petrovskaya, Deterding, and Colton 2020). The basic interaction offered by the pattern could of course be extended, by for example allowing the user to edit the lines. Such an extended pattern begins to resemble user interactions with existing co-creative poetry writing systems, such as the Poetry Machine (Kantosalo et al. 2014) or Co-PoeTryMe (Oliveira et al. 2019).

### Internal Evaluation Measures

We propose four evaluation measures to assess poetry lines produced by Casual Poetry Creators: semantic coherence, topic coherence, tautology, and diversity. In this paper, we do not aim to measure how poetical lines are.

These measures can be utilised both (1) by the designer

of the system during the system development to assess the feasibility of the generation methods and (2) by the system itself during its execution time to make informed decisions about which set of generated line candidates to show to the user. The evaluation measures are based on metrics and other measures previously proposed in the literature.

**Preliminaries** For two vectors, the *cosine similarity* is defined as the cosine of the angle  $\theta$  between the vectors.

For two sets of tokens  $S_1, S_2$  (lines consisting of words), *token similarity*  $\text{sim}(S_1, S_2)$  is defined based on their overlap as

$$\text{sim}(S_1, S_2) = \frac{2|S_1 \cap S_2|}{|S_1| + |S_2|}. \quad (1)$$

**Semantic Coherence** Candidate lines offered by a Casual Poetry Creator should usually be semantically coherent with the poem written so far. For this purpose, we define the *n-Semantic Coherence* to measure the semantic similarity of a candidate line to the  $n$  previous lines. This measure can be used by a Casual Poetry Creator to decide which lines to show the user. For instance, the measure could be used to select a set of candidates mostly consisting of lines coherent with the previous ones, but also include a few less coherent ones to allow for surprises and turns in the poem.

The  $n$ -Semantic Coherence of a candidate verse for the  $i$ th line of the poem is defined as follows. We consider the  $n$  previous lines, i.e., lines  $i - n$  to  $i - 1$ , transform them to a vector representation and compute its cosine similarity with a vector representation of the candidate line.

More specifically, we tokenize each line, remove stopwords, and compute the centroid of the word vectors obtained for each token of the  $n$  previous line from the Word2vec model (Mikolov et al. 2013a; 2013b). The  $n$ -semantic coherence of the candidate is then the cosine similarity between this vector and the vector obtained from the candidate line by following the same procedure (tokenization, stopword removal, computation of centroid by averaging word vectors).

The idea is that the two vectors are semantic encodings of the last lines of the poem and of the candidate line, respectively, and that their cosine similarity captures the degree of their semantic similarity. Line candidates introducing new subjects into the poem will have lower semantic coherence.

**Topic Coherence** Candidate lines suggested by a Casual Poetry Creator should usually be related to the keywords given by the user (if any). We define *Topic Coherence* of a candidate line as its semantic similarity with the keywords. A Casual Poetry Creator can use the topic coherence in ways analogous to semantic coherence, e.g., to ensure that the set of candidate lines contains both topic coherent and potentially surprising lines.

Technically, the topic coherence of a candidate line is defined as the cosine similarity between the centroid of (the word embeddings of) the line and the centroid of (the word embeddings obtained from) the user-given keywords.

The idea is to extend the concept of semantic coherence defined above and offer means to measure the topic drift of

candidate lines from the initial keywords. Candidates characterized by lower scores, when compared with the input keywords, would look more surprising but potentially incoherent to the user. High values, in turn, imply lower surprise and higher coherence.

**Tautology** Many sequence-to-sequence language models are prone to produce unnecessarily repetitive lines, or *Tautology*, and safe-guarding against them can be needed. (For instance, in our implementation (Boggia et al. 2022) we use mBART, which is pre-trained on denoising tasks (Liu et al. 2020). If fine-tuning is not successful, the model will tend to repeat the same verse(s) over and over again.) A measure of tautology allows a Casual Poetry Creator to filter our repetitive lines, if needed.

