Why try to build try to build a co-creative poetry system that makes people feel that they have “creative superpowers”?*

Ibukun Olatunji*

Computational Foundry, Swansea University, Crymlyn Burrows, Skewen, Swansea, United Kingdom, SA1 8EN

Abstract

The paper examines co-creative writing systems, and argues that existing Large Language Models could potentially reduce human capacity. Furthermore, existing sociocultural inequalities might be exacerbated by the widespread adoption of such generative systems. The paper instead suggests a custom approach, using co-creative poetry writing as an example. The system has architectural changes from typical language models to better support poetry. It also uses rap lyrics as part of the training data in order to help reduce sociocultural bias. A high level system implementation is proposed along with some evaluation methods. Evaluation is based on expert judgement on final outputs, and user performance on language tasks associated with human creativity. The final section of the paper explores how and why alternatives to existing co-creative systems could benefit individual users as well as wider society.

Keywords

Creativity, poetry, co-creativity, natural language processing, language models, writing support tools, data sets,

1. Introduction

This paper examines co-creative systems using poetry writing as an example. Within the paper ’poetry’ includes song lyrics. Section one of the paper explores poetry in terms of human creativity. Poetry is chosen as it is a creative task that non-expert humans can outperform machines on vs creative outputs such as image generation. After introducing the case for poetry, there is an exploration of recent work in generative computational systems. As well as being the technical state of the art, these systems provide a conceptual framework to explore sociocultural issues such as bias and inclusion. Section one then explores a range of poetry-specific systems and ends with a more detailed case study. The case study examines a system that combines elements of more powerful general models and custom architectural features specific to poetry writing. Section two details the evaluation issues and methods that might be employed for the proposed co-creative system. The emphasis on this section is on how to evaluate human improvement over time. Section three explores a high level implementation of the system. It builds on the evaluation to propose both an architecture and a method to testing if the proposed system has, in principal, any benefits over and above those described in section one. Section four is a discussion of the social and cultural limitations of current generative systems. It expands on section one in exploring bias and proposes a mitigation through the use of rap lyrics. Section five describes the theoretical and practical limitations of the paper as well as future work. Section six provides a summary of the paper’s contribution. The section ends with answers to the question: why try to build try to build a co-creative poetry system that makes people feel that they have “creative superpowers”?

1.1. Human Creativity

Human creativity is the ability to come up with ideas or artefacts that are new, surprising, and valuable. Rather than a solitary act, it results from the interaction of social elements; a culture that contains symbolic rules, a person who brings novelty into the symbolic domain, and people who recognise and validate the innovation. [1, 2, 3]. Boden makes a further distinction between *psychological* and *historical* creativity (P-creativity and H-creativity). P-creativity involves coming up with an idea that’s new to the person who comes up with it. H-creativity means that (so far as we know) no-one else has had it before: it has arisen for the first time in human history [2, 4]. Machine learning models have the potential to support human creativity [5, 6, 7]. However, questions remain on their design and influence in augmenting human capacity as opposed to reducing it [8, 9, 10]. Shneiderman suggests that "researchers’ goals shape the questions they raise, collaborators they choose, methods they use, and outcomes of their work."[11]. This leads to the question: how can designers of programming interfaces, interactive tools, and rich social environments enable more people to be more creative more often? [12]

Joint Proceedings of the ACM IUI Workshops 2023, March 2023, Sydney, Australia

[*] Ibukun Olatunji. 2023. Why try to build a co-creative poetry system that makes people feel that they have “creative superpowers”? In Joint Proceedings of the ACM IUI 2023 Workshops. Sydney, Australia, 13 pages.

[*] Corresponding author.

i.o.olatunji.2030349@swansea.ac.uk (I. Olatunji)

CEUR Workshop Proceedings (CEUR-WS.org)

Language Model Characteristics

Table 1
Summary of State-of-the-Art Language Models by Size, Model Type and Ownership

Model Name	Parameters	Model Type	Owner
BERT	110 - 340 million	Transformer	Google
GPT-2	1.5 billion	Transformer	OpenAI
LaMDA	137 billion	Transformer	Google
GPT-3	175 billion	Transformer	OpenAI
ChatGPT/InstructGPT	175 billion	Transformer	OpenAI
BLOOM	176 billion	Transformer	BLOOM Project
Megatron-Turing NLG	530 billion	Transformer	Microsoft and NVIDIA
PaLM	540 billion	Transformer	Google
GLaM	1 trillion	Mixture of Experts	Google

1.2. Computational Systems

In computational terms, automated systems are now capable of writing poetry approaching human levels [13, 14, 15]. Karimi et al consider three three main strategies by which the role of humans in creative systems can be characterized: fully autonomous systems, creativity support tools, and co-creative systems [16]. Although the paper is primarily concerned with co-creative systems, it will to blend the categories where necessary. The reasoning for this is that the human users do not make the same distinctions; also, the features and usage are often blended in the real-world, e.g an autonomous system that is used by a creator as an input and thus becomes a support tool and/or co-creative system [10]. The next section briefly outlines the state of the art in computational writing systems.

Language models (LMs) refer to systems that are trained on string prediction tasks: predicting the likelihood of a token (character, word or string) given either the preceding context or its surrounding context. Such systems are unsupervised and when deployed, take text as input, and output scores or string predictions [17]. Large Language Models (LLMs) trained on sufficiently large and diverse data sets are able to perform well across domains and there is a correlation between model performance and size [18]. State-of-the-art models are able to generate text that approach or surpasses that of *some* humans[13, 14, 15, 19]. The emphasis on *some* humans is an important with respect to user characteristics; in broad terms, humans co-creating poetry can be considered as either inexperienced or advanced users. Research on creative tasks such as improvisation suggests that users vary in cognitive processing based in part on their experience and skills levels[20, 21]. A well-designed co-creative system should therefore take differences in user support needs into account [8, 9, 22].

1.3. General Purpose Language Generation

LLMs are trained to predict the next word, or series of words, in a a text sequence. They model text corpora as probability distributions. Users write a short text prompts to tell the system what to generate. Depending on how many examples are provided in the text prompt, the system is referred to as zero-, one-, and few-shot learning [13, 15, 17]. Pretrained language models have become a cornerstone of modern natural language processing (NLP) pipelines because they often produce better performance from smaller quantities of labeled data [23]. Within general LLMs, the transformer has established itself as best performing on benchmark language processing tests [13, 15, 24]. As well as being able to perform tasks such as text summarising and question answering, LLMs have the potential to support creative writing [6, 8, 9]. Current state-of-the-art LLMs are summarized in table 1. However, despite impressive technical achievements, LLMs have limitations including: (a) models, as they scale, might eventually run into the limits of any pretraining objectives; (b) the models are expensive and difficult to perform inference on; (c) model decisions are not easily interpretable; (d) the majority of the research community, and by extension disadvantaged social groups, have been excluded from the development of LLMs as they are proprietary (see table 1) and, (e) most LLMs are primarily trained on English-language text that contains data biases [13].

