

# Creative Data Generation: A Review Focusing on Text and Poetry

**Mohamad Elzohbi, Richard Zhao**

Department of Computer Science, University of Calgary  
Calgary, Alberta, Canada T2N 1N4  
{melzohbi, richard.zhao1}@ucalgary.ca

## Abstract

The rapid advancement in machine learning has led to a surge in automatic data generation, making it increasingly challenging to differentiate between naturally or human-generated data and machine-generated data. Despite these advancements, the generation of creative data remains a challenge. This paper aims to investigate and comprehend the essence of creativity, both in general and within the context of natural language generation. We review various approaches to creative writing devices and tasks, with a specific focus on the generation of poetry. We aim to shed light on the challenges and opportunities in the field of creative data generation.

## Introduction

Data refers to information that can be stored and processed by a computer. It can take many forms and can be generated both naturally, such as through the thoughts and ideas in a person’s mind, and artificially, such as through the use of machine-based models. In particular, computer scientists use the term data to refer to anything that can be stored in the computer’s memory called binary data. The process of generating data can be (1) fully automated, like using a generative model to generate poetry from topic words, (2) semi-automated, where the output is a collaboration between the machine and the human, such as a poem draft generated with human refinements (Lamb, Brown, and Clarke 2017), or (3) entirely manual such as using a text editor to write a poem. The concept of creativity has historically been difficult to define and has not been considered seriously in AI, as it was thought that machines need first to be capable of possessing thoughts and experiencing emotions (Colton, Wiggins, and others 2012). With the advent of deep learning and transformer models, natural language generation (NLG) techniques have become more advanced and the possibility of generating creative output seems more viable.

This paper focuses on the intersection between creative data generation and NLG with an emphasis on poetry generation. First, we address the difficulties in defining creativity and present our own perspective on the key elements of creativity. Second, we review relevant metrics used and how they are relevant to the proposed criteria, then we evaluate NLG models based on those criteria. Third, we provide an overview of the practical creative applications of text generation tasks, reviewing some of the most recent work in this area. We focus on the poetry generation task, examining the methods and models employed.

## Creativity Paradox

Creativity is a complex and often-debated concept that is far from being well-defined. While many tend to focus on the creativity of the end result, the process by which it is created may also play a role in determining its level of creativity. When the steps involved in generating an output are clearly defined and easy to replicate, the output may be seen as less creative, even if it was initially considered as such. For example, the creativity of Johannes Vermeer, one of the most well-known Dutch artists in western history, was put onto the table of controversial discussions when Philip Steadman suggested that Vermeer was using optics in the painting process (Steadman 2002; Teller 2014). Furthermore, the English painter Hockney (2006) argued that most painters since the 15th century secretly used their knowledge of optical science in the painting process. It could be questioned as to why certain painters would choose to keep their techniques secret if this does not raise doubts about the authenticity of their artistic talent and creativity. One possible explanation is that artists desire for their work to be perceived as unique and distinguished from craft-based forms of expression. They may view their work as constituting a higher form of artistic expression, and therefore seek recognition as elite artists rather than as craftsmen, who are considered to have a lower social status (Markowitz 1994). This indicates that the level of creativity in the output can be significantly impacted by the degree of unpredictability. Hence, can photography be considered creative? Paul Delaroche, a classical painter, believed that the emergence of photography marked the death of painting. However, painters disagree and see painting as expressive and original, as it is not simply a mirror reflection of the real world. (Crimp 1981). On the other side, pictorialists argue that they can express their vision through photography by adding their own touches to a real photograph (Hertzmann 2018). Consequently, how valid is it to attribute creativity to machines if they are simply following a predetermined set of rules written by humans? In fact, even when we have a clear understanding of the design of a machine, we may not fully comprehend the internal processes that occur during its training. Does the hint of anonymity allow for the possibility of attributing creativity to them? Does revealing the process diminish the perceived originality of its output? Do we, as humans, understand creativity based on beauty or wonderfulness of the output?

Shneiderman (2007) summarizes the creative process into three main schools: (1) structuralists: who outline the

creative process in four stages: (i) gathering information (preparation), (ii) forming new connections (incubation), (iii) finding sudden insights (illumination) and (iv) verifying and refining those insights (verification). (2) inspirationists: who believe that creative insights can be achieved through sketching, visualization and meditation. (3) situationalists: who see creativity as a social process. In Boden (2004)’s point of view, creativity must induce (i) new, (ii) surprising, and (iii) valuable ideas or artifacts. As a situationalist, Csikszentmihalyi (1997) asserts that novelty is not enough to be considered creative, but the work must be accepted by the relevant field. Hertzmann (2018) argues that machines are merely tools in the hands of artists. These tools are not always predictable but not inherently creative. Hertzmann argues that just as we do not attribute creativity to a brush when watercolors flow on a canvas or to nature and animals when they form beautiful patterns and stunning structures, although animals have brains just like humans, we should not attribute it to machines.

Ultimately, the concept of creativity appears to be complex and multifaceted, making it seem impossible to develop a cohesive paradigm. However, we can see recurring themes and intersecting dimensions that we will summarize in the following: (see Figure 1)

1. **Originality:** To be considered creative, an output must strike a balance in originality. To achieve this, a creative model must avoid lacking originality (plagiarism) especially when learning from creative works. The greater the originality in the output, the more creative it is considered to be. However, too much originality results in confusion, disconnection, and excessive refinement (preciosity).
2. **Unpredictability:** An output that is obvious and predictable is generally seen as banal and boring. The greater the surprise factor in the output, the more creative it is considered to be. However, the output that is completely unpredictable may be perceived as nonsensical and lose its creativity. To achieve a balance between surprise and coherence, a creative model may need to learn a human-like probability distribution and optimize its decoding strategies for creative expression.
3. **Sociability:** We define sociability as the density of meaning and imagery in the output in relation to its appropriate domain. Poets and painters, for example, use language and colors, respectively, to compact complex ideas and emotions into a limited space, such as a verse or a canvas. A work that is semantically condensed provides more material for the artist or critic to engage with and assess. If the work is less sociable, it may be seen as redundant. However, it is important for the output to strike a balance, as an output that is overly dense may be complex and difficult to comprehend.

