Injection of Creativity and Emotion-Elicitation in Poetry Generation

Brendan Bena
Drury University
900 N. Benton Ave.
Springfield, Missouri 65109

Jugal Kalita
UC-Colorado Springs
1420 Austin Bluffs Pkwy.
Colorado Springs, Colorado 80918

Abstract

Poetry Generation, in the context of Natural Language Generation (NLG), involves teaching systems to automatically generate text that resembles poetic work. A system learns to recreate poetry through training on a corpus of poems and modeling the particular style of language. In this paper, we propose taking an approach of fine-tuning GPT-2, a pre-trained language model, to our downstream task of poetry generation. Specifically, we attempt to create emotion-eliciting poetry and dream poetry. Our first goal is to elicit emotions within the reader through the automatically generated text, so we believe a crowdsourced human-evaluation is the proper form of metric. Our model for the emotions of *sadness* and *joy* produced poems that correctly elicited emotions 87.5 and 85 percent of the time, respectively. Our second goal is to apply transfer learning to inject creativity and produce dreamlike poetry. Poems from this model are shown to capture elements of dream poetry with scores of no less than 3.2 on the Likert scale. For further quantitative evaluation, we make use of the Coh-Metrix tool, outlining certain metrics we use to gauge the quality of text generated.

Introduction

Many natural language processing tasks require the generation of human-like language. Some tasks, such as image and video captioning and automatic weather and sports reporting, convert non-textual data to text. Some others, such as summarization and machine translation, convert one text to another. There are additional tasks that aim to produce text, given a topic or a few keywords. These tasks include story generation, joke generation, and poetry generation, among others.

Poetry generation produces creative content, and delivers the content in an aesthetically pleasing manner, usually following a specific structure. Thus, in addition to generating text as if in a story, the lines produced usually have a certain length, quite frequently there is a rhyming scheme as well as rhythm, and organization into structures such as couplets, quatrains, quintets, and stanzas. Among other things, creativity comes from unusual usage of words through effects such as alliteration, assonance, and elision; use of metaphors, symbolism, and other linguistic devices; licensing of underlying imagery with expressed feelings, sentiments and emotions.

Work in natural language generation can be traced to pioneering rule-based simulations of chatbots such as the "psychotherapist" Eliza (Weizenbaum and others 1966) and paranoid schizophrenia-suffering PARRY (Colby 1981). Surveys such as (Hovy 1990; Reiter and Dale 2000; Gatt and Krahmer 2018; Santhanam and Shaikh 2019) have described the progress in natural language generation over 50 years. Of late, the use of deep learning has produced enviable progress in natural language generation, especially in topics such as machine translation (Bahdanau, Cho, and Bengio 2014; Wu et al. 2016), image captioning (Mao et al. 2014) and dialogue generation (Li et al. 2016).

This paper discusses an attempt to generate natural-sounding poems that are creative and can potentially evoke a response from the readers or hearers in terms of emotions and feelings they generate. We choose dreams as our form of creative expression due to its long standing history in poetry. Dream poetry is dated back to medieval times where famous 14th century authors, like Chaucer, experiment using dreams as the structure for an image or picture they wish to paint with a poem (Spearing 1976). A dream poem is said to be characterized by the 'I' of the poem and its substances of a dream or a vision included (Lynch 1998). To the best of our knowledge, prior work on poetry generation, whether using deep learning or not, has not explored the incorporation of emotion-eliciting phraseology or elements of creativity like dream poetry as we do in this paper.

Our research provides the following contributions:

- The use of GPT-2 for poetry generation
- Leveraging a word-level emotion lexicon to categorize emotion-based text
- Exploration of injecting creativity in poetry through the use of dream data

This paper is organized in the following way. Section 2 presents related work. Next, section 3 discusses our approach to creative text generation including pre-processing steps and architecture used. Section 4 talks about our experiments and results. Finally, section 5 gives an evaluation of our research.

Related Work

Early methods for poetry generation made use of template oriented and rule-based techniques. These approaches often

required a large amount of feature picking and knowledge of syntactic and semantic rules in a language (Oliveira 2009; Oliveira 2012). Other methods treated poetry generation as special cases of machine translation or summarization tasks (Yan et al. 2013; He, Zhou, and Jiang 2012). Such methods did not have the ability to learn any aspects of the language in which the poems were written, and thus we feel that they were incapable of any injection of creativity in the generation process. Forcing a model to adhere to specific rules or templates, or summarizing or translating a given text to generate new poetry was unlikely to lead to artistically expressive quality we seek to create.

