Text Generation Using Recurrent Neural Networks: A Turing Test for Poetry

Emily Yin
Department of Computer Science
Princeton University

Abstract

This article investigates computational methods of generating poetry, a domain traditionally associated with human creativity. To achieve this objective, I generated poetry using a multi-layer recurrent neural network for character-level language models and conducted a modified Turing Test using a suite of machine-generated and human-authored pieces. While many experiments have been done in this vein, I trained the neural network on two separate corpora—one archaic and one contemporary—to investigate intra-group trends. In all, 86 respondents were surveyed and partitioned into expert and non-expert groups for analysis, with faculty at Princeton University (primarily in the Computer Science and English departments) comprising the first group and university students comprising the second. My large-scale extrinsic evaluation demonstrated that the non-experts frequently misidentified 'archaic' machine poems as human-authored, while experts were generally unfooled. Both groups, however, struggled to correctly identify 'contemporary' machine poems. Moreover, most respondents indicated a preference for the human-authored poems irrespective of their accuracy on the Turing Test.

1. Introduction

The field of artificial intelligence is most fundamentally the simulation of human cognition, encompassing both the theory and development of computer systems with adaptive capabilities [12].

Since its inception, the possibility of sentient machines has long captivated the popular imagination. There is a rich lineage of films, from *The Matrix* and *Blade Runner* to *Ex-Machina* and *Her*, that explore the implications—and meaning—of 'consciousness'. Barring a major theoretical breakthrough, AI technologies are still far from attaining superintelligence. However, they have progressed rapidly over the past decade with improvements in computing power and the advent of big data. Many applications, ranging from optical character recognition to machine translation, have become quite sophisticated; recurrent neural networks, a set of machine learning algorithms notable for their computational intensity, have transformed computer vision through advances in image and video captioning [8]. Perhaps the final frontier, then, is computational creativity. What do computer-generated art, music, and poetry look like? Is it possible for machines to articulate the human experience more compellingly than humans themselves?

2. Related Work

"A taxonomy of generative poetry techniques" surveys text generation methods with varying degrees of algorithmic sophistication. The most naive include template-based generation, Markov chaining, and context-free grammars; these 'mere generation' methods often fail to achieve semantic coherence or exercise knowledge-based reasoning [9]. Indeed, generating convincing poetry involves attention to both content (a poem's semantics) and form (the sonic properties and rules that govern a poem). A Facebook AI research paper [6] proposes two methodologies with these criteria in mind.

The first model is trained at the phonetic level. More specifically, a text sequence is encoded as a phonetic representation, which serves as the training corpus for a neural language model; the generated phonemic output is then converted back to an orthographic representation. This model works well on small training sets. Moreover, conceptualizing language as acoustic symbols enables the model to learn aspects of pronunciation as well as rhyme and rhythm, encapsulating both content and form. However, this advantage is also its largest constraint. Given the diversity of poetic devices in use, samples from a model trained on a general poetry corpus "would be unlikely to maintain internal consistency of meter and rhyme" [6]. To generate sonnets, for example, a phonetic model

would need to be trained on a corpus of sonnets (and only sonnets).

An alternative approach is to decouple the training of content and form, first applying a character-level language model to represent the former and then imposing formal, i.e. rhythmic, constraints with a discriminative weighted finite-state machine. The FSM probabilistically accepts or rejects sampled lines based on how closely it adheres to the desired meter. The content can be further tuned to incorporate themes. By heuristically increasing the sampling probability of words that relate to a particular theme, *heat*, *sun*, and *warmth* would be more likely to appear in a summer-themed poem. Poetic devices, which can be represented as patterns in character sequences, are incorporated in like manner; by tracking historical frequencies of sampled characters (at a location which depends on the desired poetic device) and increasing their probabilities in subsequent time steps, we attain properties such as alliteration and assonance [6].

Char-RNN, created by Tesla's AI Director Andrej Karpathy, is a multi-layer recurrent neural network for character-level language models. The SAS Institute defines neural networks as a computing system that relays and processes data via interconnected nodes [1]. Each node consists of input connections, a function combining the inputs, and an output connection; layers of nodes collectively form a network. A multi-layer network is simply one that has multiple hidden layers—'hidden' because they are not observable from the inputs or outputs [3]. In contrast to unidirectional feed-forward networks, RNNs contain feedback loops that allow information to persist. However, recurrent neural networks still have trouble 'remembering' long sequences [13].