### Creative NLG: Aspects

Discussions about NLG in the past have centered around determining what constitutes as *real* NLG (Reiter 2016). As the field progresses, the focus has shifted towards determining what constitutes as *creative* NLG. We believe that a cre-

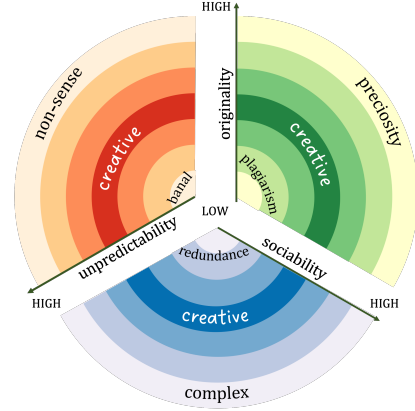


Figure 1: Creativity Dimensions.

ative NLG model must exhibit a sufficient level of originality, sociability, and unpredictability.

Despite the significant advances in NLG, the study of the originality in language models has received relatively little attention. Originality in language can take various forms, such as phrasal, thematic, semantic, or stylistic originality, among others. Although probabilistic models may lead to the generation of original phrases, there is also the risk of over-fitting and copying from the training corpus. Plagiarism in creative language is particularly noticeable, as the output deviates from the normal use of language. This has been, recently, a concern in other creative fields, such as in music generation (Yin et al. 2021). Brooks and Youssef (2021) have noted that there is currently no standard automatic test for originality in NLG and proposed an approach to test the original use of language to identify plagiarism violations using a phrasal counting approach in the ground truth. However, this approach is limited to measuring the phrasal originality which is also important to identify phrases that diverted from their connotation and lost their imagery due to repetitive use in ordinary speech, such as dead metaphors.

Shannon (1951) conducted experiments to predict language redundancy by asking humans to guess the next character in an English passage. Shannon assumed that humans sample from a probability distribution based on their previously learned letter-sequence probabilities and introduced the concept of language entropy.

$$H = - \sum_i p(w_i) \log(p(w_i))$$

where  $p(w_i)$  is the probability for the  $i^{th}$  n-gram to occur. Paisley (1966) studied the effect of several factors, including structure, on the redundancy of the language and found that prose is more redundant than poetic language. A more redundant language is less dense hence less sociable. Due to form restrictions in poems, it seems at first glance that the population of the allowed text must decrease from that of prose, but Manin (2021) postulates that poetic devices expand the population due to the relaxation of language norms. A well-known metric that is used to test NLG output quality is perplexity, which is directly related to entropy. A study

on the correspondence between perplexity and human evaluation revealed that creative output requires perplexity that is not too high and not too low (Keukeleire 2020).

A recent study (Berns and Colton 2020) has introduced the concept of active divergence, which refers to the ability of a model to actively diverge from the training data to achieve a more original and diverse output, as opposed to the traditional method of training the model to perfectly mimic the training data. However, it may be more challenging for a language model to diverge from the training data while still adhering to the rules of grammar, semantics, and syntax. The sampling methods also play an essential role in the output predictability. Applying random sampling leads to a random non-sensical output, and greedy decoding tends to prefer banal or repetitive output. Holtzman et al. (2020) noted that beam search decoding in NLG models may result in less surprising and more monotonic output compared to the high variability in human choice of words. In order to increase unpredictability while keeping randomness low, they proposed Nucleus Sampling, which randomly sample from the top- $k$  most probable words and set a probability threshold  $p$  to ensure that sampling does not occur from peaked or flat distributions. Only the top- $p$  tokens are considered where  $V^{(p)}$  is the smallest set of tokens such that:

$$\sum_{x \in V^{(p)}} P(x|x_{1:i-1}) \geq p$$

The level of creativity in language models is influenced not only by the inference algorithms, but also by how similar the machine language modeling is to human language modeling. Deep learning models, particularly those that use transfer learning, have achieved impressive results on NLG tasks. Auto-regressive (AR) models, such as Generative Pre-Training models (GPT/2/3), generate text by sampling the next token from a probability distribution. Although humans may use causal language when they speak, they often try to modify their speech when they attempt to write creative text. Masked Language Models (MLM), such as BERT, predict masked tokens in a sentence based on context, which is similar to how humans try to guess a better word after removing it from a sentence and considering the context. However, these models will not automatically identify the words to remove and replace. Guided generation allows the model to be responsive and adaptable, and to accept or reject criteria, which may be a step towards a more social model. A study on the steerability of Generative Adversarial Networks (GANs) (Jahanian, Chai, and Isola 2019) shows that they can extrapolate yet are limited to the diversity of the data set. Previous NLG models have been limited by fixed or monotonically increasing sequence lengths, but Levenshtein Transformer (Gu, Wang, and Zhao 2019) and Edit-based Transformer (Xu and Carpuat 2021) allow for dynamic length changes. However, current large language models (LLMs) tend to overlook the integration of the unique elements associated with various forms of creative writing, like poetic diction, in their training process.

## Creative NLG: Devices and Tasks

As discussed earlier, creativity in natural language appears in the original, rich and unusual use of language. Figura-

tive language utilizes figures of speech to modify the literal meanings of words and produce imagery, or utilizes figures of sound to produce an appealing form. These rhetoric devices are used in creative writing in special ways to ensure novelty, richness and unpredictability. This includes similes, metaphors, hyperboles, sarcasm, humor, rhythms and rhymes, among others. These techniques are combined to shape the elements of various types of creative writing such as poetry and prose novels. While most research has focused on the detection of these literary devices, it was until recently that researchers showed interest in models that generate such devices. The employment of MLM and AR based models, along with the versatile BART model that amalgamates the strengths of both approaches and the use of COMET (Bosselut et al. 2019), designed for generating commonsense knowledge, underpinned the observed trend. Next, we will review recent work and its impact on downstream NLG tasks before we introduce poetry generation.