1.4. Poetry Specific Language Generation

Creating poetry is creative skill that requires extensive vocabulary, phonemic awareness to produce complex rhyme patterns, and general knowledge of enough subjects about the world to be able to tell interesting stories about a range of topics [20, 25, 26, 27].

Table 2
An Overview of Selected Poetry Writing Tools by Type

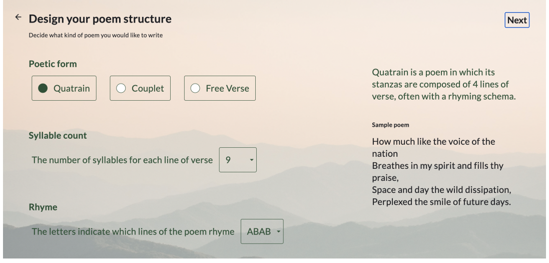
Type	Example	Key Features	Constraints
Autonomous	ChatGPT	Natural language input Generates poems and lyrics	Plain text output Customisation by text prompt
Autonomous	co:here	Natural language input Generates poems and lyrics	Plain text output High latency
Autonomous	Rytr	UI has song lyric option Extensive text processing	Uses GPT-3 models Not trained on song data
Support	RhymeZone	Rhyming dictionary/thesaurus Generates rhyme suggestions	Single word only Cannot be used to write text
Support	Rhymer	Rhyming dictionary Generates range of word types	Single word only Cannot be used to write text
Support	Poetry Foundation	Poetry archives and tutorials Guides user to external resources	No support for real-time creation No user customisation options
Co-creativity	Poem Generator	Customise inputs to create poem Variety of formal poetic outputs	Input variables fixed Limited user interaction or feedback
Co-creativity	DeepBeat	Generates and/or suggests lyrics Displays sources of lyric inspiration	Confusing user interface Unoriginal output vs GPT-3 models
Co-creativity	Verse by Verse	Suggests stanzas in style of known poets Language model accounts for bias	Limited forms of poetry Trained on selected U.S poets

1.5. Overview of Poetry Support Tools

Historically, poetry creation systems tended to built on the model of the an AI writing a full poem by itself, thus writing in a closed system [28, 29, 30, 31]. Early systems tended to be rule-based [32]. More recently, some approaches have started to explore human interaction when composing poems[33, 34]. Table 2 provides a broad summary of selected systems including autonomous, support tools and co-creative as defined by Karimi et al [16]. The category distinction helps frame a range of (human) creative processes and (technology) interactions. It is also a useful way to consider ways in which the proposed system is different to those that currently exist; and as importantly, ways in which it is similar. At a high level, the autonomous systems are designed to be able to create finished works (sometimes called 'products' or 'artefacts'). The support tools are used as part of the creative workflow. For instance, RhymZone or Rhymer help a user find words that sound similar to those they might use in a poem [35, 36]. Co-creative systems facilitate humans and computational systems to make shared products. That said, the distinction is not fixed. For example, *Rytr*, contains text editing, display and other features that allow it to operate as both a co-creative and autonomous system [37]. Having looked at the computational systems, it is instructive to briefly consider poetry writing from a human perspective. It will help inform the design of a new poetry writing system.

Writing poetry requires a range of general creative skills that can be framed in terms of *divergent* and *convergent* thinking; these are used in varying ways throughout a multi-stage writing process. For simplicity, the stages include (a) exploration which is characterised by divergent thinking [21, 38, 39, 40]; (b) focused work is uses convergent thinking [21, 41] and, (c) re-drafting. It is useful in the stages to distinguish between *internal* and *external* co-creation system activities. Internal is when the user interacts with the system in real-time, e.g writing or redrafting text; external is when the user participates in activities such as browsing, reading or other things that do not not use the system. The framing of internal and external system activities is based on the reasoning that; (a) *skill*: inexperienced users are unlikely to possess the improvisational skill required to create full poems in real-time due to cognitive processing constraints [20, 42]; (b) *speed*: users might choose to write poems over multiple sessions, in this case external system stimuli could have supported the writing; (c) *knowledge*: advanced writers are usually familiar with a body of existing that informs their work [1] and, (d) *process*: reflecting and redrafting is an important part of writing . The reflecting stage often takes place separately to the creation of the work itself [1, 10].

1.6. Case Study: Verse by Verse



Design your poem structure

Decide what kind of poem you would like to write

Poetic form

● Quatrain ○ Couplet ○ Free Verse

Syllable count

The number of syllables for each line of verse: 9

Rhyme

The letters indicate which lines of the poem rhyme: ABAB

Quatrain is a poem in which its stanzas are composed of 4 lines of verse, often with a rhyming schema.

Sample poem

How much like the voice of the nation
Breathes in my spirit and fills thy praise,
Space and day the wild dissipation,
Perplexed the smile of future days.

Figure 1: Google's Verse by Verse: users select from a range of US poets and custom design a poem by choosing from features including the number of syllables per line and the number of stanzas.

Screenshot from Verse by Verse application by Google

Google Research *Verse by Verse* is relevant case study as it is arguably the most technically advanced poetry-specific generative system. As well as using transformer model architecture, it also uses informational retrieval, and considers bias within its design. Verse by Verse augments user poetry composition by offering suggestions to a user as they compose a poem. The authors of the system argue that relative to a creating full poems, "this is a much more challenging task, as one needs be able to offer suggestions with minimal latency while meeting constraints of the poem structure and handle the challenges of user input[34]. Figure 1 shows part of the system's user interface (for PC). From a user's point of view, the experience is as follows (a) the user selects poet(s) to inform the suggestions; (b) the user designs poem structure as illustrated in figure 2; (c) the user writes the first line of text and, (d) the system offers suggestions in the style of the poet(s) the user selected earlier. The user can then work with, modify or have the system create new verses. The Verse by Verse design has an external system context that, in general , LLMs do not. To some extent, the system helps poetry writers become better readers. In his work on creativity it was suggested to Csikszentmihalyi that "the only way you become a poet...is because you've read a poem...poetry depends on the whole poetic tradition of the past...you have to decide...out of all that previous poetry, what is most interesting to me?" [1] Verse by Verse, by making users aware of the work of other poets, helps users become readers *in order to inform* their own poetic development.

2. Experiment Design

Verse by Verse ran comparative evaluations of the system against poems written by classic poets. Although the system was intended to be used as an interactive co-creator for the human writing a poem, the author's stated it was still worth evaluating how the system could perform on its own in writing a poem given a first line of verse [34]. This approach has been adopted within the proposed system experimental design, implementation and evaluation. The next subsection explores evaluation prior to looking at implementation. The rationale is that the evaluation is perhaps a harder problem as it involves an intersection of multiple disciplines (e.g. computational sciences, arts, linguistics, and pedagogy). Implementation can mostly be restricted to computational science domains.