To solve this problem, ***Char-RNN*** employs long short-term memory, a four-layer RNN variant capable of learning long-term dependencies [10]. Character-level language models generate text by repeatedly feeding a character into the RNN and sampling from the probability distribution of characters that are likely to follow it [8]. This approach provides several advantages over word-level models. A more minimal vocabulary translates to greater efficiency in training. Moreover, their ability to learn affixes allows them to assimilate words not contained in the training corpus [6]. Indeed, Karpathy's ***Char-RNN*** has been trained on data sets as diverse as Paul Graham's essays, Shakespearean works, and Latex source files, with impressive results [8].

3. Model and Datasets

In this study, the corpora were trained on ***Torch-RNN***—a more efficient re-implementation of the aforementioned ***Char-RNN*** with the Torch code base—on large corpora of poetry. After conducting minimal experiments using comparatively lightweight Markov chain generators, I found the outputs to be uncreative; that is, they were very similar in terms of style and phraseology to the input texts. In contrast, its predecessor ***Char-RNN*** has a proven track record of generating convincing prose results [8]. ***Torch-RNN*** uses Adam, a method for stochastic optimization, and hard-codes the neural network's forward and backward passes for space/time efficiency.

My initial hypothesis was that the output of models trained on modern poetry would be more difficult to distinguish than those trained on stylized, Romantic-era poetry; avant-garde works are generally regarded as more inaccessible to the common reader than, say, Yeats. To my knowledge, there exists only one large centralized repository of contemporary poetry: a Kaggle dataset scraped from the Poetry Foundation website [5]. I augmented this dataset by scraping poems from a selection of literary journals. Collectively, these poems form a 30 MB corpus that I will henceforth call Poetry Foundation+. The criteria I used for journal selection were repute, i.e. poems in these publications are regularly included in prestigious anthologies such as the *Pushcart* and *Best American Poetry*, as well as stylistic diversity, with magazine aesthetics ranging from conventional to experimental.

To establish a counterpoint to my Poetry Foundation+ outputs, I also trained ***Torch-RNN*** on the 121 MB Gutenberg poetry corpus, which consists of approximately three million lines of poetry extracted from books offered through Project Gutenberg. Its high concentration of formal poetry offered a means to test my hypothesis that samples generated from contemporary, more experimental poetry would be more convincing than that generated from metrical verse.

For the Gutenberg poetry corpus, I downloaded the original JSON file from its GitHub repository, parsed its constituent lines of poetry, and wrote them to a text file. I similarly extracted a 22 MB Poetry Foundation dataset available on Kaggle. To augment the Poetry Foundation dataset, poems were scraped from a selection of online literary publications and concatenated to the Kaggle

dataset.

Data cleaning was performed in two passes. First, masthead information, contributor bios and poems written in foreign languages (as poems translated into English for publication are often preceded by the original) were manually removed. While non-English words can easily be filtered out, quite a few English-language poems contain fragments in other languages; a programmatic solution would have rendered such "bilingual" poems—and every piece containing made-up words—incomplete. To remove accents, and non-Roman character sets such as Arabic and Mandarin—none of which ***Torch-RNN***'s preprocessor script can handle—I cleaned the data programmatically, removing all diacritics using the ***sed*** command and stripping out the remaining non-ASCII characters using ***perl***.

4. Hyperparameters

Artificial neural networks have two hyperparameters that determine the topology of the network: the number of layers (*num_layers*) and the number of nodes in the network (*rnn_size*). The most relevant default hyperparameters in *Torch-RNN* are *num_layers* = 2, *rnn_size* = 128, *model_type* = lstm, *max_epochs* = 50 (for training) and *temperature* = 1 (for sampling).

The full specification can be found in the *Torch-RNN* GitHub repository. While "model hyperparameter configuration is more art than science" [3], greater depth has produced better results, empirically, for a wide range of tasks [4]. As such, using deeper networks—and erring on the side of more rather than fewer layers—offers a heuristic for configuring hyperparameters [3]. Both corpora were trained with three layers; the Poetry Foundation+ corpus was trained with 500 nodes and the Gutenberg corpus with 700.

During the sampling process, the *temperature* hyperparameter was also toggled. This value, which falls in the range (0, 1], divides the predicted probabilities before the application of the Softmax classifier used in the final RNN layer. Lower temperatures lead to more conservative predictions and outputs, while higher temperatures lead to more novel but also mistake-ridden outputs [2].

