Haiku Generation using Word Associations
COMP 550 Project

Karim Koreitem (260460964) and **Yi Tian Xu (260520039)**
McGill University

Abstract

We formulate a method for automatic haiku generation based on the importance of association in creativity. Our model takes input words, generates an outline using a word association model and finally outputs a free form 3-line haiku through a word to sentence Recurrent Neural Network. Generated haikus show similar association score to human haikus based on our metric.

1 Introduction

Poetry, known as the oldest form of literature, uses linguistically appealing qualities such as figure of speech and sound patterns to grant ideas immortality. At the beginning of this millennium, interest spread in developing automatic methods for generating poetry in Computational Creativity and Natural Language Generation. In particular, the short and concise free form 3-line haikus become famous candidates for pioneers in this research area.

In this project, we propose a high level method for generating free form 3-line Haikus. Given user input word(s), it construct an outline based on a word association models, and then composes the haiku using a word to line representation based on Recurrent Neural Network (RNN) model. Our experiment shows some potential in generating simple short 3-line Haikus with a pinch of creativity, controllable by the user.

2 Related Works

Early automatic methods for poetry generation back in the 1960s were often in the form of combinatorial processes to existing poems. State of the art approach usually involves a language model (e.g.: Markov model, Recurrent Neural Network (RNN), etc.) to capture *grammaticality* and to create aesthetically and creatively pleasing text. (Oliveira, 2017) The leniency of certain language rules and the presence of linguistic phenomena such as figurative language and *poeticness* (i.e.: the recognizable form of poetry; respecting rhyme, meter, etc) distinguishes poetry from other natural language forms. According to (Manurung, 2004), beside *poeticness*, poem should also obey *grammaticality* (i.e.: linguistic conventions) and hold *meaningfulness* (i.e.: bearing some meaningful interpretation).

2.1 Haikus

Haiku, a poetry form originating from Japan, uses extremely economical linguistic form to evoke emotions. The traditional form is 3 lines of 5, 7 and 5 syllables, involving reference to nature. Modern English free from haikus often break the line, syllables and theme restrictions, but maintains concision (Netzer et al., 2009). We adopt the definition of free form haikus, while keeping the 3-lines quality as the only concrete *poeticness* constraint.

2.2 Word Associations

Associations - connecting two ideas to evoke something new - has long been recognized as a cornerstone in creativity (Mednick, 1962; Benedek et al., 2017). Association resources can be in the form of word thesauri created by collecting human responses to cue words (De Deyne and Storms, 2008; Nelson et al., 2011; Rotmistrov, 2014). (Netzer et

al., 2009) found that haikus have more associations than news articles and proses, and these associations can can be found more often in Word Association Norm (Nelson et al., 1998) than in WordNet (Miller et al., 1998). Inspired by this idea, we focus on the creative aspect in the *meaningfulness* of a poem, and use association as the main element to induce creativity.

2.3 Word Embedding

Recent unsupervised methods for learning word embedding such as skip-gram, CBOW (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) can be efficient solutions for modelling word associations. Based on the distributional semantics hypothesis stating that words with the same neighboring distribution have the same meaning, they map words to dense vectors such that a distance-based function can estimate the similarity between word meanings. In addition to their word compression capabilities, these models have demonstrated potentials in the persistence of meanings when performing arithmetic operations on the embedded space, which can be an advantage for modeling association score functions.

2.4 Recurrent Neural Networks

Recent methods for poetry generation have relied on Recurrent Neural Networks (RNNs) to perform the poem generation task by, for instance, learning character models and conditional probabilities of characters given a certain context (Ballas, 2015; Zhang and Lapata, 2014; Wang et al., 2016). For example, an initial line is generated from user input keywords by relying on some corpus of poetic lines, and an RNN is trained on character distributions using a corpus of poems. Similar to these methods, we employ RNNs to learn Haiku structures and generate new plausible poetic lines. Yet, instead of operating on the character level, we use the word as the smallest linguistic construct. In particular, we are interested in exploring the word to poem line relationship and model our generation task over that relationship.

3 Method

At the core of our method, we assume that certain words between the lines in a haiku have associations that are creatively appealing, which is analogous to the ideas in (Netzer et al., 2009). Our method relies on a word embedding trained on a haiku corpus to model relevant word associations in haikus. We collected a corpus of more than 34K 3-lines haikus by crawling around 22.8K from online haiku publications[1], 11.2K from social media[2], as well as 0.2K translated works from historically famous Japanese poets Matsuo Bashō and Yosa Buson[3].

3.1 Word Association Construction

We employ GloVe (Pennington et al., 2014) to train our word embedding. This model create a co-occurrence matrix of word pairs within a window size of ω. Then it applies factorization to obtain normalized unit vectors on a d-dimensional space. GloVe results high cosine similarity between two vectors if their corresponding words have similar distribution of neighboring words, We can therefore consider words appearing within a haiku or consecutive lines of a haiku, and other words with similar neighboring distribution to be highly associated.