2.1. Evaluation Overview

Evaluating co-creative systems is still an open research question and there is no standard metric for measuring computational co-creativity [16, 43]. Karimi et al describe the limited research investigating how co-creative systems can be evaluated. They present four questions as a way to compare how (existing) co-creative systems evaluate creativity: who is evaluating the creativity, what is being evaluated, when does evaluation occur ,and how the evaluation is performed [16]. Calderwood et al point out that "writers engaged with co-creative systems are looking for creative insight, something not measured by perplexity or by a language model's ability to solve the canonical downstream NLP tasks [5]. For the evaluation of the system proposed to be effective it is insightful to restate its goals in more detail. The co-creative poetry system's goal is "making people feel that they have "creative superpowers"? To achieve this, the system supports users to create better poetry than they might otherwise have done without the system. The terms *supports* and *better* will be further explored as they form the basis of evaluation.

Augmenting human users is central to HCAI and a contrast to a closed model that creates on behalf of the user [8, 34, 44]. This point is made in recent work that refers to pitfalls when designing human-AI co-creative systems, as well as other work which asserts that generative models can help writers without writing for them [5, 9, 22]. The arguments these, and similar work, make is that too much automated creation can be at the expense of human users [9, 22]. Adopting this thinking, it is useful to evaluate the system and its users independently, as well as in combination. This in theory allows (*system*) internal and (*human*) internal and external measurement.

The end goal here is that human users develops their capacity; this could be external to the system, whereby the system as acted as a creative prompt. A description of how this could work in principle follows. A later section describes system implementation.

2.2. Process and Objectives

The system would run a number of experiments with the purpose of establishing which system components most support users to write “better” poetry; in goal terms, better is evaluated (a) subjectively by users via a Likert scale [45] and (b) by performance on related tasks such as the Divergent Action Task, Bridge-the-Associative-Gap Task, or rhyme creation and identification [46, 47] The tasks would be completed external to the system. The goals of the evaluation are to measure to what extent *users are actually improving* their poetry writing abilities, and the degree to which any improvement is as a result of internal system features. For a user, improvement is concerned with "the writer’s goals or their desire to have an individual voice" [9]. With this as a basis, the evaluation process takes the form of a number of hypotheses and related experiments, the purpose of which is to explore; (a) how well general vs poetry specific language models can write full poems; (b) if poetry specific language models can better represent individual users *style* than generalised language models; and, (c) the extent to which users benefit when writing poems from system recommendations. The hypotheses and experiments are concerned with *poetic* text style which describes the ways (an author) uses language, including prosody, word choice, sentence structure and use of figurative language [48, 49].

A central challenge for the proposed system is that the development and attainment of an individual poetic voice is highly subjective. Beyond subjectivity, poetry is from a societal perspective often a question of cultural *value* which over time may well change. In reference to Kendrick Lamar’s 2018 Pulitzer Prize, a first for a rap album, their administrator of prizes said, "..this is not a genre we’ve seen celebrated before, so that in that sense it’s historical." [50] Furthermore, as Boden states, "...even in science, values are often elusive and sometimes changeable...because values are highly variable, it follows that many arguments about creativity are rooted in disagreements about value. This applies to human activities no less than to computer performance." [2]

1. *Hypothesis-A* that poetry specific language generation could outperform general language generation with respect to creating poems. *Experiment A*: each system-state generates complete poetic texts. The prompts would also be given to users (inexperienced and advanced) with the same constraints as the system in terms of keywords, topics, character limits etc. The evaluation for experiment A is by humans who judge the quality of the poems (which are anonymous) by a Likert scale and free text summary.

2. *Hypothesis-B* that poetry specific language generation customised for a given user could outperform vanilla poetry specific generation with respect to creating poems. *Experiment B*: each system state generates complete poetic text but some states are pretrained to customise characteristics with respect to given users and their poetic styles. The evaluation for experiment B is by humans who judge the quality of the poems by a Likert scale and free text summary. The evaluation is focused on how well the poems represent the given users’ individual style.

3. *Hypothesis-C* that external recommendations, full or part poems, based on given user characteristics are supportive with respect to users writing their poems. *Experiment C*: for given users generated poetic text inputs, the system state generates (external to system) poetic text recommendations that the user reads and reflects on before completing their poem. The evaluation for experiment C is by humans who judge how well the poem recommendations helped them write poems in the theme, topic or style they were attempting to achieve.

The approach described provides a sense of how user activities (internal and external) with respect to the system can be evaluated. In practice, more fine-grained evaluation criteria would be required based on further research and operational or implementation design; as far as possible, a complete system would have an awareness of all relevant evaluation data including for instance, external system reading of poems. At this stage, the evaluation proposed is limited to the extent necessary in order to support the explanation of how and why the system might work. A later section (*Limitations and Future Work*) will the explore the limitations as suggest possible remedies.

3. Proposed Implementation

The system would have a number of *states* that range from full automation to text prompts acting as a starting point for the user. The support states envisaged are:

1. *State-A:* general language system implemented as standard.

2. *State-B:* general language system implemented with modified architecture to include user generated content within training set and/or network architecture preferences.
3. *State-C:* poetry specific system implemented with standard architecture.
4. *State-D:* poetry specific system implemented with modified architecture to include user generated content within training set and/or network architecture preferences.

The LLM component of the system would use publicly available APIs an, where possible, modify network architecture directly where possible [51, 52, 53]. In most cases (table 1) LLM are closed black box systems as illustrated in (figure 3). In part for this reason, ideally a custom poetry and lyric language model would be implemented; aside from practicalities (which will be discussed) there is a a technical challenge in that a poetry and lyric LM would be far smaller than a general LLM. Given the research on LLM size and performance, a custom poetry and lyric LM would in theory therefore under perform against state-of- the-art LLMs [18, 54, 15]. In line with a recent study, which experimented with user experiences of language models, the system could be implemented with a combination of JavaScript, React, Python and Flask [8]. The system would then be deployed as a web application for mobile phones. Mobile is preferred to PC on the basis of its greater reach as a device for both reading and creating contemporary poetry [55, 56].

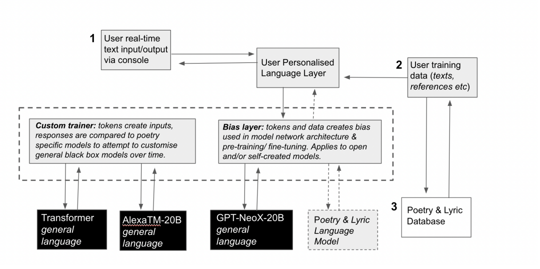

1
User real-time text input/output via console
User Personalised Language Layer
2
User training data (*texts, references etc*)
Custom trainer: tokens create inputs, responses are compared to poetry specific models to attempt to customise general black box models over time.
Bias layer: tokens and data creates bias used in model network architecture & pre-training/ fine-tuning. Applies to open and/or self-created models.
Transformer *general language*
AlexaTM-20B *general language*
GPT-NeoX-20B *general language*
Poetry & Lyric Language Model
3
Poetry & Lyric Database

Figure 2: SParse And Dense Network Model Elements

1. Text input by user is returned as partially completed poetic text and/or poetic and lyrical recommendations for the user to consider. 2. User personalised data submitted as poems or lyrics and/or recommendations of favourite artists and their work. These are used to create a corpus of user text. Prior examples of user generated text uploaded to system; recommender and/or database search to enhance user text with additional poetic texts (e.g from web crawl) 3. Database of poetic texts (and song lyrics) from web crawl. Clean text is included as well as metadata such as rhyme scheme and Parts of Speech (PoS).