For a candidate line, we define tautology as the number of tokens that are shared between the candidate and the previous line of the poem, normalized by the total number of tokens in the two verses. We can express this measure using token similarity simply as  $\text{sim}(S_i, S_{i-1})$ , where  $S_i$  and  $S_{i-1}$  are the sets of words obtained from the candidate and the previous poem, respectively.

**Diversity** A Casual Poetry Creator should produce a diverse set of candidate lines at each generation step. This ensures that the user has a real choice and is more likely to feel ownership and pride of the resulting poem. We define the *Diversity* of a set of lines by the amount of words shared between them. Usually, a Casual Poetry Creator would try to maximize the diversity in the candidate lines it offers to the user.

To measure the diversity of a set of lines, we utilise token similarity  $\text{sim}(S_1, S_2)$  between two lines, where  $S_1$  and  $S_2$  are the set of words extracted from the lines. The diversity is computed as the average *dissimilarity* between the lines in the line set, where dissimilarity between two word sets  $S_1$  and  $S_2$  is  $1 - \text{sim}(S_1, S_2)$ . That is, for a set of poem lines, we first extract the words from them to obtain a set of word sets  $\mathbf{S} = (S_1, \dots, S_n)$ , and then compute diversity  $\text{div}(\mathbf{S})$  in a following manner:

$$\text{div}(\mathbf{S}) = \frac{n(n-1)}{2} \sum_{i=0}^{n-1} \sum_{j=i+1}^n (1 - \text{sim}(S_i, S_j)). \quad (2)$$

## Empirical Validation and Application

In a parallel paper (Boggia et al. 2022), we empirically validate that the semantic coherence and diversity metrics measure what they are supposed to, and argue that topic coherence and tautology will also behave favorably. We also apply these measures on an actual poetry generation method and report on our empirical findings.

## Planned External Evaluations

We have implemented a Casual Poetry Creator as a command line interface running on a local instance, using the poetry generation method of Boggia et al. (2022). Basic evaluation with end users is already possible with this interface, but we intend to implement it as a web-based tool

for easier access. Offering the system as a web service will also allow easier systematic evaluation of co-creative experiences of users. It would be interesting to investigate the relationship between the internal evaluation metrics and users’ co-creative experiences with the system, offering further insight into the beneficial use of these metrics in systems aiming for Casual Poetry Creation.

## Conclusion

We presented a simple interaction design pattern to facilitate the creation of Casual Poetry Creators. The pattern is based on line-by-line generation and selection of poem contents, and is well suited for human-computer co-creation.

The Casual Poetry Creator design pattern only allows very simple user interaction. The user starts the interaction by providing a small set of keywords. Candidates for the first line are then produced with these keywords as inspiration, and the user selects one of the lines. After that, candidates for the next lines are produced based on the previous lines. The poem is constructed iteratively and incrementally in this manner, until user decides to stop.

The interaction is highly limited on purpose. The goal is to make the threshold for poetry writing as low as possible by keeping the interface simple. This follows the Casual Creator philosophy: the aim of the system is to help the novice user feel joy of creativity, not to accomplish a task.

A successful Casual Poetry Creator has the ability to produce and select suitable sets of candidate lines. We argue that in Casual Poetry Creators, good candidate lines should be coherent with the preceding poem as well as poetic; additionally, the set of candidates should have diversity. While poetry generation methods are outside the scope of this paper, we proposed evaluation measures that can be used as internal filters by different Casual Poetry Creators. The proposed metrics measure the coherence, diversity, and repetition in lines of poetry. Whatever method is used to generate alternative lines, these measures can be used to control what kind of candidate sets are offered to the user.