5. Sampling Procedure

By default, checkpoint files are generated every 1000 iterations during training. The quality of checkpoint samples does not monotonically increase; from empirical observation, sample coherence increases steadily for the bulk of the training period, then degenerates markedly after a certain point—a hallmark of model overfitting. In this stage, I had two aims: to explore the characteristics of sampled outputs across different checkpoints and temperatures and to determine the aforementioned inflection point in validation loss, enabling high-quality samples for evaluation. A protocol was established to systematize this process for every (dataset, rnn_size, num_layers).

I began by sampling the first, middle and last checkpoints (denoted by c_{first}, c_{mid}, and c_{last}, respectively), generating two 1000-character outputs per checkpoint—one with *temp* = 0.7 and one with *temp* = 0.8. Then, I manually performed a binary search procedure to determine the best checkpoint value. If c_{mid} generated the best outputs among the three, I terminated the procedure. Otherwise, if c_{last} was better than c_{first}, I averaged the checkpoint values of c_{mid} and c_{last} and sampled from $c_{0.5 * (mid + last)}$. If these sampled outputs were higher-quality than those from c_{mid} and c_{last}, I terminated the procedure. Otherwise, the process continued until I determined a neighborhood of good checkpoint files; any exact value is necessarily a somewhat arbitrary selection due to the subjectivity of the process. The above temperature values were found to yield the most illuminating results—low temperatures produced overly-conservative (and thus generic) outputs, while temperatures of 0.9 or 1 frequently generated incoherent outputs.

N.B. While I cleaved to the aforementioned procedure while sampling the Poetry Foundation+ corpus, I observed that Gutenberg corpus outputs—save for those sampled from the earliest checkpoints—were, to my eye, rather indistinguishable in terms of quality. As such, I simply sampled from the last checkpoint file. To obtain high-quality output from the selected checkpoint file, ten different 1000-character samples were taken for every temperature from 0.4 to 0.9.

6. Exploratory Data Analysis

6.1. NLTK Analytics

- *Text Characteristics*: *Love*, *one*, and *now* are keywords in both corpora; the prevalence of 'love' is not particularly surprising given the emotive function of poetry. That said, the more modern Poetry Foundation+ corpus favors corporeal language (*eye*, *hand*, *head*, *face*) while the Gutenberg corpus favors the abstract (*heart*, *life*, *soul*) and religious (*God*, *heaven*). Notably, *man*, or its variant *men*, feature prominently in Gutenberg poems while *mother* makes frequent appearances in Poetry Foundation+.
- *Lexical Diversity*: To measure lexical diversity, I divide the vocabulary size (number of distinct words) by the total number of words[1]. By this metric, the Poetry Foundation+ corpus contains a richer lexicon than does the Gutenberg corpus.
- *Hapaxes*: Hapaxes are words that appear only once in a corpus. Refer to Table 2. Despite their rarity (or rather singularity), *pygmean*, *idolatresses* and *grunsel*—an obsolete spelling of groundsill—evince the dated nature of the Gutenberg corpus.

Corpus	Number of Tokens	Vocabulary Size	Lexical Diversity	Number of Hapaxes
Poetry Foundation+	6354113	190486	0.03	108607
Gutenberg	26774905	412469	0.015	217724

Table 1: Properties of Each Corpus

6.2. Preprocessing by the Numbers

Table 3 displays the statistics generated by ***Torch-RNN***'s preprocessing script. There is a significant disparity between the number of tokens outputted by ***NLTK*** and ***Torch-RNN***'s script, which results from different tokenization processes.

[1]The type-token ratio is a rather crude measure that "can produce erratic outcomes, especially when the numbers of tokens vary substantially between the texts to be compared" [7]; however, it suffices for my purposes.

Hapaxes	
Poetry Foundation+	Gutenberg
Northgate	PANDEMONIUM
cliffwork	idolatresses
ocean-hole	grunsel-edge
spookfish	sluiced
cephalopods	bullion-dross
Self-annihilation	pygmean
Wrong-Bodied	Hell-doomed
Lacrimal	furnace-mouth

Table 2: A selection of hapaxes

Corpus	Number of Tokens	Training Size	Val Size	Test Size
Poetry Foundation+	30233618	24186896	3023361	3023361
Gutenberg	121559783	97247827	12155978	12155978

Table 3: Preprocessing Statistics

6.3. Training by the Numbers

Table 4 displays training times for each corpus. The Poetry Foundation+ corpus took 13.3 hours of CPU time while the Gutenberg corpus took around 20 hours.