3.1. Sparse And Dense Network Model

The system (figure 2) operates as a Sparse And Dense Network (SPAD). The name refers to the system being sparse with respect to user input tokens as compared to tokens contained in the LM/LLMs. Against this, the system is dense in terms of leveraging transformer models and their associated attention layers (table 1). The intuition is to use a small amount of personalised user text to attempt to customise the output of powerful LMs/LLMs. This differs from existing approaches in the following ways.

- State-of-the-Art LLMs form part of the SPAD in order to help improve the SPADs performance; in other words, the LLMs are source of input training data and as such multiple LLMs could in theory be included in the SPAD architectural design.
- A poetry specific LLM (GPT-NeoX) forms part of the design; *poetry specific* refers to adaptations to the underlying model architecture in order that token processing and output is more optimal with respect to poetry than prose. An example of this might be applying additional linguistic layers within the network to favour text strings with syllable frequencies found more regularly in poems than say news articles or web pages. Although *architecture* is referred to, much of any benefit at this stage might come from modifying the training data and associated recipes. The poetry specific LLM would also leverage data from the general LLM (for simplicity any interaction between the two elements is not included in figure 2).
- Poetry and lyric LM is a custom model whose network architecture and training data is specific to poetry. In practical terms it is not a LLM as the available training data is not likely to be sufficiently extensive vs the current state of the art. As well as providing a data contrast to the LLMs, this part of the network will also act as a style transfer layer in so far as it identifies and tries to modify input text to create poetic styles. These styles will be mapped onto user styles upstream within the system.

The result of the models described above, is a system that contains information on generalized poetic style as well as individual style preference(s) unique for each user. This allows the system to support users with specific co-writing tasks (e.g text generation) as well as offer personalised recommendations further reading of relevant poems and/or poets. In user experience terms, this might be delivered via an interface that allows the user to switch between (a) writing text; (b) editing generated text; and (c) reading and reflecting on specific poetic recommendations made by the system.

At this stage, the proposed mode is high-level. There are open questions relating to issues such as real world implementation, customisation of user text, acquisition of training data and other areas. The penultimate section will revisit some of the open design questions and attempt to provide answers. The next section explores the soical significance of poetry and how the a system design could use this to enhance cultural inclusiveness.

4. Discussion

An important goal for poetry is for each writer to discover or develop their own unique style, or artistic voice. Part of a writers development will a result of what poetry they have previously been exposed to. Robert Graves stated that, "only a poet of experience...can hope to put himself in the shoes of his predecessors, or contemporaries, and judge their poems by recreating technical or emotional dilemmas which they faced while at work on them." [57] It can be argued that this statement is, in contemporary terms, biased in gender terms given the assumption of 'poet' being male. Graves's central argument about experience however is echoed in recent studies on language models. A study by Cheng and Uthus made the point that "as creative works are often shaped by the lived experiences and timely issues of the creator's life, a poetry composition system trained on poems from different authors of different eras may reflect a variety of societal biases." [58] Within computer science, social bias is a subject gathering more research attention [17, 59, 60] However, as well as attempting to mitigate negative impacts for disadvantaged groups, considering bias also offers possibility of designing systems that leverage cultural, poetic and linguistic resources that would otherwise be missed. This can benefit all user groups. The next section provides a more concrete example.

4.1. Bias in Language Models

It has been recognised and accepted in recent years that LLM used for text generation contain bias [17, 60] A study by Uthus suggests that "biases in creative language applications are under explored"; it goes on to say it is important to examine biases in these applications because they intended for contexts such as self-expression, collective social enjoyment, and education [58]. One of the key sources of bias in LLM is in the training data sets. LLM retains the biases of the data they have been trained on [15]. Typically the model's pick up on, or reflect, biases and overtly abusive language patterns in training data. This can lead to harms for some users such as encountering derogatory language or discriminatory language (e.g. racist, sexist or ableist) [17]. Studies have how that harms can also exist because of (a) exclusionary social norms in language within language. For example, 'family' is often defined as a basic social unit consisting of a married woman, man and their children; language models internalizing such social norms could be highly discriminatory towards people outside this definition [60]; (b) greater propensity to label of language of marginalized or underrepresented groups as toxic in hate speech detection (e.g. the 'angry black woman' stereotype) [60]; and (c) over representation of certain groups such as white males 18-34 within widely used training data (e.g Reddit posts) [17]. Bender et al assert that, "in the case of US and UK English...white supremacist and misogynistic, ageist, etc. views are over represented in the training data, not only exceeding their prevalence in the general population." [17]. The authors go on to say that the data underpinning LMs stands to "misrepresent social movements and disproportionately align with existing regimes of power."

There are a number of studies that explore bias mitigation through computational techniques such as (a) augmentation of the training data using style transfer [58] or (b) using counterfactuals to reduce sentiment bias [59]. However, in their study describing GPT-3 the authors caution against on over reliance on computational solutions. They instead ask for "...more research that engages with the literature outside NLP, better articulates normative statements about harm, and engages with the lived experience of communities affected by NLP systems...mitigation work should not be approached purely with a metric driven objective to 'remove' bias...but in a holistic manner [15]. For the use case of a poetry co-creation system, bias could be potentially mitigated by including rap lyrics as a key part of the training data set.

4.2. Towards Culturally Responsive Models

Emerging from a hobby of African American youth in the 1970s, rap (as an element of hip-hop) has quickly evolved into a mainstream culture and is the most popular music genre in the U.S and many other territories [61, 62, 63, 64]. Writing rap lyrics requires both creativity to construct a meaningful, interesting story and lyrical skills to produce complex rhyme patterns [26, 48, 65]; within the culture of rap, writers are evaluated by peers on the basis of their wordplay, linguistic complexity and ability to use multiple rhyme types (perfect and imperfect) as well as multi-syllabic rhymes [26, 66]. In many ways, the writer within the hip-hop tradition sets language puzzles for their audience. In a recent BBC documentary, Chuck D, the founder of Public Enemy remarked that "poets were always...going to give you everything the truth...that's very important not only in the realm of hip hop...but in the realm of artistry." [67] Recent computational studies have explored rap on account it its complexity and cultural significance [65, 68, 69]. Rap has historically been excluded from most mainstream discussions on co-creative systems and poetry writing. There may well be valid reasons for this such as language appropriateness, perception around negative sentiments, offensive content, and difficulties in accessing material under copyright. However, although there are challenges, the benefits of using extensive rap lyrics within LM data sets include:

- Training data that represents wider audience concerns, thoughts and feelings.
- Training data will be dynamic and reflect contemporary sociopolitical issues.
- Opens up the possibility of bring voices from excluded communities into the NLP community.
- LMs would be enhanced by a linguistically rich and varied source of data..
- Allows lyrics to be part of a wider conversation which potentially generates. new research insight (for computational, language and social researchers).