**Metaphor:** Automated metaphor generation has only recently received increased attention (Tong, Shutova, and Lewis 2021). MERMAID (Chakrabarty et al. 2021) finetunes BART with an automatically generated parallel corpus using an MLM model. The MLM will replace metaphorical verbs with their literal complement using the symbol relation from COMET. Then, they use a generator-discriminator approach to transform literal expressions to metaphorical ones with a verb-replacement objective and a top- $k$  sampling strategy. The generated sentences are scored using a RoBERTa-based metaphor detection model to favor sentences with higher quality metaphorical verbs. Stowe et al. (2021) propose a similar model that replaces literal verbs with metaphorical ones guided by conceptual mapping. The authors propose two methods, one by training a frame embedding model and the other by finetuning BART using a generated parallel corpus similar to the one used in MERMAID but tagging each sentence with FrameNet frame labels. A limitation of these models is that they only generate metaphorical verbs and assume a given context.

**Simile:** Chakrabarty, Muresan, and Peng (2020) finetune BART on an automatically built corpus that transforms similes to their literal compliments using the “has property” relationship provided by COMET to generate similes from a given literal sentence by replacing a word by a novel simile. A limitation of the proposed model is that it explicitly replaces only an adjective or an adverb from a given input. Zhang et al. (2021a) propose a simile insertion approach based on the general context of the sentence without replacing words. The approach consists of two main stages: (1) detecting an appropriate simile insertion position in the input sentence using a BERT-based model (2) generating a simile at that position using a standard transformer decoder. Both models are trained jointly in a multi-task learning setup.

**Hyperbole:** MOVER (Zhang and Wan 2022) finetunes BART on a dataset of hyperboles retrieved from an online corpus using a BERT-based hyperbole detection model. The model is trained on a mask-filling task where potential hyperbolic tokens are identified, masked, and then regener-

ated based on their part-of-speech (POS) n-gram and ranked by an unexpectedness score. To generate a hyperbole, the model masks a non-hyperbole sentence span using a POS n-gram and then selects the highest-ranked of multiple possible hyperbole candidates using a BERT-based hyperbole ranker. HypoGen (Tian, Krishna Sridhar, and Peng 2021) focuses on clause-level hyperboles with a specific pattern of the form (A1 is so A2 that B is D). To train this model, the authors collected and annotated a dataset of hyperboles from Reddit posts (HYPO-Red), selecting a subset that contains the target pattern and analyzing the relationships between clauses. They then used COMET and reverse-COMET models to generate candidate clauses with commonsense and counterfactual relations, respectively, and used BERT-based and neural-based classifiers to select the top- $k$  candidates.

**Sarcasm:** To generate sarcasm, Chakrabarty et al. (2020) negate or replace evaluative words with their antonyms and provide a context to emphasize the semantic incongruity between the intended sarcasm and the context. Several context candidates are retrieved using a concept derived from the COMET model and an online corpus. The candidates are ranked by a RoBERTa model for semantic incongruity and the highest scored are concatenated to the sarcastic sentence. Oprea, Wilson, and Magdy (2021) argue that it is not sufficient to only negate the literal meaning of a sentence to produce sarcasm and propose a framework that is based on the implicit display theory of sarcasm to generate a sarcastic response to an input. The framework identifies an ironic environment by negating the event in the input utterance. A COMET-based rule-based method is used to produce an insincere negative attitude response based on the input event and one of the following relations: the action needed to perform the event, the attribute needed to perform the action, the user reaction, and the effect of the action on the user.

**Puns:** Yu, Tan, and Wan (2018) use a conditional LSTM model to generate homographic puns given two senses of a pun word. They use a joint beam search decoding with a forward and backward generation centered by the target pun word. Both sequences are then concatenated to form the final pun sentence. AMBIPUN (Mittal, Tian, and Peng 2022) generates homographic pun sentences using GPT-3 to generate related context words given two different senses of a pun word. The context words are then combined using a T5 model to generate candidate sentences and a BERT-based model to rank and pick a final pun sentence.

**Multi-Figurative:** Creative writing tasks need to employ various literary techniques in the generation. It is more practical to have one multi-figurative generation model to do so. Lai and Nissim (2022) propose a multi-figurative model that can transform between literal or figurative forms to another figurative form. The authors first finetune a BART model on a denoising objective to infuse multi-figurative sense into the model, then they fine-tune the model on a literal to figurative paraphrasing objective with a parallel data corpus and another time with a figurative to figurative paraphrasing objective with a cross attention layer to leak the target figure of speech to the encoder. For inference, they either transform

the input text to the target figure of speech directly or to a literal form first and then to the target figurative form.

## Impact on NLG Downstream Tasks

Generating creative language plays an essential role in many text-to-text NLG tasks, especially in creative writing tasks such as poetry and prose novels or stories. It also improves the quality of downstream tasks such as machine translation, dialogue generation, text summarization, style transfer, and even other data-to-text applications.

**Storytelling** is to generate an open-ended text that conveys to the reader a comprehensive story. It is one of the creative writing tasks written in prose form as opposed to poetry. Storytelling needs to account for soft and hard constraints such as the topics, plots and characters among other aspects to keep the story consistent and coherent. AI-aided approaches were used (Goldfarb-Tarrant, Feng, and Peng 2019) to collaboratively write and plan story plots. Plug-and-Play Language Model (PPLM) (Dathathri et al. 2019) is used for controllable story generation by plugging an attribute classifier on top of a GPT-2 transformer that checks how similar the next token is to a given topic. The classifier will be exposed to the history of latent representations of the generated words, then repeatedly perform backward and forward passes through the classifier and the generator calculating gradients each time, updating the representations, and handing them back to the generator. This will steer the generation process towards the topics at inference time with minimal training. The work was limited to assistive stories generation, completing a story skeleton given a story theme or sentiment. Brahman and Chaturvedi (2020) introduce multiple emotion-aware GPT-2-based models coupled with the COMET model to generate stories given a title and an emotional arc of the protagonist. Their work was the first to generate stories with an emotional trajectory plan.

**Text Style Transfer** is gaining an increasing popularity in natural language generation (Garbacea and Mei 2020). Variational Auto-Encoders (VAEs) were used to rewrite modern text in Shakespeare style (Mueller, Gifford, and Jaakkola 2017). Riedl (2020) used a transformer-based model (XLNet) to generate a parody of lyrics, changing the lyrics while preserving the rhythm and syllable count.