Corpus	Real	User	Sys
Poetry Foundation+	798m54.857s	672m41.340s	126m29.988s
Gutenberg	1204m44.361s	919m47.956s	285m25.812s

Table 4: Training Statistics

7. Training Results

7.1. Early Stage Training Samples

As a proof of concept, ***Torch-RNN*** was trained on prose, for which large corpora, such as free book PDFs, are readily available online.

For my first attempt, I trained *The Unbearable Lightness of Being* (676 KB) with default hyperparameters. Below is a sample generated after 90 minutes:

As I was you specialist, yet in love. Then what's voluny was not judge and more hotelf are wished their body's love.

As is apparent, some of the "words" aren't even recognizable English words. This is unsurprising given that 676 KB is an extremely small dataset for machine learning, and the RNN has to learn the rules of the language from scratch. However, the model already 'knows' punctuation (e.g. apostrophes and commas followed by a space) as well as the fact that 'words' are separated by spaces.

After that, I downloaded all seven volumes of *In Search of Lost Time*, which is collectively 7 MB—still small, but much better than *The Unbearable Lightness of Being*—and trained with a larger RNN size (300 versus the default of 128) and 3 layers instead of 2. Here is an excerpt:

It was the only blood of the moral of a person who had arranged to it the scene of the startled connexion of the continuous presence of the person who had been able to see them the person who had always been the secret who has heard the portrait of the stage.

The model clearly hasn't learned proper punctuation; the excerpt is a series of run-on sentences. However, there are no ill-formed words, the syntax is fairly regular, and some interesting turns of phrase (such as "the faces of my heart") were generated.

Next, I began to train the Poetry Foundation+ dataset. Below are two representative[2] sample outputs, with temperature = 0.4, checkpoint = 136,000, runtime = 4 hours):

Excerpt 1: *To sweet deformed dawn! while the sun smoked the others and dye/From the kids as the back door opens; by thugs come/As the singer emptiness only to accept. My father/The same vice the dead, of the odds, with the world/Throwing glass leaves bend numb*

Excerpt 2: *when the fire was all the way to the land where the cold clouds/were drawn by the stones the small strands of a window/starts to run their spines and the corners/of*

[2] with respect to coherence

the truck dripping out of the world/and the bright streets of the children stand up/and
watching the breakdowns of the scars of the trees.

The model has learned the caesura that many contemporary poets favor as a method of separating clauses. There are also rather novel phrases such as "the corners/of the truck dripping out of the world," but the excerpt lacks semantic coherence.

7.2. Final Stage Samples

Torch-RNN did not finish training either of the datasets. For the Gutenberg corpus, a write error occurred after around 20 hours (on the 7th epoch out of 50). This error, which appears to be non-deterministic, has been cited several times in the issues thread of the ***Torch-RNN*** GitHub repository and has yet to be resolved. I suspect that the Poetry Foundation+ corpus would have yielded appreciably better results if fully trained.

The last generated checkpoint was 389,000; checkpoints between 289,000 and 295,000, and temperature values between 0.4 and 0.6 (inclusive), were determined to yield the best samples. Here is an excerpt trained on the Poetry Foundation+ corpus, checkpoint = 290,000 and temp = 0.5:

the branches of the battle and the streetlamps
of the house, cannot be the same
as the bare flowers of the moon,
the shadow of the paper-colored fields,
and the planes still through the sun.

The last Gutenberg checkpoint was 227,000, from which I sampled exclusively. After extensive sampling, I judged that temperature values between 0.7 and 0.9 yielded creative but still convincing outputs. Below is an excerpt trained on the Gutenberg corpus, checkpoint = 227,000 and temp = 0.8:

While from the sandy mingled brook
A sacred westward brim of blue-eyed flame

Under the cloud's delighted head;

And blossom'd through the lightning water,

With tender furies falling by

As to thy verdant earth comes drown'd with light,

8. Survey Design

Participants were shown a portfolio of machine-generated and human-authored poems, then asked to rate each poem on a scale of 1 to 5 (1 = strongly disagree; 5 = strongly agree) with respect to four metrics:

- *Impact*: Was the respondent moved? Did any parts of the piece resonate?
- *Poeticism*: Did the respondent perceive the piece as 'poetic'?
- *Originality*: Did the respondent find the piece surprising in terms of syntax and/or word choice?
- *Bot or Not*: The Turing Test component—did the respondent think this piece was machine-generated?