More recently, deep learning methods have become prevalent in natural language generation, including poetry generation. Zhang and Lapata (2014) used Convolutional (CNN) and Recurrent Neural Networks (RNN) to generate Chinese Poetry. RNNs allow for short-term memory of the language to be maintained by inputting the generated output of a network cell back into itself, essentially building context.

Ghazvininejad et al. (2017) used Long Short-Term Memory (LSTM) units, which are advanced gated versions of RNNs, to the task of poetry generation. Wei, Zhou, and Cai (2018) attempted to address the style issue by training the networks using particular poets and controlling for style in Chinese poetry. They found that with enough training data, adequate results could be achieved. The structure problem was addressed by (Hopkins and Kiela 2017). They generated rhythmic poetry by focusing on training the network on only a single type of poetry to ensure produced poems adhered to a single rhythmic structure. It was found in human evaluations that while the poems produced were rated to be of lower quality than human produced poems, they were indistinguishable from human produced poems. Lau et al. (2018) took the LSTM approach one step further with the *Deepspeare* model by employing an attention mechanism to model interactions among generated words. They also use three neural networks, one for rhythm, one for rhyming and another for word choice in their quest to generate Shakespeare-like sonnets.

Vaswani et al. (2017) developed a deep neural architecture called the Transformer that did away with any sort of need for recurrence. The Transformer also employed an elaborate attention mechanism that has been shown to be useful in natural language tasks. Radford et al. (2019) used this architecture in their Generative Pretrained Transformer 2 (GPT-2) model. GPT-2 is capable of many downstream tasks like text generation, but to our knowledge research has not been published using the GPT-2 model specifically for poetry generation.

On a slightly different but related note, natural language generation influenced by multi-modal input was attempted by (Vechtomova et al. 2018) to generate song lyrics in the style of specific artists by fusing outputs coming from lyrical inputs processed by an RNN and audio clips processed by a CNN. Text generation has also been influenced, in a cross domain manner, through images. The works of (Liu et al. 2018) have shown that coupled visual-poetic embeddings can be used to pick out poetic clues in images, which in turn can be used to inspire the generated text. Though influenced natural language generation in and of itself is not a novel idea, we feel our attempt to style text with the intent of eliciting particular emotions provides a creative way to explore this subtask.

Approach

Our work involves a preliminary step of scoring a corpus of downloaded poems for emotion to produce subsets of poems that express one of eight different identified emotions. This step is followed by the actual generation of poems by finetuning the pre-trained GPT-2 natural language model. Emotion poem generation involves training eight separate models, one on each type of emotion poem, to learn how to model a poetic style of language. These poems are evaluated using automated techniques as well as humans for the emotions they express or elicit in a reader. Finally, we describe how we attempt to introduce aspects of what is deemed as creativity in poetry into poems that are composed automatically. To do so, we gather dream data and apply transfer learning by finetuning on dreams, then again on poetry. A high-level overview of the emotion elicitation portion of our project is shown in Figure 1.

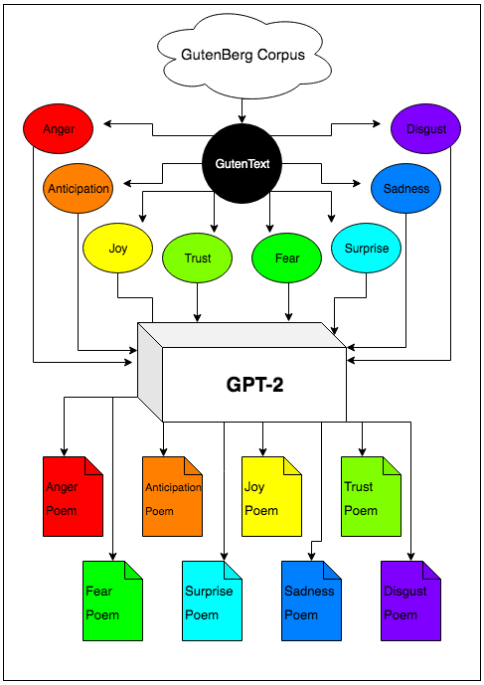

Figure 1: A high-level overview of our project implementation for emotion eliciting poetry

Poem Emotion Scoring

To decipher text depending on the emotions they elicit, we make use of the EmoLex dictionary (Mohammad and Turney 2013). EmoLex is a word-level emotion lexicon that associates English words with the 8 different emotion categories we wish to explore. Each poem (or book of poems) in our dataset is given a score that is the total of the associated

emotion scores in EmoLex for each word. The maximum emotion word score is taken and the poem is labeled under that emotion category. This classification method allows us to train multiple models on our split dataset.