Ultimately, as contemporary music's biggest genre, and the one most concerned with rhyme and wordplay, there are multiple reasons to explore using rap lyrics as training data.

5. Limitations and Future Work

The paper has a number of limitations. Below some of these are described along with suggested directions for future work. *System Design and Implementation*: the paper does not fully explore how the proposed system could be built. In particular, there are challenges around the following:

- Building custom LLMs. One of the design limitations is how to effectively experiment with models of varying degrees of openness (for convenience referred to as *black*, *grey* and *white* box). For black box models (e.g GPT-3) there is no way at present to modify the architecture. What instead might be possible is to fine-tune the model via custom queries over a period of time. So, what combinations of prompts generate the most favourable outputs. Grey box models (BLOOM or GPT-NeoX) offer the possibility of powerful models with open-source training and evaluation code plus model weights [53]. However, the costs of running and/or adapting these models could be substantial and not something the paper has explored.
- Customizing models for individuals: this is a system objective but has not been tested. Technically, there is a conflict between the scale and performance benefits of LLM/LM and the comparatively small datasets of individual users. However, as Vigliensoni et al argue, working with small-scale datasets is an overlooked but powerful mechanism for enabling greater human influence over generative AI systems within in creative contexts [70]. The authors describe an experimental project, *ReRites* by Johnston which involved fine-tuning GPT-2 on the artists' custom poetry corpus to generate poems. An approach such as this could be taken although clearly using models such as GPT-2 (for which source code is available) has the limitation of performance vs current state-of-the art LLMs. The personalizing of LLMs to individual users is an open topic that requires further research.

- Acquiring training data: training data for poetry and rap lyrics would not be readily available in the way that the Pile or equivalents are used for general LLMs [19]. The solution to this would be to source data from scraping the web for lyrics, or directly from services such as MusixMatch [71] Poetry training data, much of which will be out of copyright, can be acquired via sites such as Project Gutenberg and Poetry Foundation. This approach to training data was used in a 2019 experiment to create a poetry-specific LLM based on the GPT-2 model [72].

Evaluation: literature on evaluating the creativity in a co-creative systems considers a wide number of factors such who evaluates the creativity (e.g. system itself or human users), what is being evaluated (e.g. user interaction or output), when does evaluation occur (e.g in real time or at the end of a session) and how the evaluation is performed (e.g. methods and related metrics) [16]. There is a broad set of metrics for developing computational models for evaluating creativity. With respect to the system described, the most relevant include a proposed computational model by Agres et al. The model reflects human conceptualization of musical and poetic creativity [73]. Future work could explore the kind of model described alongside other linguistic-based metrics such as the *Divergent Action Task*, *Bridge-the-Associative-Gap Task*, or rhyme creation and identification tasks. [46, 47] Additionally, building on machine learning practices, metrics could be derived for accuracy in terms of the degree to which generated output matches a reference dataset. For example, if the user has a target poetic style, it might be possible to computationally determine the extent to which the completed poem was accurate or not. The paper has not explored these kinds of evaluation in detail and they would form part of future work. Finally, though the evaluations proposed are limited, they could nevertheless contribute to the wider discussion around the topic. As Karimi et al assert "evaluating co-creative systems is still an open research question and there is no standard metric that can be used across specific systems." [16].

6. Conclusion

Artistic creativity is a process, in which an initial improvisational phase is followed by a period of focused re-evaluation and revision [20]. Spontaneous improvisation is a complex cognitive process that shares features with what has been characterized as a 'flow' state [1, 20]. Much current work on co-creative settings focuses on the role of the system as a generator that augments what people can achieve in creative tasks [9]. There are problems with this such aligning the system capabilities and user expectations, language model bias, system interpretability, and user interaction design [8, 22, 74]. Studies have found that different mental expectation of users affects their strategies and perception of the system role in the co-writing process [9, 74].

This position paper explored the recent background to co-creative writing systems, with poetry as a use case. Poetry was defined as including song lyrics for which the paper argued that rap was the most relevant genre. The paper then proposed a system that, as far as the author is aware, has novel features relative to the state of the art. The system and how it could be evaluated and implemented were then described. Importantly, the design includes recommendations for user activities external to the system. The rationale for this is that the system priority is to help the human user to develop an artistic style rather than to create text on the users behalf. Issues around the mitigating some system bias using rap lyrics was also discussed. Future work could include more detailed analysis of evaluation methods as well as how these could be delivered internally to the system. Further work on user interface design is also a topic to develop. Additionally, the implementation proposal is high level and constraints such as latency, database design, and other factors have not been considered. In order to build a viable prototype, software architecture would most likely form the next stage of the research. Finally, to revisit the title of the paper: why build a co-creative poetry system that makes people feel that they have "creative superpowers"? Studies demonstrate that poetry is an emotional capable of engaging the brain's areas of primary reward [75]. It is a form of communication that has existed throughout human and across cultures. In modern society, poetry has become a central part of the most popular music genre. Poetry matters to society. By extension, it is worth building system that can help people experience it firsthand and connect with its traditions. The aim though should not be to make people *feel* they have "creative superpowers"; instead a system should *support people to actually build* "creative superpowers".

7. Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council. The author would also like to acknowledge the support of Swansea Council.

References

[1] M. Csikszentmihalyi, Creativity : the psychology of discovery and invention, Harper Perennial Modern Classics, 2013.

[2] M. Boden, Creativity in a nutshell, Think 5 (2009) 83–96. doi:10.1017/S147717560000230X.

[3] J. P. Guilford, The nature of human intelligence. (1967).

[4] M. A. Boden, The creative mind : myths and mechanisms, Routledge, 2005.

[5] A. Calderwood, V. Qiu, K. Gero, L. B. Chilton, How novelists use generative language models: An exploratory user study, in: HAI-GEN+user2agent@IUI, 2020.

[6] M. Henderson, R. Al-Rfou, B. Strope, Y.-h. Sung, L. Lukacs, R. Guo, S. Kumar, B. Miklos, R. Kurzweil, Efficient natural language response suggestion for smart reply, arXiv.org (2017). URL: https://arxiv.org/abs/1705.00652. doi:10.48550/arXiv.1705.00652.

[7] H. Gonçalo Oliveira, T. Mendes, A. Boavida, Co-poetryme: a co-creative interface for the composition of poetry, Proceedings of the 10th International Conference on Natural Language Generation (2017). URL: https://aclanthology.org/W17-3508/. doi:10.18653/v1/w17-3508.

[8] F. Lehmann, N. Markert, H. Dang, D. Buschek, Suggestion lists vs. continuous generation: Interaction design for writing with generative models on mobile devices affect text length, wording and perceived authorship, in: Proceedings of Mensch Und Computer 2022, MuC '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 192–208. URL: https://doi.org/10.1145/3543758.3543947. doi:10.1145/3543758.3543947.