8.1. Portfolio Composition

I selected fifteen poems for inclusion in the survey—five generated by Torch-RNN from the Poetry Foundation+ corpus, five from the Gutenberg corpus, and five written by human authors. Each human-authored poem is taken from a reputable literary publication. Three of the Gutenberg poems were sampled with *temperature* = 0.7 and two were sampled with *temperature* = 0.8; Poetry Foundation+ poems were sampled with *temperature* = 0.5, 0.7, 0.8, 0.8, and 0.9. All machine-generated pieces in the survey have $temp \geq 0.5$ to avoid overly-conservative samples which might attenuate my findings. I excerpted all pieces, both machine-generated and human-authored, but did not otherwise alter them.

8.2. Respondent Groups

There were two respondent groups. The first, consisting of students from Princeton, Arizona State University, Harvard, MIT, and the University of Pennsylvania, form my non-expert group. For

my expert group, I invited Princeton faculty from the English, Computer Science, Neuroscience, Psychology, and Philosophy departments. There were 86 respondents in all—7 expert, 79 non-expert.

9. Survey Findings

Each survey poem's average metric scores were computed by respondent group. The four metrics were impact, poeticism, originality, and Turing Test (Bot or Not); ratings were on a scale of 1 to 5, where 1 indicates strong disagreement (e.g., the respondent did not consider the piece impactful, poetic, original, or machine-like) while 5 indicates strong agreement. As Table 5 indicates, the same three poems, all human-authored, had the highest impact scores in both respondent groups. Interestingly, a machine poem (generated from the Poetry Foundation+ corpus) had the third-lowest Turing Test score in the expert group, indicating that it was widely considered human-like. For both respondent groups, however, the poems with the second-lowest and lowest Turing score averages were human-authored. One human-authored poem was widely rated as very machine-like, receiving the first- and second-highest Turing scores in the non-expert and expert groups, respectively. However, the other poems in the highest Turing score column were machine-generated.

Overall, both respondent groups were biased (to varying degrees) toward identifying pieces as machine-generated. Table 6 shows that the non-expert group's average Turing Test score for human-authored poems sat at 2.69 (out of 5) and the expert group's at 2.51. Both groups also identified a human-authored poem, David Michael Wolach's "Sebaceous a Poetics," as one of the

Group	Highest Avg Impact Score	Lowest Avg Turing Score	Highest Avg Turing Score
Non-Expert	3.92 (Emily Lee Luan's "Sunflowers") 3.76 (Kaveh Akbar's "Heritage") 3.39 (Victoria Chang's "Dear P")	1.43 (Emily Lee Luan's "Sunflowers") 1.99 ((Kaveh Akbar's "Heritage") 2.27 (Victoria Chang's "Dear P")	4.32 (David Michael Wolach's "Sebaceous a Poetics") 3.76 (Poetry Foundation+ generated) 3.44 (Poetry Foundation+ generated)
Expert	4.29 (Emily Lee Luan's "Sunflowers") 3.86 (Kaveh Akbar's "Heritage") 3.57 (Victoria Chang's "Dear P")	1.29 ((Kaveh Akbar's "Heritage") 2 (Emily Lee Luan's "Sunflowers") 2.14 (Poetry Foundation+ generated)	4.14 (Poetry Foundation+ generated) 3.71 (David Michael Wolach's "Sebaceous a Poetics") 3.57 (tie between Gutenberg and Poetry Foundation+ generated poems)

Table 5: Poem-Level Observations

Group	Impact	Poeticism	Originality	Turing
Non-Expert	3.04 (avg) 1.32 (std dev)	3.42 1.08	3.61 0.97	2.69 1.50
Expert	3.09 (avg) 1.36 (std dev)	3.31 1.08	3.37 0.88	2.51 1.46

Table 6: Metrics for Human-Authored Poetry

Group	Impact	Poeticism	Originality	Turing
Non-Expert	2.89 (avg) 1.18 (std dev)	3.37 0.97	3.47 0.89	3.22 1.27
Expert	2.6 (avg) 1.06 (std dev)	3.09 1.01	3.31 0.87	3.2 1.43

Table 7: Metrics for Poetry Foundation+ Generated Poetry

most machine-like pieces (indicated by a low Turing Test score). While distinguishing human- and machine-generated poetry proved a challenge to both the expert and non-expert groups, they were most moved by the same three human-authored pieces. In fact, in the non-expert group, there was a perfect correspondence between the three poems with the highest impact scores and those with the lowest Turing Test scores. In the expert group, however, a machine poem generated from the Poetry Foundation+ corpus made it into the top three most human-like pieces.