Currently, the emotions of *joy, anticipation, trust, anger*, and *sadness* represent a large portion of our data while the emotions of *surprise, disgust,* and *fear* are severely underrepresented. Table 1 shows key differences in models including the number of tokens in the text and the final average loss during training.

GPT Architecture

To create a model for poetic language, we propose finetuning OpenAI's GPT-2 architecture. GPT-2 is a Transformer-based model that was trained simply to predict the next word in a 40GB text corpus (Radford et al. 2019). This 40GB dataset, *WebText*, was scraped from the internet with certain heuristics that aimed to gather only quality text (i.e. only outbound Reddit links from posts with a karma rating of 3 stars or better). By training on such a largely encompassing corpus of text, the architecture has proven to model the English language well and has obtained state-of-the-art results on downstream text-based tasks such as machine translation, question answering, and summarization. We leverage GPT-2's pre-trained knowledge of language for our downstream task of peotry generation.

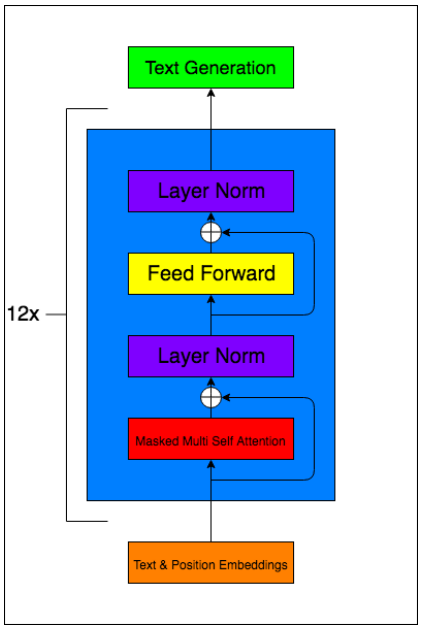

Figure 2: GPT Architecture. Adapted from (Radford et al. 2018; Radford et al. 2019)

GPT-2 (Radford et al. 2019) is the successor of OpenAI's first Transformer-based architecture, GPT (Radford et al. 2018), with a few changes to the structure. The medium version of GPT-2 we use contains 345M parameters and is a 24 layer, decoder-only Transformer architecture. GPT-2 moves layer normalization to the input of each sub-block, adds another layer normalization after the final self-attention block

Data	Model Size	# of Tokens	Final Loss
anger	345M	1,292,457	0.27
anticipation	345M	2,314,637	1.30
joy	345M	11,668,792	3.19
sadness	345M	2,090,915	1.03
trust	345M	16,667,178	3.39

Table 1: Comparison of 5 emotion models trained.

and increases context size from 512 to 1024 tokens. This architecture allows for long term dependencies to be captured better in language modeling. GPT-2's attention mechanism is referred to as a masked multi self-attention head. This technique allows for a relationship to be modeled for all words in an input sequence. Words that have multiple meanings can then be represented based on the context they appear in. Higher attention scores from surrounding words relate to a larger contribution to the representation of a word. GPT-2 makes use of byte-pair encoding (BPE) like its predecessor GPT but on UTF-8 byte sequences (Sennrich, Haddow, and Birch 2015). GPT-2's encoding is somewhere in between character level and word level. The model also prevents different versions of common words from being duplicated (i.e. *fate!*, *fate?*, and *fate* would not be joined). This technique improves the quality of the final byte segmentation. GPT-2's encoding rids the need for pre-processing or tokenization of data and is able to assign a probability to any Unicode string.

The task-agnostic nature of GPT-2 allows us to employ what we claim to be a semi-supervised fine-tuning approach to our downstream task of poetry generation. Though the GPT-2 model learns in an unsupervised manner, our poetry data is split into categories, so that we can train already pre-trained GPT-2 on a sub-corpus of poems that demonstrate a certain emotion or dream-like text without explicitly being told to do so.