[9] K. Arnold, A. Volzer, N. Madrid, Generative models can help writers without writing for them, in: Joint Proceedings of the ACM IUI 2021 Work-

shops, 2021. URL: https://ceur-ws.org/Vol-2903/IUI21WS-HAIGEN-1.pdf.

[10] A. Ploin, R. Eynon, I. Hjorth, M. A. Osborne, Ai and the arts: How machine learning is changing artistic work. report from the creative algorithmic intelligence research project, 2022. URL: https://www.oii.ox.ac.uk/news-events/reports/ai-the-arts/.

[11] B. Shneiderman, Design lessons from ai's two grand goals: Human emulation and useful applications, IEEE Transactions on Technology and Society 1 (2020) 73–82. doi:10.1109/tts.2020.2992669.

[12] B. Shneiderman, Creativity support tools, Communications of the ACM 45 (2002) 116–120.

[13] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, 2019. URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[14] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, M. Pieler, U. S. Prashanth, S. Purohit, L. Reynolds, J. Tow, B. Wang, S. Weinbach, Gpt-neox-20b: An open-source autoregressive language model, arXiv.org (2022). URL: https://arxiv.org/abs/2204.06745. doi:10.48550/arXiv.2204.06745.

[15] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020. URL: https://arxiv.org/abs/2005.14165.

[16] P. Karimi, K. Grace, M. L. Maher, N. Davis, Evaluating creativity in computational co-creative systems, CoRR abs/1807.09886 (2018). URL: http://arxiv.org/abs/1807.09886. arXiv:1807.09886.

[17] E. M. Bender, T. Gebru, A. McMillan-Major, S. Shmitchell, On the dangers of stochastic parrots: Can language models be too big? , in: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 610–623. URL: https://doi.org/10.1145/3442188.3445922. doi:10.1145/3442188.3445922.

[18] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei, Scaling laws for neural language models, CoRR abs/2001.08361 (2020). URL: https://arxiv.org/abs/2001.08361. arXiv:2001.08361.

[19] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, C. Leahy, The pile: An 800gb dataset of diverse text for language modeling, arXiv.org (2020). URL: https://arxiv.org/abs/2101.00027. doi:10.48550/arXiv.2101.00027.

[20] S. Liu, H. M. Chow, Y. Xu, M. G. Erkkinen, K. E. Swett, M. W. Eagle, D. A. Rizik-Baer, A. R. Braun, Neural correlates of lyrical improvisation: An fmri study of freestyle rap, Scientific Reports 2 (2012). URL: https://www.nature.com/articles/srep00834. doi:10.1038/srep00834.

[21] W. Zhang, Z. Sjoerds, B. Hommel, Metacontrol of human creativity: The neurocognitive mechanisms of convergent and divergent thinking, NeuroImage 210 (2020).

[22] D. Buschek, L. Mecke, F. Lehmann, H. Dang, Nine potential pitfalls when designing human-ai co-creative systems, 2021. URL: https://arxiv.org/abs/2104.00358. doi:10.48550/ARXIV.2104.00358.

[23] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, J. Tow, A. M. Rush, S. Biderman, A. Webson, P. S. Ammanamanchi, T. Wang, B. Sagot, N. Muennighoff, d. Moral, O. Ruwase, R. Bawden, S. Bekman, A. McMillan-Major, I. Beltagy, H. Nguyen, L. Saulnier, S. Tan, P. O. Suarez, V. Sanh, H. Laurençon, Y. Jernite, J. Launay, M. Mitchell, C. Raffel, A. Gokaslan, A. Simhi, A. Soroa, A. F. Aji, A. Alfassy, A. A. Rogers, A. K. Nitzav, C. Xu, C. Mou, C. Emezue, C. Klamm, C. Leong, v. Strien, D. I. Adelani, D. Radev, E. G. Ponferrada, E. Levkovizh, E. Kim, E. B. Natan, D. Toni, G. Dupont, G. Kruszewski, G. Pistilli, H. Elsahar, H. Benyamina, H. Tran, I. Yu, I. Abdulmumin, I. Johnson, I. Gonzalez-Dios, R. Javier, J. Chim, J. Dodge, J. Zhu, J. Chang, J. Frohberg, J. Tobing, J. Bhattacharjee, K. Almubarak, K. Chen, K. Lo, V. Werra, L. Weber, L. Phan, L. B. allal, L. Tanguy, M. Dey, M. R. Muñoz, M. Masoud, M. Grandury, M. Šaško, M. Huang, M. Coavoux, M. Singh, M. T.-J. Jiang, M. C. Vu, M. A. Jauhar, M. Ghaleb, N. Subramani, N. Kassner, N. Khamis, O. Nguyen, O. Espejel, d. Gibert, P. Villegas, P. Henderson, P. Colombo, P. Amuok, Q. Lhoest, R. Harliman, R. Bommasani, R. L. López, R. Ribeiro, S. Osei, S. Pyysalo, S. Nagel, S. Bose, S. H. Muhammad, S. Sharma, S. Longpre, S. Nikpoor, S. Silberberg, S. Pai, S. Zink, T. T. Torrent, T. Schick, T. Thrush, V. Danchev, V. Nikoulina, V. Laippala, V. Lepercq, V. Prabhu, Z. Alyafeai, Z. Talat, A. Raja, B. Heinzerling, C. Si, E. Salesky, S. J. Mielke, W. Y. Lee, A. Sharma, A. Santilli, A. Chaffin, A. Stiegler, D. Datta, E. Szczechla, G. Chhablani, H. Wang, H. Pandey, H. Strobelt, J. A. Fries, J. Rozen, L. Gao, L. Sutawika, B. M. Saiful, M. S. Al-shaibani, M. Manica, N. Nayak, R. Teehan, S. Albanie, S. Shen, S. Ben-David, S. H. Bach, T. Kim, T. Bers, T. Fevry, T. Neeraj, U. Thakker, V. Raunak, X. Tang, Z.-X. Yong,