This paper is a first conceptual step towards Casual Poetry Creators. In a parallel paper (Boggia et al. 2022), we propose matching poetry generation methods, and validate and apply internal evaluation measures. The next step is an evaluation of the concept and of the measures with actual users. Do the users get joy of creativity, and how do various factors—as potentially indicated by the measures proposed—affect how much fun it is? On the technical side, we plan to explore measures related to the concept of poeticness, e.g. by measuring poetic devices such as rhyming and using machine learning, either as part of learning to generate new lines of poetry, or as a separate evaluation step.

## Author Contributions

The concept of Casual Poetry Creators was developed through discussions between all authors. AK acted as a co-creativity expert and wrote the introduction, background and the description of the casual creation pattern together with HT. MB, SI, SL and HT formulated the internal evaluation measures and wrote the corresponding sections and

conclusions. All authors contributed to the writing of the manuscript iteratively.

## Acknowledgments



MB has been partially supported by the ERC Consolidator Grant FoTran (agreement N<sup>o</sup> 771113) and by the grant 825153 (EMBEDDIA) under the European Union's Horizon 2020 research and innovation programme.

SI has been funded by the European Union's Horizon 2020 research and innovation program under grants 825153 (EMBEDDIA) and 770299 (NewsEye). SL and AK have been funded by Academy of Finland (Grant #328729).

## References

- Boggia, M.; Ivanova, S.; Linkola, S.; Kantosalo, A.; and Toivonen, H. 2022. One line at a time — generation and internal evaluation of interactive poetry. In *Proceedings of the 13th International Conference on Computational Creativity*. ACC.
- Clark, E.; Ross, A. S.; Tan, C.; Ji, Y.; and Smith, N. A. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces, IUI '18*, 329–340. New York, NY, USA: ACM.
- Colton, S.; Nelson, M.; Powley, E.; Gaudl, S.; Saunders, R.; Perez Ferrer, B.; Ivey, P.; and Cook, M. 2018. A parameter-space design methodology for casual creators. In *Proceedings of the Ninth International Conference on Computational Creativity*, 264–271. ACC.
- Colton, S.; McCormack, J.; Berns, S.; Petrovskaya, E.; and Cook, M. 2020. Adapting and enhancing evolutionary art for casual creation. In *Artificial Intelligence in Music, Sound, Art and Design*, 17–34. Springer.
- Compton, K., and Mateas, M. 2015. Casual creators. In *Proceedings of the Sixth International Conference on Computational Creativity*, 228–235. Brigham Young University.
- Compton, K. E. 2019. *Casual Creators: Defining a Genre of Autotelic Creativity Support Systems*. UC Santa Cruz.
- Ghazvininejad, M.; Shi, X.; Priyadarshi, J.; and Knight, K. 2017. Hafez: an interactive poetry generation system. In *Proceedings of The 55th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, 43–48. ACL.
- 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: ACL.
- Kantosalo, A.; Toivanen, J. M.; Xiao, P.; and Toivonen, H. 2014. From isolation to involvement: Adapting machine creativity software to support human-computer co-creation. In *Proceedings of the Fifth International Conference on Computational Creativity*, 1–7. Jožef Stefan Institute.
- Kantosalo, A.; Toivanen, J. M.; and Toivonen, H. 2015. Interaction evaluation for human-computer co-creativity: A case study. In *Proceedings of the Sixth International Conference on Computational Creativity*, 276–283. Brigham Young University.
- Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; and Zettlemoyer, L. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics* 8:726–742.
- Mikolov, T.; Chen, K.; Corrado, G. S.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (Workshop Presentation)*.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*. NIPS.
- Oliveira, H. G.; Mendes, T.; Boavida, A.; Nakamura, A.; and Ackerman, M. 2019. Co-PoeTryMe: interactive poetry generation. *Cognitive Systems Research* 54:199–216.
- Petrovskaya, E.; Deterding, C. S.; and Colton, S. 2020. Casual creators in the wild: A typology of commercial generative creativity support tools. In *Proceedings of the Eleventh International Conference on Computational Creativity*. ACC.