**Machine Translation** Low (2011) examines the challenges and strategies involved in translating jokes and puns. The author argues that while some forms of humor are easily translatable, others, such as those that rely on wordplay and cultural references, can be particularly difficult. To effectively translate this kind of humor, the paper suggests using a variety of strategies, such as sense transferring, semantic leaps and cultural substitution.

**Dialogue Generation** can be modulated to generate creative dialogue content such as improv games, battle rapping, interactional jokes and conversational narratives, among others. In fact, recent chatbots such as LaMDA (Thoppilan et al. 2022) and ChatGPT<sup>1</sup> have shown the ability to write

---

<sup>1</sup>[openai.com/blog/chatgpt/](https://openai.com/blog/chatgpt/)

and complete jokes, poems, and stories in conversations.

**Text Summarization** is the task of condensing a longer text preserving the most important information within its content. Similarly, creative language generation can be seen as summarizing broad ideas and beautiful imagery using words. In fact, iPoet (Yan et al. 2013) formulates the poetry generation task as a text summarization task. Given users' intents, write a poem that obeys poem requirements by retrieving terms out of a poetry corpora.

**Data-to-text generation:** Creative data-to-text tasks are analogous to how humans describe, in creative writing, what they see or hear from natural beauty, scenery, natural soundscape, etc. Loller-Andersen and Gambäck (2018) used Inception, ConceptNet and LSTM models to generate image-inspired poems. Liu et al. (2018) proposed a multi-adversarial CNN-RNN-based GAN model to caption images with a poem. Uehara et al. (2022) proposed transformer-based encoder and decoder models to generate emotional narratives from visual embedding extracted from images using CNNs. Achlioptas et al. (2021) demonstrated the power of utilizing their explained emotion-captioned image data set in enhancing language models to express and explain emotions triggered by artistic images. In addition, Chen and Lerch (2020) proposed a SeqGAN-based model that generates a matching line of lyrics with an input melody.

## Poetry Generation

Modeling poetry is more difficult than language modeling as it requires the machine not only to understand but also to use language as a creative tool. As previously discussed, creativity is not yet well-defined, and it is uncertain if computers can achieve it.

**Definition** Before discussing how to generate poetry, we first need to define poetry, which is equally difficult as defining creativity itself. In fact, there are as many ways to describe poetry as there are people (Murfin and Ray 2009). Poetry is often defined as a type of writing that uses distinctive style, rhythm, and language to convey intense feelings and ideas <sup>2</sup> or the writing that concentrates imaginative experience to elicit a specific emotional response through meaning, rhythm, and sound patterns <sup>3</sup>. According to Milic (1970), poetry is the writing that violates the logical sequence and semantic categories of prose. Both prose and poetry may use literary devices such as metaphors, similes, ironies, and puns to express ideas and evoke emotions. Poetry, however, employs specific, but not essential, devices such as rhyme, rhythm, and meter. Meanwhile, the most notable devices found in a poem are its versification and form coherence.

**Creativity Dimensions** Various definitions of poetry highlight the three dimensions of creativity previously mentioned, delivered through language. A poem is a language artifact that requires the use of language to express meaning, and thus must adhere to linguistic rules to some extent while also allowing for violations, commonly referred to as

poetic license. Originality and unpredictability can be observed through the surprising deviation from prose and normal speech rules, as well as through the use of poetic devices and euphony. The social aspect of poetry can be emphasized through the concentration of ideas, imagery and emotions within the boundaries of verses and stanzas, as well as the ability for the poem to be subject to criticism.

**Forms** English poetry can be classified based on various poetic characteristics. Notably, rhythm and rhyme schemes are often used to classify poems. The most popular forms include: (1) *Free Verse*: which has no constraints on a specific form. (2) *Haiku*: a short form of poetry that follows a 5-7-5 syllabic pattern. (3) *Sonnet*: characterized by specific rhyme and meter schemes. Shakespearean sonnet is considered one of the most well-known with 3 quatrains and a couplet: ABAB CDCD EFEF GG rhyme scheme. (4) *Blank Verse*: consists of unrhymed but metered verses. (5) *Limerick*: consists of five lines with seven to ten syllables with a verbal rhythm and the first, second, and fifth lines rhyming.

**Goals** The most common goals of writing poetry according to Preminger (2016) are to imitate reality, attain special effects on the readers, communicate emotions, and be art for art's sake. Similarly, the automatic poetry generation must represent adequate computational goals. Milic (1970) upheld in an early view that the possible usefulness of computer poetry is to influence the doer (designer/end-user) to learn more about language, poets and poetry.

## Poetry Generation Techniques

Automatic poetry generation was documented as early as 1959 with a word salad approach by manually enhancing permutations of poem words (Lutz 1959). Since the few similar early attempts, the automatic poetry generation topic was inactive until the 1990s, where some works started getting attention in the field (Oliveira 2009). Early poetry generation techniques can be categorized (Gervás 2002) into: template-based, generate-and-test, evolutionary, and case-based reasoning. Manurung (2003) groups poetry generation approaches into: word salad, template/grammar-based, form-aware and poetry generation systems. Manurung argues that the first three methods do not satisfy the definition of poetry generators being: aware of meaningfulness, grammaticality and poeticness. Lamb, Brown, and Clarke (2017) propose a generic categorization of poetry generation methods that works on any creative task: mere, human-enhanced, and computer-enhanced approaches. Oliveira (2017) reviews poetry generation methods from multiple intersecting angles: languages, forms, content quality, techniques, and evaluation. In addition, Oliveira extends the list of techniques with: chart generation, constraint satisfaction, multi-agent, and language models such as Markov and RNN models. In this paper, we use a different taxonomy based on the intersection of chronology and methodology, pointing to the proposed creativity dimensions whenever applicable. We have found that the development of poetry generation techniques starts with rule-based, heuristic-based, statistical-based, then deep-learning-based approaches that are roughly proposed in this chronological order (Fig. 2).