Z. Sun, S. Brody, Y. Uri, H. Tojarieh, A. Roberts, H. W. Chung, J. Tae, J. Phang, O. Press, C. Li, D. Narayanan, H. Bourfoune, J. Casper, J. Rasley, M. Ryabinin, M. Mishra, M. Zhang, M. Shoeybi, M. Peyrounette, N. Patry, N. Tazi, O. S Sanseviero, v. Platen, P. Cornette, P. F. Lavallée, R. Lacroix, S. Rajbhandari, S. Gandhi, S. Smith, S. Requena, S. Patil, T. Dettmers, A. Baruwa, A. Singh, A. Cheveleva, A.-L. Ligozat, A. Subramonian, A. Névéol, C. Lovering, D. Garrette, D. Tunuguntla, E. Reiter, E. Taktasheva, E. Voloshina, E. Bogdanov, G. I. Winata, H. Schoelkopf, J.-C. Kalo, J. Novikova, J. Z. Forde, J. Clive, J. Kasai, K. Kawamura, L. Hazan, M. Carpuat, M. Clinciu, N. Kim, N. Cheng, O. Serikov, O. Antverg, v. , R. Zhang, R. Zhang, S. Gehrmann, S. Pais, T. Shavrina, T. Scialom, T. Yun, T. Limisiewicz, V. Rieser, V. Protasov, V. Mikhailov, Y. Pruksachatkun, Y. Belinkov, Z. Bamberger, Z. Kasner, A. Rueda, A. Pestana, A. Feizpour, A. Khan, A. Faranak, A. Santos, A. Hevia, A. Unldreaj, A. Aghagol, A. Abdollahi, A. Tammour, A. HajiHosseini, B. Behroozi, B. Ajibade, B. Saxena, C. M. Ferrandis, D. Contractor, D. Lansky, D. David, D. Kiela, D. A. Nguyen, E. Tan, E. Baylor, E. Ozoani, F. Mirza, F. Ononiwu, H. Rezanejad, H. Jones, I. Bhattacharya, I. Solaiman, I. Sedenko, I. Nejadgholi, J. Passmore, J. Seltzer, J. B. Sanz, K. Fort, L. Dutra, M. Samagaio, M. Elbadri, M. Mieskes, M. Gerchick, M. Akinlolu, M. McKenna, M. Qiu, M. Ghauri, M. Burynok, N. Abrar, N. Rajani, N. Elkott, N. Fahmy, O. Samuel, R. An, R. Kromann, R. Hao, S. Alizadeh, S. Shubber, S. Wang, S. Roy, S. Viguier, T. Le, T. Oyebade, T. Le, Y. Yang, Z. Nguyen, A. R. Kashyap, A. Palasciano, A. Callahan, A. Shukla, A. Miranda-Escalada, A. Singh, B. Beilharz, B. Wang, C. Brito, C. Zhou, C. Jain, C. Xu, C. Fourrier, D. L. Periñán, D. Molano, D. Yu, E. Manjavacas, F. Barth, F. Fuhrimann, G. Altay, G. Bayrak, G. Burns, H. U. Vrabec, I. Bello, I. Dash, J. Kang, J. Giorgi, J. Golde, J. D. Posada, K. R. Sivaraman, L. Bulchandani, L. Liu, L. Shinzato, M. Hahn, M. Takeuchi, M. Pàmies, M. A. Castillo, M. Nezhurina, M. Sänger, M. Samwald, M. Cullan, M. Weinberg, D. Wolf, M. Mihaljcic, M. Liu, M. Freidank, M. Kang, N. Seelam, N. Dahlberg, N. M. Broad, N. Muellner, P. Fung, P. Haller, R. Chandrasekhar, R. Eisenberg, R. Martin, R. Canalli, R. Su, R. Su, S. Cahyawijaya, S. Garda, S. S. Deshmukh, S. Mishra, S. Kiblawi, S. Ott, S. Sang-aroonsiri, S. Kumar, S. Schweter, S. Bharati, T. Laud, T. Gigant, T. Kainuma, W. Kusa, Y. Labrak, Y. S. Bajaj, Y. Venkatraman, Y. Xu, Y. Xu, Y. Xu, Z. Tan, Z. Xie, Z. Ye, M. Bras, Y. Belkada, T. Wolf, Bloom: A 176b-parameter open-access multilingual language model, arXiv.org (2022). URL: https://arxiv.org/abs/2211.05100. doi:10.48550/arXiv.2211.05100.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, arXiv.org (2017). URL: https://arxiv.org/abs/1706.03762. doi:10.48550/arXiv.1706.03762.

[25] N. L. Hadaway, S. M. Vardell, T. A. Young, Scaffolding oral language development through poetry for students learning english, The Reading Teacher 54 (2001) 796–796. URL: https://go.gale.com/ps/i.do?id=GALE%7CA75085276&sid=googleScholar&v=2.1&it=r&linkaccess=abs&issn=00340561&p=AONE&sw=w&userGroupName=anon%7E20f961a3.

[26] A. Bradley, Book of rhymes : the poetics of hip hop, Basic Civitas, 2017.

[27] I. Alonso, L. Davachi, R. Valabrègue, V. Lambrecq, S. Dupont, S. Samson, Neural correlates of binding lyrics and melodies for the encoding of new songs, NeuroImage 127 (2016) 333–345. URL: https://pubmed.ncbi.nlm.nih.gov/26706449/. doi:10.1016/j.neuroimage.2015.12.018.

[28] H. G. Oliveira, A rest service for poetry generation, 2017. URL: https://www.semanticscholar.org/paper/A-REST-Service-for-Poetry-Generation-Oliveira/5b0039186ddb41ad5d037e5dbacfae837eaa5079.

[29] H. G. Oliveira, Poetryme : a versatile platform for poetry generation, 2012. URL: https://www.semanticscholar.org/paper/PoeTryMe-%3A-a-versatile-platform-for-poetry-Oliveira/0c62affa157a453e01514042b55babff428928fa.

[30] X. Zhang, M. Lapata, Chinese poetry generation with recurrent neural networks, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014). URL: https://aclanthology.org/D14-1074/. doi:10.3115/v1/d14-1074.

[31] T. Van de Cruys, Automatic poetry generation from prosaic text, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020). URL: https://aclanthology.org/2020.acl-main.223/. doi:10.18653/v1/2020.acl-main.223.

[32] J. H. Lau, T. Cohn, T. Baldwin, J. Brooke, A. Hammond, Deep-speare: A joint neural model of poetic language, meter and rhyme, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2018). URL: https://aclanthology.org/P18-1181/. doi:10.18653/v1/p18-1181.

[33] Google, Verse by verse, 2022. URL: https://sites.research.google/versebyverse/.

[34] D. Uthus, M. Voitovich, R. Mical, Augmenting poetry composition with verse by verse, 2022. doi:10.18653/v1/2022.naacl-industry.3.

[35] WriteExpress, Rhymer, 2023. URL: https://www.rhymer.com/.

[36] Datamuse, Rhymezone rhyming dictionary and thesaurus, 2023. URL: https://www.rhymezone.com/.

[37] Rytr, Rytr - best ai writer, content generator writing assistant, 2022. URL: https://rytr.me/.

[38] M. A. Runco, Divergent thinking, creativity, and ideation. (2010).

[39] C. Lewis, P. J. Lovatt, Breaking away from set patterns of thinking: Improvisation and divergent thinking, Thinking Skills and Creativity 9 (2013) 46–58.

[40] M. A. Runco, S. Acar, Divergent thinking as an indicator of creative potential, Creativity research journal 24 (2012) 66–75.

[41] A. Cropley, In praise of convergent thinking, Creativity Research Journal - CREATIVITY RES J 18 (2006) 391–404. doi:10.1207/s15326934crj1803_13.

[42] A. T. Landau, C. J. Limb, The neuroscience of improvisation, Music Educators Journal 103 (2017) 27–33. URL: https://doi.org/10.1177/0027432116687373. doi:10.1177/0027432116687373.