Text Generation and Sampling

As stated by Radford (2019), the core approach of GPT-2 is language modeling. A language model can be thought of as a probability distribution over a sequence of words in the form:

$$p(w_1, ..., w_n) \tag{1}$$

Likewise, natural language tends to have a sequential order so it can be modeled as the probability of word given preceding words in the form (Bengio et al. 2003):

$$p(w_n|w_1, ..., w_{n-1}) \tag{2}$$

We make use of the probabilistic style of language modeling by sampling from the distribution in a semi-random fashion. Just as the GPT-2 paper does for its text generation, we make use of Top K sampling, limiting the possible guesses of words to 40. In addition to Top K, we make use of a temperature constant of 0.75 which controls randomness in the distribution. A temperature closer to 0 correlates to less randomness and a temperature of 1 relates to more randomness. Finally, at the end of the generation process, we

employ a simple text cleaning algorithm that allows poems to end more naturally and not trail off as they do sometimes.

Experiments and Results

Datasets and Resources

In order to classify emotion-eliciting poems or books, we use the NRC Word-Emotion Association Lexicon (EmoLex) resource. EmoLex was created by the National Research Council of Canada and includes 14,182 English words that are associated with different emotions and positive or negative sentiment (Mohammad and Turney 2013). Words in EmoLex have been manually annotated via crowd-sourcing and emotions fall into one or more categories of eight basic emotions: *joy, trust, fear, surprise, sadness, anticipation, anger, and disgust* (Plutchik 2014). This resource provides us with a way to fabricate a ground truth in the types of emotion-infused texts we wish to use for training data.

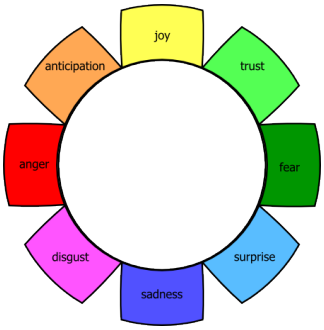

Figure 3: American pyschologist Robert Plutchik's Wheel of Emotions

To handle the training and generation portions of the project, we draw data from the Project Gutenberg website. Project Gutenberg is a massive online database containing over 59,000 eBooks. We limit this corpus to a smaller subcorpus using an adaptation of the GutenTag tool (Brooke, Hammond, and Hirst 2015). This tool allows us to place constraints on the amount of literature we choose to use in our work. Our final dataset includes approximately three million lines of poetic text from the Gutenberg database and is further divided by poem/book into our eight emotion categories.

We attempt to create dream poetry by making use of the *DreamBank* dataset. The *DreamBank* was created by Schneider & Domhoff at UC-Santa Cruz. The dataset contains a collection of over 20,000 dreams from users age 7 to 74. We scraped this dataset from the website assuring that dreams collected were recorded only in English. The *DreamBank* allows us to attempt transfer learning by finetuning on the dream dataset first, then further finetuning on our poetry dataset.

Amidst the chaos throng'd, with angry voices each
His rival's mockery; loud their scorn was fill'd;
So fierce their rage, and in their eager power
Met on the walls of Troy, were fill'd with dismay.

Figure 4: A hand-picked poem from the anger model

Heard I a song of joy,
A song of happy sound,
Fills all the air I breathe,
To him I sing, to him
I sing the happy song.
All night long on the steep green grass
I ride and sing

Figure 5: A hand-picked poem from the joy model

1 2 3 4

Initially, we have retrained 6 GPT-2 based models. Default training parameters were used each of the 5 different emotion datasets and our dream dataset. All were trained for 12,000 steps (other than our dream model that was trained for 12k steps on both dreams then poetry) with a learning rate of 0.0001. When generating text, we do not input context and allow the model to write the poem entirely through the sampling of conditional probability from the language it has modeled.

Figures 4 through 8 give examples of 5 poems that we have hand-picked to illustrate the quality of poems generated. A cursory glance at the poems shows that the quality of the text in terms of lexical choice, grammatically, and semantic cohesion is high. We discuss how we quantitatively assess the poems below.