For the non-expert group, the Gutenberg machine poems were actually perceived as more human-like than the human-authored poems themselves. In contrast, one of the Gutenberg poems appears in the expert group's top three most machine-like pieces, suggesting that those in the expert group—particularly professors in the English department—were far more likely to see the forest for the trees, as it were, and detect the pieces' lack of thematic coherence. A Welch's t-test confirms this intuition: the expert group found the Gutenberg-generated poems less poetic and less original,

Group	Impact	Poeticism	Originality	Turing
Non-Expert	2.86 (avg) 0.99 (std dev)	3.70 0.84	2.95 0.89	2.58 1.12
Expert	2.66 (avg) 1.08 (std dev)	3.11 0.90	2.54 0.78	3.17 1.29

Table 8: Metrics for Gutenberg Generated Poetry

on average, than the non-expert group. There were no statistically-significant differences between the groups' average metric scores for the human-authored or Poetry Foundation+ generated poems.

To determine whether there were associations between different survey metrics, I ran pairwise Kendall's tau-b correlations for each respondent group. In the non-expert group, impact scores are positively correlated with poeticism and negatively correlated with Turing Test scores; these associations hold for the expert group, which also tended to rate more impactful poems as more original.

For the last stage of analysis, I aggregated the results for the two respondent groups and computed Pearson correlation coefficients to test for linear correlation. Emotional oscillation scores were found to be positively correlated with poeticism and negatively correlated with originality as well as Turing Test scores. That is, poems with rapidly-oscillating sentiment were rated as more poetic, less original, and less machine-like.

10. Conclusion

The initial hypothesis was that poems generated from free verse—that is, the Poetry Foundation+ corpus—would look more realistic, but it seems that those generated from formal, more archaic verse (the Gutenberg corpus) are far more convincing from a layman's perspective. Several non-expert respondents were interviewed after the survey was closed; a common refrain was that consistency at the sonic level, particularly the clear sense of rhythm, imposed a kind of artificial logic on the Gutenberg-generated pieces.

The survey results[3] reveal that most of my machine-generated poems cannot fool experts. While non-experts struggled to distinguish Gutenberg-generated poems from human-authored ones, they were largely unconvinced by pieces trained on the Poetry Foundation+ corpus. Most strikingly, survey respondents collectively gravitated toward the human-authored poems irrespective of their accuracy on the Turing Test component. But what if a Turing Test isn't the end goal? Machine-generated poems can be alien, nonsensical, but also delightfully novel—precisely because they

[3]The survey and its key can be found at this link.

are unconstrained by human experience, because they confound human understanding. As Dan Rockmore wrote in an article for *The New Yorker*, "if we stop assessing robots by their imitative powers—how close they get to mastering humanness—we might begin to appreciate their creative powers in a new light. We might even be moved by them." [11]

11. Acknowledgments

Many thanks to my adviser, Professor Brian Kernighan, for his guidance on this project; Dr. Zoe LeBlanc for her helpful suggestions; and everyone who took the time to evaluate my suite of computer- and human-generated poems—particularly Victoria Ritvo (Neuroscience), Dev Dabke (PACM), Professors Dolven (English), Fellbaum (Computer Science), Graziano (Psychology; Neuroscience), Halvorson (Philosophy), Martin (English) and Wolfson (English).

12. Funding

This work was supported by the Department of Computer Science at Princeton University.

References

[1] *Artificial Intelligence: What it is and why it matters*. [Online]. Available: https://www.sas.com/en_us/insights/analytics/what-is-artificial-intelligence.html

[2] *Char-RNN*. [Online]. Available: https://github.com/karpathy/char-rnn

[3] J. Brownlee, "How to configure the number of layers and nodes in a neural network," *Machine Learning Mastery*, 2018.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning (adaptive computative and machine learning series)*. The MIT Press, 2016.

[5] J. Hallman, *Poetry Foundation Dataset*. Available: https://www.kaggle.com/johnhallman/complete-poetryfoundationorg-dataset

[6] J. Hopkins and D. Kiela, "Automatically generating rhythmic verse with neural networks," Facebook Research, Tech. Rep., 2017.