[43] Studying the Impact of AI-based Inspiration on Human Ideation in a Co-Creative Design System, 2021. URL: https://ceur-ws.org/Vol-2903/IUI21WS-HAIGEN-7.pdf.

[44] B. Shneiderman, Human-Centered AI, Oxford University Press, 2022.

[45] A. Joshi, S. Kale, S. Chandel, D. Pal, Likert scale: Explored and explained, British Journal of Applied Science Technology 7 (2015) 396–403. URL: https://eclass.aspete.gr/modules/document/file.php/EPPAIK269/5a7cc366dd963113c6923ac4a73c3286ab22.pdf. doi:10.9734/bjast/2015/14975.

[46] J. A. Olson, J. Nahas, D. Chmoulevitch, S. J. Cropper, M. E. Webb, Naming unrelated words predicts creativity, Proceedings of the National Academy of Sciences 118 (2021). URL: https://www.pnas.org/content/118/25/e2022340118. doi:10.1073/pnas.2022340118.

[47] J. Ocumpaugh, M. Mercedes, T. Rodrigo, K. Porayska-Pomsta, I. Olatunji, R. Luckin, Becoming better versed: Towards the design of a popular music-based rhyming game for disadvantaged youths, Proceedings of the 26th International Conference on Computers in Education. Philippines: Asia-Pacific Society for Computers in Education (2018). URL: https://apsce.net/icce/icce2018/wp-content/uploads/2018/12/C6-04.pdf.

[48] H. Hirjee, D. Brown, Using automated rhyme detection to characterize rhyming style in rap music, Empirical Musicology Review 5 (2010) 121–145. doi:10.18061/1811/48548.

[49] Z. Hu, R. K.-W. Lee, C. C. Aggarwal, A. Zhang, Text style transfer: A review and experimental evaluation (2020). URL: https://arxiv.org/abs/2010.12742. doi:10.48550/ARXIV.2010.12742.

[50] R. Roberts, Kendrick lamar's pulitzer prize sparks lively — and at times snobby — conversations on the aesthetics of music, 2018. URL: https://www.latimes.com/entertainment/music/la-et-ms-kendrick-pulitzer-reactions-20180420-story.html.

[51] OpenAI, Openai api, 2021. URL: https://openai.com/api/.

[52] Amazon, Alexatm 20b is now available in amazon sagemaker jumpstart | amazon web services, 2022. URL: https://tinyurl.com/amazonGPT.

[53] HuggingFace, Gpt-neox, 2022. URL: https://huggingface.co/docs/transformers/main/en/model_doc/gpt_neox#overview.

[54] A. Komatsuzaki, Current limitations of language models: What you need is retrieval, 2020. URL: https://www.researchgate.net/publication/344261335_Current_Limitations_of_Language_Models_What_You_Need_is_Retrieval.

[55] F. Hill, K. Yuan, How instagram saved poetry: Social media is turning an art form into an industry, 2018. URL: https://www.theatlantic.com/technology/archive/2018/10/rupi-kaur-instagram-poet-entrepreneur/572746/.

[56] H. Oliver, Instagram is the future of poetry, 2021. URL: https://unherd.com/2021/10/instagram-is-the-future-of-poetry/.

[57] M. Schmidt, Lives of the Poets, Phoenix, 1999.

[58] E. Sheng, D. C. Uthus, Investigating societal biases in a poetry composition system, ACL Anthology (2020) 93–106. URL: https://aclanthology.org/2020.gebnlp-1.9/.

[59] P.-S. Huang, H. Zhang, R. Jiang, R. Stanforth, J. Welbl, J. Rae, V. Maini, D. Yogatama, P. Kohli, Reducing sentiment bias in language models via counterfactual evaluation, 2019. URL: https://arxiv.org/abs/1911.03064. doi:10.48550/ARXIV.1911.03064.

[60] A. K., M. P. Gangan, D. P., L. V. L., Towards an Enhanced Understanding of Bias in Pre-trained Neural Language Models: A Survey with Special Emphasis on Affective Bias, Springer Nature, Singapore, 2022.

[61] J. Lynch, Hip-hop passes rock to become most popular music genre for first time in history: Nielsen, 2018. URL: https://www.businessinsider.com/hip-hop-passes-rock-most-popular-music-genre-nielsen-2018-1?r=US&IR=T.

[62] A. Texas, Hip-hop is the most listened to genre in the world, 2015. URL: https://www.nme.com/news/music/various-artists-1151-1214849.

[63] Wikipedia, Hip hop, 2021. URL: https:

//en.wikipedia.org/wiki/Hip_hop.

[64] T. Ingham, Nearly a third of all streams in the us last year were of hip-hop and rnb artists as rock beat pop to second, 2021. URL: https://www.musicbusinessworldwide.com/nearly-a-third-of-all-streams-in-the-us-last-year-were-of-hip-hop-and-rb-music/.

[65] E. Malmi, P. Takala, H. Toivonen, R. Tapani, A. Gionis, Dopelearning: A computational approach to rap lyrics generation *, KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016). doi:10.1145/2939672.2939679.

[66] J. Eastwood, E. Hinton, We wrote an algorithm to unravel the rhymes of hit musical 'hamilton', 2016. URL: http://graphics.wsj.com/hamilton/.

[67] C. D, Fight the power: How hip hop changed the world, ???? URL: https://www.bbc.co.uk/programmes/p0dj70yd.

[68] N. Condit-Schultz, MCFlow: A Digital Corpus of Rap Flow, Ph.D. thesis, 2016. URL: https://etd.ohiolink.edu/apexprod/rws_etd/send_file/send?accession=osu1461250949&disposition=inline.

[69] J. Eastwood, E. Hinton, How wsj used an algorithm to analyze 'hamilton' the musical, 2016. URL: http://graphics.wsj.com/hamilton-methodology/.

[70] A Small-Data Mindset for Generative AI Creative Work, 2022.

[71] Musixmatch developer api, 2023. URL: https://developer.musixmatch.com/.

[72] S. Presser, Gpt-2 neural network poetry, 2019. URL: https://www.gwern.net/GPT-2.

[73] S. Mcgregor, K. Agres, M. Purver, G. Wiggins, From distributional semantics to conceptual spaces: A novel computational method for concept creation, Journal of Artificial General Intelligence 6 (2015) 55–86. doi:10.1515/jagi-2015-0004.

[74] D. Yang, Y. Zhou, Z. Zhang, T. Jia, J. Li, R. Lc, Ai as an activewriter: Interaction strategies with generated text in human-ai collaborative fiction writing, 2019. URL: https://ceur-ws.org/Vol-3124/paper6.pdf.

[75] E. Wassiliwizky, S. Koelsch, V. Wagner, T. Jacobsen, W. Menninghaus, The emotional power of poetry: neural circuitry, psychophysiology and compositional principles, Social Cognitive and Affective Neuroscience 12 (2017) 1229–1240. doi:10.1093/scan/nsx069.