[1] https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm
[2] https://www.gutenberg.org/
[3] https://www.dreambank.net/
[4] https://github.com/nshepperd/gpt-2

We have reached the peak of the highest mountain in the world
The mountain of dreams.
This is the view
Across the valley,
One hour's journey back,
We crossed it on the way between
A band of beautiful young women.
There was

Figure 6: A hand-picked poem from the anticipation model

A long trail of falling mist
Had made its way here, and now
Aerily it seemed, as if to drown
The discordant thunder clang.
It seemed to drown the music of the rain;
In this lost place of sorrow
Far off

Figure 7: A hand-picked poem from the sadness model

The other, who with one accord
Wrote my essay, in that he was dear
And good, and knew well, how we ought to treat
A man of such renown, and such love?
He's a good honest man, no doubt

Figure 8: A hand-picked poem from the trust model

A thousand stars at once,
An hundred thousand stars!
The sun was low,
And the stars were bright,
My heart would do the same.
A thousand stars at once,
A hundred thousand stars!
The night had begun,
And the stars were all the same.
When I came back from the dead,
I saw the stars

Figure 9: A hand-picked poem from the dream model

For she was mine.
I was the only one
She had,
And a thousand other friends,
And a hundred more
She held me dear.
Her eyes were clear, her cheeks were bright,
Her heart was like a rose,
Her mouth was full of music,
Her lips were white
As snow,
And the music she sang

Figure 10: A hand-picked poem from the dream model

Emotion	Anger	Antic.	Joy	Sadness	Trust
%	65	40	85	87.5	32.5

Table 2: Average percentage of correctly elicited emotion across four poems in each category

Poem	1	2	3	4
Qual 1	**5**	4.9	4.8	4.5
Qual 2	3.5	**4.1**	3.2	3.3
Qual 3	3.9	**4.2**	3.7	3.7

Table 3: Average Likert score of users for each poem

Evaluation

In the first crowd-sourced analysis of our emotion-eliciting poetry we presented 4 poems, of the five data-represented emotion categories, to ten human reviewers of undergraduate level educational backgrounds. These reviewers were asked to rate each poem based on the emotions elicited within them after reading. Table 2 illustrates the results from our evaluation. When taking the average percentage of correctly emotion-eliciting poems, the models of joy, sadness, and anger produced the most promising results while the trust and anticipation models were less than satisfactory.

To preserve consistency in our experiments, we evaluate our dream model poetry in a similar manner to the emotion poems. 4 poems from the model are presented to the same ten judges and they were asked to assess the poems based on qualities of dream poetry. A dream poem is said to have these qualities:

- The poem is generally a first-person expression
- The poem main substance is dream or vision like
- The poem recounts or foretells an experience or event

Analysis of results show that poems are able to capture the first person perspective well, achieving between 4.5 and 5 average Likert scores. The poems also appear to retell a story or an event often, scoring between 3.7 and 4.2 average Likert scores. The nature of poetry and dream recounts that make up our data is often narrative so this result stands to reason. However, Quality 2 scores of the poem substance containing a dream or vision are questionable. We suspect the Quality 2 score is lower due to the ambiguity in ascertaining dream text from regular text. Table 3 highlights our results for the dream model.

Currently, there exists no widely available standard for evaluating poetry generation. Scores like BLEU, ROUGE, METEOR, etc. are more suited for Machine Translation (MT) tasks (Zhang et al. 2019). For example, they compare how similar sentence P is to translated-sentence $\hat{P}$. Instead, we outline some metrics from the Coh-Metrix web tool that helps us further quantitatively evaluate the quality of text generated. With the goal in mind of eliciting emotions, we claim that subjective analysis of generated poetry will be superior to any available objective metrics.

Model	RDFRE	RDFKGL	WRDIMGc	WRDCNCc	LDTTRa	PCREFp	PCSYNp	PCNARp
anger	93.073	2.011	445.914	407.159	0.527	0.680	80.780	53.190
anticipation	100	0.832	440.931	403.104	0.404	7.780	83.650	81.860
joy	100	0.394	446.231	403.072	0.389	11.900	91.310	78.520
sadness	98.200	1.180	444.963	403.252	0.444	1.880	88.690	72.910
trust	100	0.156	434.664	412.717	0.334	18.140	84.610	91.310
dream	100	0	427.363	377.476	0.238	99.900	65.170	70.880

Table 4: Average Coh-Metrix evaluations across 25 randomly selected poems from each model.