<sup>2</sup>oed.com/view/Entry/146552

<sup>3</sup>merriam-webster.com/dictionary/poetry

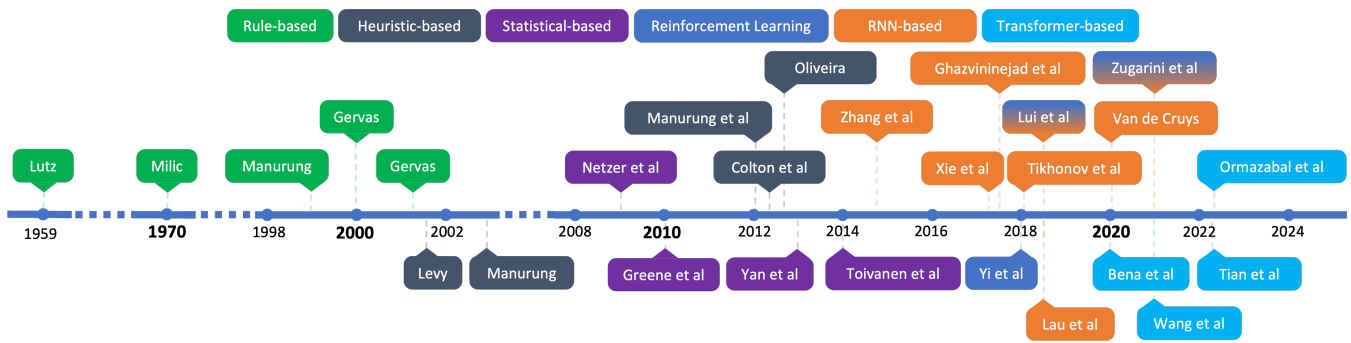


Figure 2: Poetry Generation Timeline.

**Rule-based Approaches** The early methods are based on following crafted rules and filling poem templates. Templates are the poem shells where actual words are taken out from original poems, while some of their characteristics, such as part of speech, are preserved as placeholders to be systematically filled with other words matching the characteristics. RETURNER (Milic 1970) is one of the earliest programs used to automatically generate English poems. The author designed three versions of a template-based model to generate three poems. The first version generates a poem given only the vocabulary of a real poem “Return by Alberta Turner.” The second version was based on subjects and verbs with randomly decided optional modifiers, complements, and end conjunctions to provide the opportunity for iteration. The third version is based on a six-slot grammatical matrix and a set of rules to guide the generation process. Manurung (1999) uses a form-aware chart-based generation to generate strings that satisfy rhythm constraints. Meter validation is performed at each step of the process to reduce the search space. WASP (Gervás 2000) fills Spanish poem templates based on initial vocabulary and verse patterns, generating and validating verse drafts at each iteration according to length, rhyme, and rhythm constraints. The main focus of these early systems is on the poetic form rather than on the creative language. ASPERA (Gervás 2001) is an evolved version of WASP with case-based reasoning to retrieve an existing form and vocabulary that are highly similar to the user-defined target specifications and then adapt to meet the desired specifications. However, this model is not fully automated and needs to manually interact with a human user to validate or correct the generated draft.

**Heuristic-based Approaches** Using rule-based approaches is advantageous as they provide control and adherence to specific poetic forms or templates, but they are very predictable. Heuristic methods, however, are less predictable, and unpredictability is essential to creative writing. Levy (2001) proposes a poet-critic model theory that consists of a computer-based generator and an evaluator module. The author did not implement the described theory but proposed a prototype that utilizes the concept of evolution to create original outputs and uses classical neural networks as evaluators trained on real-world human evaluations. The evolutionary algorithm is a meta-heuristic approach in which the best population of poems is chosen

by evaluator modules, modified by crossover operators and mutation operators and continuously evaluated until a satisfactory result is obtained. Levy uses such an approach to enhance poem features, words and rhymes. Manurung (2003) uses a similar approach with the following operations: add, delete, or edit as forms of mutation. In a later update of the framework, Manurung, Ritchie, and Thompson (2012) point to the fact that abusing the poetic license may lead to non-sense in a poetic form and that poetry generation approaches must conform to a restricted definition of poetry to generate falsifiable output. Similarly, Oliveira (2012) uses extracted semantic networks, grammar generators, and syllabic templates to generate Portuguese poems using genetic algorithms. Colton, Goodwin, and Veale (2012) address the lack of story behind the generated poems. The authors avoid to retrieve from a corpus of existing poems to restrain plagiarism, instead they choose short similes and non-poetic text corpora. They combine extracted key phrases from the non-poetic corpus with produced variations of existing similes to fill a user-given template. They also attempt to automatically generate a commentary on the generation process by recording context statements at each step of the generation.

**Statistical Approaches** A problem in heuristic-based models is that they are not capable of modeling the language. Netzer et al. (2009) uses statistical methods to analyze a corpus of English Haikus and generate words related to a theme provided by a user. The authors pick a real Haiku poem POS template at random and fill-in words from the theme set. The sentences are then ranked according to the degree of association of words. In order to augment the unpredictability of the output, they give more weight to pick second degree association instead of first degree. Greene, Bodrumlu, and Knight (2010) focus on the use of statistical methods to analyze, generate and translate poems. They use finite state transducers to map English sentences to a sequence of syllables and perform unsupervised learning to analyze the stress patterns of words in a poetry corpus. They used a finite state cascade to generate English love poems. In addition, the authors used a statistical phrase-based machine translation to translate Italian poems into English with the help of an iambic pentameter model. Poetry generation was also formulated as a text summarization task (Yan et al. 2013): Given the intentions of a user, write a poem that



meets the poem requirements by retrieving terms from a poetry corpora. The authors use generative summarization and aim to produce correlated verses taking into consideration the poetic form and semantic coherence. Word associations were also used to fill poetry templates (Toivanen et al. 2014) replacing words from a poem with other words based on syntactic similarity. They use the templates of poems from the Imagist movement to emphasize figurative language.

**Deep Learning Approaches** Chomsky criticized statistical models as incomprehensible and inefficient compared to humans (Norvig 2017). The substantial amount of digital data created each second, the recent advances in computational power, and the ability of deep learning methods to derive complex data correlations, have led to state-of-the-art performance in NLG, including poetry generation.