[7] R. Hout and A. Vermeer, "Comparing measures of lexical richness," *In: H. Daller, J. Milton J. Treffers-Daller (eds.), Modelling and assessing vocabulary knowledge (93-116). Cambridge: Cambridge University Press.*, 01 2007.

[8] A. Karpathy, "The unreasonable effectiveness of recurrent neural networks," *Github*, 2015.

[9] C. Lamb, D. G. Brown, and C. L. Clarke, "A taxonomy of generative poetry techniques," *Journal of Mathematics and the Arts*, vol. 11, no. 3, pp. 159–179, 2017.

[10] C. Olah, "Understanding lstm networks," *Github*, 2015.

[11] D. Rockmore, "What happens when machines learn to write poetry," *The New Yorker*, Jan. 2020.

[12] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2009.

[13] Sciforce, "A comprehensive guide to natural language generation," *Medium*, 2019.

13. Appendix

Statistical Tests and Results			
Factor(s)	Statistical Test	P-Value	Finding
Impact (human)	Welch's T-test	0.42	No appreciable difference between impact scores among experts and non-experts
Poeticism (human)	Welch's T-test	0.29	No appreciable difference between poeticism scores among experts and non-experts
Originality (human)	Welch's T-test	0.07	No appreciable difference between originality scores among experts and non-experts
Turing (human)	Welch's T-test	0.25	No appreciable difference between Turing Test scores among experts and non-experts
Impact (Poetry Foundation+)	Welch's T-test	0.07	No appreciable difference between impact scores among experts and non-experts
Poeticism (Poetry Foundation+)	Welch's T-test	0.06	No appreciable difference between poeticism scores among experts and non-experts
Originality (Poetry Foundation+)	Welch's T-test	0.16	No appreciable difference between originality scores among experts and non-experts
Turing (Poetry Foundation+)	Welch's T-test	0.47	No appreciable difference between Turing Test scores among experts and non-experts
Impact (Gutenberg)	Welch's T-test	0.14	No appreciable difference between impact scores among experts and non-experts
Poeticism (Gutenberg)	Welch's T-test	<0.001	Mean poeticism score is lower for the expert group than for the non-expert group
Originality (Gutenberg)	Welch's T-test	<0.001	Mean originality score is lower for the expert group than for the non-expert group
Turing (Gutenberg)	Welch's T-test	<0.001	Mean Turing Test score is higher for the expert group than for the non-expert group

Table 9: Statistical Tests and Results

Statistical Tests and Results			
Factor(s)	Statistical Test	P-Value	Finding
Impact ↔ Poeticism (Expert)	Kendall's tau-b correlation	<0.001	Strong, positive correlation
Impact ↔ Originality (Expert)	Kendall's tau-b correlation	<0.001	Moderate, positive correlation
Impact ↔ Turing Test (Expert)	Kendall's tau-b correlation	<0.001	Moderate, negative correlation
Poeticism ↔ Originality (Expert)	Kendall's tau-b correlation	<0.001	Moderate, positive correlation
Poeticism ↔ Turing Test (Expert)	Kendall's tau-b correlation	<0.001	Moderate, negative correlation
Originality ↔ Turing Test (Expert)	Kendall's tau-b correlation	<0.001	Weak, negative correlation
Impact ↔ Poeticism (Non-Expert)	Kendall's tau-b correlation	<0.001	Moderate, positive correlation
Impact ↔ Originality (Non-Expert)	Kendall's tau-b correlation	<0.001	Trivial (non-existent) correlation
Impact ↔ Turing Test (Expert)	Kendall's tau-b correlation	<0.001	Moderate, negative correlation
Poeticism ↔ Originality (Expert)	Kendall's tau-b correlation	<0.001	Trivial (non-existent) correlation
Poeticism ↔ Turing Test (Expert)	Kendall's tau-b correlation	<0.001	Moderate, negative correlation
Originality ↔ Turing Test (Expert)	Kendall's tau-b correlation	0.48	No appreciable correlation
EOS ↔ Impact	Linear correlation	0.16	No correlation
EOS ↔ Poeticism	Linear correlation	0.006	Moderate, positive correlation
EOS ↔ Originality	Linear correlation	0.008	Moderate, negative correlation
EOS ↔ Turing Test	Linear correlation	0.023	Moderate, negative correlation

Table 10: Statistical Tests and Results, Continued