Coh-Metrix

To provide a quantitative calculation of the caliber of text our models produce, we outline relevant metrics from the University of Memphis Coh-Metrix tool (Graesser et al. 2004). Coh-Metrix is a text evaluation software kit and from it, we have chosen 8 forms of assessment. The first two, Flesch-Kincaid Grade Level (RDFKGL) and Flesch Reading Ease (RDFRE), are two standard measures that deal with text readability and ease (Klare 1974). The RDFKGL scores a text from grade level 0 to 18, while the RDFRE score is a 0-100 index with 100 being an easily readable text. We aim to produce text that is readable by all, so a low RDFKGL score and high RDFRE score would be ideal. The next metrics we use evaluate at the word level. The word the imageability (WRDIMGc) and word concreteness (WRDCNCc) scores measure content words on their ability to create an image in the reader's mind and their ability to appeal to a reader's senses, respectively (Coltheart 1981). We aim for our art to create a connection between the reader and poem, so we believe imageability and concreteness of content words are two good measures with this in mind. We also make use of three text easibility principal component scores in narrativity (PCNARp), referential cohesion (PCREFp), and syntactic simplicity (PCSYNp) (Graesser et al. 2004). All text easibility PC scores are percentile scales, and thus we aim for higher numbers for these scores. Finally, we make use of the Lexical Diversity Type:Token Ratio score (LDTTRa) for all words. LDTTRa measures the ratio of *type* (unique) words to all *tokens* in the text. Because our text is relatively short, we aim for a middle ground in the LDTTRa ratio, meaning there is uniqueness in the word choice of the text, but cohesion is still upheld.

Inspection of our Coh-Metrix results show that randomly selected poems from all models fall at or below the 2nd-grade reading level in RDFKGL scores and are greater than 93 on the RDFRE scale. This suggests generated poems are easily readable by the majority of viewers. Looking at the WRDIMGc and WRDCNCc, we see our poems, except for the dream model concreteness, fall in the 400s. Words with higher imageability and concreteness fall around the low 600s while words that are lower fall around the upper 200s on this scale. These scores reveal that our models are creating text that is both concrete in word choice and paint a picture. Our dream model scoring lower in the concreteness is reasonable as the word choice of dreams tends to be more abstract. Lastly, percentile scores of PCSYNp and PCNARp show that the majority of models are producing poems that are both syntactically simplistic and narrative. Most PCREFp scores are on the lower end of the scale, but because the poems are not necessarily related and were all input at once, we suspect that is the reason these scores are lower. Table 4 highlights these scores for each poetry model.

Conclusion & Future Work

In this paper we attempted to influence natural language generation in the form of poetry generation through the use of classified emotion poems and dream text. To do so, we first leveraged a word-level emotion lexicon to construct a meaning for emotion-eliciting text and used that text to train separate language models. Next, we gathered data of dream records and employed transfer learning in attempts to create dream-like poetry. This paper seeks to create art in the form of auto-generated poetry while opening the door to more projects involving emotion-eliciting text-based tasks and influenced creative neural generation.

Future research in this project will involve gathering data for the underrepresented emotion categories, allowing us to have a language model for each emotion. We will also consider external crowd-sourced evaluation methods like Amazon Turk for a more expansive judgment of our results. In addition, we are interested in the exploration of using other word-level or segment-level emotion lexicons to influence our text generation. Finally, we wish to seek out additional forms of replicating creativity that artists incorporate in their work.

Acknowledgement

The work reported in this paper is supported by the National Science Foundation under Grant No. 1659788. Any opinions, findings and conclusions or recommendations expressed in this work are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[Bahdanau, Cho, and Bengio 2014] Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[Bengio et al. 2003] Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

[Brooke, Hammond, and Hirst 2015] Brooke, J.; Hammond, A.; and Hirst, G. 2015. Gutentag: an nlp-driven tool for digital humanities research in the project gutenberg corpus. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, 42–47.

[Colby 1981] Colby, K. M. 1981. Modeling a paranoid mind. *Behavioral and Brain Sciences* 4(4):515–534.

[Coltheart 1981] Coltheart, M. 1981. The mrc psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A* 33(4):497–505.

[Gatt and Krahmer 2018] Gatt, A., and Krahmer, E. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research* 61:65–170.