• **RNN-based** RNNs were first introduced by John Hopfield in the early 1980s. However, they became popular as language models only a decade ago (Mikolov et al. 2010) for their ability to model long-term dependencies in sequential data such as text. Zhang and Lapata (2014) use a Convolutional Sentence Model (CSM) to capture sentence characteristics and use a character-based Recurrent Neural Network (RNN) model to incrementally generate poems. A set of keywords provided by the user is expanded with a poetic phrase taxonomy to generate candidate phrases, the top-ranked phrase will be the first line in the poem, and the following lines are sequentially generated given the previous line representations. The training process considers both the poetic form and semantics. Xie, Rastogi, and Chang (2017) designed two RNN-based models where the first combines char-level and word-level LSTMs with GloVe word embeddings for the model to capture not only the semantics of the words, but also morphemes and rhymes during training. The second model uses CNNs to generate word embeddings. Both models showed significant performance in coherence, poeticness, and form characteristics over vanilla char-based and word-based RNN models. Ghazvininejad et al. (2017) proposed a sonnet generation system based on RNNs and finite state acceptors and used embedding rhymes with a word2vec model. The system is interactive and allows the user to input topic words, adjust style configurations, and evaluate the generated poem with a 5-star rating; the evaluation was used to update the system performance. Lau et al. (2018) proposed a multi-model sonnet generator trained on Shakespeare’s sonnets by means of bidirectional word-based and char-based LSTMs with the attention mechanism. The model consists of language, rhythm and rhyme submodels with Bi-LSTMs with an attention mechanism; the paper showed that Bi-LSTMs can capture poetic forms such as rhythm and rhyme very efficiently. Tikhonov and Yamshchikov (2018) used phoneme-based and char-based Bi-LSTMs, word-based LSTM, word embeddings and document embeddings for an author-stylized multilingual text generation framework with an application on English and Russian poems. Van de Cruys (2020) proposed an encoder-decoder model with double-layered GRUs using the attention mechanism. The decoder generated English and French poems line by line starting with the rhyming words back-

ward to ensure a coherent verse with the forced rhyming scheme. To show the model’s capability of presenting poeticness from scratch, the authors trained their model on non-poetic corpora. They proposed a latent semantic model to ensure topic coherence and a global optimization framework to score the verses and keep the highest-scoring ones.

• **RNNs and reinforcement learning (RL)** RNNs are able to handle input sequences of variable length, but they may suffer from gradient vanishing during backpropagation, which can cause the loss of distant context in the generated text. Additionally, Yi et al. (2018) argue that models based on maximum likelihood estimation, such as RNNs, tend to only optimize token-level loss rather than considering the poem as a whole, as humans do. To address this, they applied mutual RL to generate classic Chinese poems. The authors presented several reward modules instead of maximum likelihood to better mimic human behavior with reward schemes for fluency, coherence, meaning, and overall quality. Zugarini et al. (2021) used deep RL by combining Bi-LSTM models and RL in an iterative refinement approach to generate and revise poems following an author style and a rhyme scheme. Liu et al. (2018) also proposed a deep RL using a multi-adversarial CNN-RNN-based GAN model to caption images with a free verse English poem.

• **Transformer-based** Transformers (Vaswani et al. 2017) were introduced to overcome RNNs limitations caused by their sequential design, by allowing parallelization of training. Instead of processing a sequence token by token, sequences are processed holistically with positional encoding. Additionally, transformers introduced self-attention, allowing inputs-to-inputs and outputs-to-outputs attention to enhance context encoding and decoding. Bena and Kalita (2019) fine-tuned GPT-2 on a dream corpus to endow imagery language and fine-tuned it again on various emotion-based subcorpora to invoke one of the following emotions on the reader: joy, trust, anticipation, anger, fear, surprise, sadness, and disgust. LimGen (Wang et al. 2021) used GPT-2 to predict the next words in multi-candidate templates extracted from human-written limerick, then filtered out the words for meter and rhyme consistency, and output the top N lines using a variation of beam search that calculated a diversity score between templates. Ormazabal et al. (2022) proposed a structure-aware training of an autoregressive transformer model to generate formal verse poems in Spanish and Basque with only non-poetic corpora. Similarly, Tian and Peng (2022) proposed a multimodel framework trained on non-sonnet corpora to generate sonnets. The authors separated the training from the decoding and utilize a series of content planning, rhyme pair generation, polishing and constrained decoding to generate sonnets.

## Recommendations

A major concern in rule-based and heuristic-based methods is the originality of language output, particularly when using poem templates and simple synonym replacements. Similar concerns apply to models trained on poetry corpora and those using transfer learning to limit mimicking the language, style, theme, and semantics of the source. Further

study on active divergence in NLG can potentially enhance originality, enabling the production of creative content that incorporates novel elements. Another important area is optimizing decoding strategies through the use of enhanced sampling techniques, which can increase the unpredictability factor of the generated output. Recent advancements in improving nucleus sampling with randomized heads and introducing new sampling methods (Zhang et al. 2021b) have shown promising results in terms of diversity and novelty of generated texts. To address the sociability aspect of poetry, we recommend the utilization of iterative refinement models guided by discriminator models or reinforcement learning models to provide feedback during the generation process in a poet-critic-like framework. Additionally, we recommend developing creative LLMs that integrate elements of poetic diction during pretraining and incorporating multi-figurative generation models to aid in effective poetry generation.

## Conclusion

In this work, we bring attention to the problem of generating creative data for researchers interested in the field. We analyze natural language generation models based on the three dimensions of creativity we proposed: originality, sociability and unpredictability. We provide a brief overview of text generation tasks and their potential creative applications, as well as a review of recent works. Finally, we provide a comprehensive overview examination of the task of poetry generation, including the methodologies and models employed in the literature. We hope that this review encourages researchers and provides insights for those interested in creativity in natural language generation and poetry, particularly in considering the use of suggested creativity and poeticness criteria in language models.

## Acknowledgements

We thank Michael Ulliyot for his insightful feedback and the anonymous reviewers for their valuable input.

## References

Achlioptas, P.; Ovsjanikov, M.; Haydarov, K.; Elhoseiny, M.; and Guibas, L. J. 2021. Artemis: Affective language for visual art. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11569–11579.

Bena, B., and Kalita, J. 2019. Introducing aspects of creativity in automatic poetry generation. In *Proceedings of the 16th International Conference on Natural Language Processing*, 26–35.

Berns, S., and Colton, S. 2020. Bridging generative deep learning and computational creativity. In *Proceedings of the Eleventh International Conference on Computational Creativity*, 406–409.