[Ghazvininejad et al. 2017] Ghazvininejad, M.; Shi, X.; Priyadarshi, J.; and Knight, K. 2017. Hafez: an interactive poetry generation system. In *Proceedings of ACL 2017, System Demonstrations*, 43–48. Vancouver, Canada: Association for Computational Linguistics.

[Graesser et al. 2004] Graesser, A. C.; McNamara, D. S.; Louwerse, M. M.; and Cai, Z. 2004. Coh-metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, Computers* 36:193–202.

[He, Zhou, and Jiang 2012] He, J.; Zhou, M.; and Jiang, L. 2012. Generating chinese classical poems with statistical machine translation models. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[Hopkins and Kiela 2017] Hopkins, J., and Kiela, D. 2017. Automatically generating rhythmic verse with neural networks. In *Proceedings of the 55th Annual Meeting of the ACL (Volume 1: Long Papers)*, 168–178.

[Hovy 1990] Hovy, E. H. 1990. Pragmatics and natural language generation. *Artificial Intelligence* 43(2):153–197.

[Klare 1974] Klare, G. R. 1974. Assessing readability. *Reading research quarterly* 62–102.

[Lau et al. 2018] Lau, J. H.; Cohn, T.; Baldwin, T.; Brooke, J.; and Hammond, A. 2018. Deep-speare: A joint neural model of poetic language, meter and rhyme. *CoRR* abs/1807.03491.

[Li et al. 2016] Li, J.; Monroe, W.; Ritter, A.; Galley, M.; Gao, J.; and Jurafsky, D. 2016. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

[Liu et al. 2018] Liu, B.; Fu, J.; Kato, M. P.; and Yoshikawa, M. 2018. Beyond narrative description: Generating poetry from images by multi-adversarial training. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, 783–791. New York, NY, USA: ACM.

[Lynch 1998] Lynch, K. L. 1998. *Medieval Dream-Poetry*. Cambridge University Press.

[Mao et al. 2014] Mao, J.; Xu, W.; Yang, Y.; Wang, J.; Huang, Z.; and Yuille, A. 2014. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*.

[Mohammad and Turney 2013] Mohammad, S. M., and Turney, P. D. 2013. Crowdsourcing a word-emotion association lexicon. 29(3):436–465.

[Oliveira 2009] Oliveira, H. 2009. Automatic generation of poetry: an overview. *Universidade de Coimbra*.

[Oliveira 2012] Oliveira, H. G. 2012. Poetryme: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence* 1:21.

[Plutchik 2014] Plutchik, R. 2014. *Emotions*. Psychology Press.

[Radford et al. 2018] Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

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

[Reiter and Dale 2000] Reiter, E., and Dale, R. 2000. *Building natural language generation systems*. Cambridge university press.

[Santhanam and Shaikh 2019] Santhanam, S., and Shaikh, S. 2019. A survey of natural language generation techniques with a focus on dialogue systems-past, present and future directions. *arXiv preprint arXiv:1906.00500*.

[Sennrich, Haddow, and Birch 2015] Sennrich, R.; Haddow, B.; and Birch, A. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

[Spearing 1976] Spearing, A. C. 1976. *The High Medieval Dream*. Stanford University Press.

[Vaswani et al. 2017] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is all you need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc. 5998–6008.

[Vechtomova et al. 2018] Vechtomova, O.; Bahuleyan, H.; Ghabussi, A.; and John, V. 2018. Generating lyrics with variational autoencoder and multi-modal artist embeddings. *CoRR* abs/1812.08318.

[Wei, Zhou, and Cai 2018] Wei, J.; Zhou, Q.; and Cai, Y. 2018. Poet-based poetry generation: Controlling personal style with recurrent neural networks. In *2018 International Conference on Computing, Networking and Communications (ICNC)*, 156–160. IEEE.

[Weizenbaum and others 1966] Weizenbaum, J., et al. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM* 9(1):36–45.

[Wu et al. 2016] Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

[Yan et al. 2013] Yan, R.; Jiang, H.; Lapata, M.; Lin, S.-D.; Lv, X.; and Li, X. 2013. I, poet: automatic chinese poetry composition through a generative summarization framework under constrained optimization. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

[Zhang and Lapata 2014] Zhang, X., and Lapata, M. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 670–680.

[Zhang et al. 2019] Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. Bertscore: Evaluating text generation with BERT. *CoRR* abs/1904.09675.