Boden, M. A. 2004. *The creative mind: Myths and mechanisms*. Routledge.

Bosselut, A.; Rashkin, H.; Sap, M.; Malaviya, C.; Celikyilmaz, A.; and Choi, Y. 2019. Comet: Commonsense transformers for knowledge graph construction. In *Association for Computational Linguistics (ACL)*.

Brahman, F., and Chaturvedi, S. 2020. Modeling protagonist emotions for emotion-aware storytelling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5277–5294.

Brooks, J., and Youssef, A. 2021. Got: Testing for originality in natural language generation. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, 68–72.

Chakrabarty, T.; Ghosh, D.; Muresan, S.; and Peng, N. 2020.  $r^3$ : Reverse, retrieve, and rank for sarcasm generation with commonsense knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7976–7986.

Chakrabarty, T.; Zhang, X.; Muresan, S.; and Peng, N. 2021. Mermaid: Metaphor generation with symbolism and discriminative decoding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4250–4261.

Chakrabarty, T.; Muresan, S.; and Peng, N. 2020. Generating similes effortlessly like a pro: A style transfer approach for simile generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6455–6469.

Chen, Y., and Lerch, A. 2020. Melody-conditioned lyrics generation with seqgans. In *2020 IEEE International Symposium on Multimedia (ISM)*, 189–196.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-face poetry generation. In *Proceedings of 3rd International Conference on Computational Creativity*, 95–102.

Colton, S.; Wiggins, G. A.; et al. 2012. Computational creativity: The final frontier? In *Ecai*, volume 12, 21–26. Montpellier.

Crimp, D. 1981. The end of painting. *October. Art World Folies (Spring, 1981)* 16:69–86.

Csikszentmihalyi, M. 1997. Flow and the psychology of discovery and invention. *HarperPerennial, New York* 39.

Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; and Liu, R. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Garbacea, C., and Mei, Q. 2020. Neural language generation: Formulation, methods, and evaluation. *arXiv preprint arXiv:2007.15780*.

Gervás, P. 2000. Wasp: Evaluation of different strategies for the automatic generation of spanish verse. In *Proceedings of the AISB-00 symposium on creative & cultural aspects of AI*, 93–100.

Gervás, P. 2001. An expert system for the composition of formal spanish poetry. In *Applications and Innovations in Intelligent Systems VIII*. Springer. 19–32.

Gervás, P. 2002. Exploring quantitative evaluations of the creativity of automatic poets. In *Workshop on Creative Systems, Approaches to Creativity in Artificial Intelligence and Cognitive Science, 15th European Conference on Artificial Intelligence*.



- Ghazvininejad, M.; Shi, X.; Priyadarshi, J.; and Knight, K. 2017. Hafez: an interactive poetry generation system. In *Proceedings of ACL 2017, System Demonstrations*, 43–48.
- Goldfarb-Tarrant, S.; Feng, H.; and Peng, N. 2019. Plan, write, and revise: an interactive system for open-domain story generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 89–97.
- Greene, E.; Bodrumlu, T.; and Knight, K. 2010. Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, 524–533.
- Gu, J.; Wang, C.; and Zhao, J. 2019. Levenshtein transformer. *Advances in Neural Information Processing Systems* 32.
- Hertzmann, A. 2018. Can computers create art? *Arts* 7(2):18.
- Hockney, D. 2006. *Secret knowledge: Rediscovering the Lost Techniques of the Old Masters*. London: Thames & Hudson, new and expanded ed edition. OCLC: ocm70399577.
- Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2020. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.
- Jahanian, A.; Chai, L.; and Isola, P. 2019. On the steerability of generative adversarial networks. *arXiv preprint arXiv:1907.07171*.
- Keukeleire, P. 2020. Correspondence between perplexity scores and human evaluation of generated tv-show scripts. *M.S. thesis, Dept. Comput. Sci. Eng., Delft Univ. Technol., Delft, The Netherlands*.
- Lai, H., and Nissim, M. 2022. Multi-figurative language generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, 5939–5954.
- 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.
- Lau, J. H.; Cohn, T.; Baldwin, T.; Brooke, J.; and Hammond, A. 2018. Deep-speare: A joint neural model of poetic language, meter and rhyme. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1948–1958.
- Levy, R. P. 2001. A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proceedings of the ICCBR-01 Workshop on Creative Systems*.
- Liu, B.; Fu, J.; Kato, M. P.; and Yoshikawa, M. 2018. Beyond narrative description: Generating poetry from images by multi-adversarial training. In *Proceedings of the 26th ACM international conference on Multimedia*, 783–791.
- Loller-Andersen, M., and Gambäck, B. 2018. Deep learning-based poetry generation given visual input. In *Proceedings of the Ninth International Conference on Computational Creativity*, 240–247.
- Low, P. A. 2011. Translating jokes and puns. *Perspectives: Studies in Translatology* 19(1):59–70.
- Lutz, T. 1959. Stochastische texte. *augenblick* 4(1):3–9.
- Manin, D. 2021. Running in shackles: The information-theoretic paradoxes of poetry. In *Handbook of the Mathematics of the Arts and Sciences*. Springer. 1001–1014.
- Manurung, R.; Ritchie, G.; and Thompson, H. 2012. Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence* 24(1):43–64.
- Manurung, H. M. 1999. A chart generator for rhythm patterned text. In *Proceedings of the First International Workshop on Literature in Cognition and Computer*, 15–19.
- Manurung, H. 2003. An evolutionary algorithm approach to poetry generation. *Ph.D. thesis, University of Edinburgh*.
- Markowitz, S. J. 1994. The distinction between art and craft. *Journal of Aesthetic Education* 28(1):55–70.
- Mikolov, T.; Karafiát, M.; Burget, L.; Černocký, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.
- Milic, L. T. 1970. The possible usefulness of poetry generation. *The Computer in Literary and Linguistic Research*, ed. R. A. Wisbey. 169–82.
- Mittal, A.; Tian, Y.; and Peng, N. 2022. Ambipun: Generating humorous puns with ambiguous context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1053–1062.
- Mueller, J.; Gifford, D.; and Jaakkola, T. 2017. Sequence to better sequence: continuous revision of combinatorial structures. In *International Conference on Machine Learning*, 2536–2544. PMLR.
- Murfin, R. C., and Ray, S. M. 2009. *The Bedford Glossary of critical and literary terms*. Bedford/St. Martins.
- Netzer, Y.; Gabay, D.; Goldberg, Y.; and Elhadad, M. 2009. Gaiku: Generating haiku with word associations norms. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, 32–39.
- Norvig, P. 2017. On chomsky and the two cultures of statistical learning. *Berechenbarkeit der Welt?* 61–83.
- Oliveira, H. 2009. Automatic generation of poetry: An overview. *Universidade de Coimbra*.
- Oliveira, H. G. 2012. Poetryme: A versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence* 1:21.
- Oliveira, H. G. 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.
- Oprea, S.; Wilson, S.; and Magdy, W. 2021. Chandler: An explainable sarcastic response generator. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 339–349.
- Ormazabal, A.; Artetxe, M.; Agirrezabal, M.; Soroa, A.; and Agirre, E. 2022. Poelm: A meter-and rhyme-controllable language model for unsupervised poetry generation. *arXiv preprint arXiv:2205.12206*.

- Paisley, W. J. 1966. The effects of authorship, topic, structure, and time of composition on letter redundancy in english texts. *Journal of verbal learning and verbal behavior* 5(1):28–34.
- Preminger, A. 2016. *Princeton Encyclopaedia of Poetry and Poetics*. Macmillan International Higher Education.
- Reiter, E. 2016. Ehud reiter’s blog. [www.ehudreiter.com/2016/12/18/nlg-vs-templates/](http://www.ehudreiter.com/2016/12/18/nlg-vs-templates/). [Online; accessed 10-Jan-2023].
- Riedl, M. 2020. Weird ai yankovic: Generating parody lyrics. *arXiv preprint arXiv:2009.12240*.
- Shannon, C. E. 1951. Prediction and entropy of printed english. *Bell system technical journal* 30(1):50–64.
- Shneiderman, B. 2007. Creativity support tools: accelerating discovery and innovation. *Communications of the ACM* 50(12):20–32.
- Steadman, P. 2002. *Vermeer’s Camera: Uncovering the Truth Behind the Masterpieces*. Oxford University Press, USA.
- Stowe, K.; Chakrabarty, T.; Peng, N.; Muresan, S.; and Gurevych, I. 2021. Metaphor generation with conceptual mappings. *arXiv preprint arXiv:2106.01228*.
- Teller, R. J. 2014. *Tim’s Vermeer*. [Film]. Sony Pictures Classics.
- Thoppilan, R.; De Freitas, D.; Hall, J.; Shazeer, N.; Kulshreshtha, A.; Cheng, H.-T.; Jin, A.; Bos, T.; Baker, L.; Du, Y.; et al. 2022. Lambda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Tian, Y., and Peng, N. 2022. Zero-shot sonnet generation with discourse-level planning and aesthetics features. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3587–3597.
- Tian, Y.; krishna Sridhar, A.; and Peng, N. 2021. Hypogen: Hyperbole generation with commonsense and counterfactual knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 1583–1593.
- Tikhonov, A., and Yamshchikov, I. P. 2018. Guess who? multilingual approach for the automated generation of author-stylized poetry. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, 787–794. IEEE.
- Toivanen, J.; Gross, O.; Toivonen, H.; et al. 2014. The officer is taller than you, who race yourself!: Using document specific word associations in poetry generation. In *Proceedings of the Fifth International Conference on Computational Creativity*. Jožef Stefan Institute.
- Tong, X.; Shutova, E.; and Lewis, M. 2021. Recent advances in neural metaphor processing: A linguistic, cognitive and social perspective. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4673–4686.
- Uehara, K.; Mori, Y.; Mukuta, Y.; and Harada, T. 2022. Vinter: Image narrative generation with emotion-arc-aware transformer. *arXiv preprint arXiv:2202.07305*.
- Van de Cruys, T. 2020. Automatic poetry generation from prosaic text. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2471–2480.
- Vaswani, A.; Shazeer, N.; Parmar, N.; G. A. N. Uszkoreit, J.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.
- Wang, J.; Zhang, X.; Zhou, Y.; Suh, C.; and Rudin, C. 2021. There once was a really bad poet, it was automated but you didn’t know it. *arXiv preprint arXiv:2103.03775*.
- Xie, S.; Rastogi, R.; and Chang, M. 2017. Deep poetry: Word-level and character-level language models for shakespearean sonnet generation. In *Natural Lang. Process. Deep Learn.*
- Xu, W., and Carpuat, M. 2021. Editor: an edit-based transformer with repositioning for neural machine translation with soft lexical constraints. *Transactions of the Association for Computational Linguistics* 9:311–328.
- Yan, R.; Jiang, H.; Lapata, M.; Lin, S.-D.; Lv, X.; and Li, X. 2013. I, poet: automatic chinese poetry composition through a generative summarization framework under constrained optimization. In *Twenty-Third International Joint Conference on Artificial Intelligence*.
- Yi, X.; Sun, M.; Li, R.; and Li, W. 2018. Automatic poetry generation with mutual reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3143–3153.
- Yin, Z.; Reuben, F.; Stepney, S.; and Collins, T. 2021. “a good algorithm does not steal—it imitates”: The originality report as a means of measuring when a music generation algorithm copies too much. In *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*, 360–375. Springer.
- Yu, Z.; Tan, J.; and Wan, X. 2018. A neural approach to pun generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1650–1660.
- Zhang, X., and Lapata, M. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 670–680.
- Zhang, Y., and Wan, X. 2022. Mover: Mask, over-generate and rank for hyperbole generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 6018–6030.
- Zhang, J.; Cui, Z.; Xia, X.; Guo, Y.; Li, Y.; Wei, C.; and Cui, J. 2021a. Writing polishment with simile: Task, dataset and a neural approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35 issue 16, 14383–14392.
- Zhang, X.; Sun, M.; Liu, J.; and Li, X. 2021b. Lingxi: A diversity-aware chinese modern poetry generation system. *arXiv preprint arXiv:2108.12108*.
- Zugarini, A.; Pasqualini, L.; Melacci, S.; and Maggini, M. 2021. Generate and revise: Reinforcement learning in neural poetry. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.