The word embedding space is trained without punctuation and stopwords. We use standard functions in Python NLTK package for tokenization and stopword removal. The embedding space is then used to construct outline-haiku pairs, which are used to train the RNN in the surface realization stage (see Section 3.2.3), as well as for computing the range of association score that is suitable for constructing the outline in the planing stage. We consider two association score functions. First, given two word vectors, u, v, in the embedded space, D, with corresponding vocabulary set, vocab(D),

$$S_1(u, v) = \text{cos\_sim}(u, v) \quad (1)$$

[1]Daily Haiku: `http://www.dailyhaiku.org/`,
The Heron's Nest: `http://www.theheronsnest.com/`,
Temps Libres: `http://www.tempslibres.org/tl/tlphp/dblang.php?lg=e`,
Haiku Society of America: `http://www.hsa-haiku.org/haikucollections.htm`,
Haiku Foundation: `https://www.thehaikufoundation.org/haiku-registry`,
Modern Haiku: `http://www.modernhaiku.org/`,
Best Haikus of All Time: `http://www.thehypertexts.com/Best%20Haiku.htm`

[2]Twitter with "#haiku": `https://twitter.com/search?q=%23haikusrc=typd&lang=en` , the Haiku subreddit: `https://www.reddit.com/r/haiku/`, blogs and other websites like `https://mikhaemoji.wordpress.com/2016/06/08/haiku-corpus-1/` and `http://writeahaiku.com/`

[3]`https://www.poemhunter.com/`

Second, given three word vectors $u, v, w \in D$,

$$S_2(u, v, w) = \text{cos\_sim}(u + v, w) \tag{2}$$

One can interpret that Equation 1 compares between two words, and Equation 2 compares between the combination of two words and another word.

3.2 Haiku Generation

Our haiku generation approach is similar to the Chinese poem generation approach by (Wang et al., 2016), which is inspired by the idea that human poets usually makes an outline before writing a poem. Our method therefore has two steps: (1) given user input word(s), it finds associated words to construct an outline, with one word for each line of the haiku (*planning*); and (2) it expands each word into a line through a RNN (*surface realization*).

3.2.1 Outline Reconstruction

Let $O[h]$ be a triple representing the outline of haiku h. Let "topic word" denote a word w_i such that $\exists h : O[h]_i = w_i$, which satisfies 2 conditions: (1) w_i has a corresponding vector in D (i.e.: $w_i \in \text{vocab}(D)$), and (2) w_i is present in h at line $1 \leq i \leq 3$.

To reconstruct $O[h]$ for each haiku h in the corpus, consider each candidate topic word w' in h, such that w' satisfies $w' \in \text{vocab}(D)$. Given all candidate topic words $T_h = T_{h_1} \cup T_{h_2} \cup T_{h_3}$ where T_{h_i} is the set of candidate topic words for line h_i, we pick the top n triples, (w'_1, w'_2, w'_3), where $w'_i \in T_{h_i}$ for each i such that Equation 3 is maximized

$$\begin{aligned} \text{S}^a(w'_1, w'_2, w'_3) &= \text{avg}(S_1(w'_1, w'_2), S_1(w'_2, w'_3)) \\ \text{S}^b(w'_1, w'_2, w'_3) &= \max_{(i,j,k)\in P_3} S_2(w'_i, w'_j, w'_k) \end{aligned}$$

$$S_3(T_h) = \max_{w'_1, w'_2, w'_3} \left(\max_{f \in \{S^a, S^b\}} f(w'_1, w'_2, w'_3) \right) \tag{3}$$

To obtain the range of association score that is suitable for the planing stage, we compute the mean and variance, $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2$, of $S_1(w_1, w_2)$ and $S_2(w_1, w_2, w_3)$ for all outlines $(w_1, w_2, w_3) \in \{O[h]\} \forall h$, respectively, across all reconstructed outlines.

3.2.2 Planning

We consider the case where the user either gives one or two input words. By default, all inputs are considered to be topic words (unless they are not in vocab(D)) and each correspond to a line of the generated haiku according to the input order. If user inputs ones word, w_1, we randomly pick w_2 such that $S_1(w_1, w_2) \sim \mathcal{N}(\mu_1 + \alpha\sigma_1, \beta^2\sigma_1^2)$, and w_3 such that $S_2(w_1, w_2, w_3) \sim \mathcal{N}(\mu_2 + \alpha\sigma_2, \beta^2\sigma_2^2)$, where α and β are parameters that shift the mean and rescale the variance. Similarly, if the user inputs two words, we only need to pick w_3 in the same manner. We call the distribution $\mathcal{C}_i(\alpha, \beta) = \mathcal{N}(\mu_i + \alpha\sigma_i, \beta^2\sigma_i^2)$ the *creative range* as a high range allows words with strong associations, while a low range allow words that are rarely associated.

3.2.3 Surface realization

We formulate an RNN that can sample haiku lines given an input words from the outline, trained on the outline-haiku pairs constructed in Section 3.2.1. We first convert the outline-haiku pairs into word-line pairs for training. Let L_{max} represents the longest sentence in our corpus. Each line is padded by an endline characters until it reaches a length of $L_{max} + 1$. This padding helps the model to learn the general length of haiku lines and where to end a line. Using word vector representation, each word is converted to a vector of size W, where W is the dimension of the vector space, and each line, to a $(L_{max} + 1) \times W$ matrix.

Our RNN is a 3 layer LSTM neural network with 256 memory units to regress over the word-line pairs using the Adam optimized algorithm for gradient descent (Kingma and Ba, 2014). Semantic and coherence constraints (e.g: grammaticality) are naturally captured by the recurrent neural network architecture through learning the representation of sentences present in haikus.

We explore representing words using a 1-hot encoding embedding and a CBOW word2vec embedding (Mikolov et al., 2013). The word2vec embedding compresses each word representation to a 100-dimensional vector with values between $[-1, 1]$ and is trained on a Wikipedia pages corpus. We used a python interface to the Google word2vec repository (Rodriguez, 2015). The 1-hot encoding version requires to setup the RNN as a multi-class classifi-

cation task with a softmax activation layer at the end and a categorical crossentropy error. However, we notice that due to the large number of unique words in the corpus (i.e.:more than 22K words from 34K haikus), framing the problem as a muti-class problem may be unreasonable. Thus, we choose to use the CBOW word2vec embedding and train the RNN architecture with a *tanh* activation end layer (to follow the shape of the embedding) and with a mean squared error loss function. The output vectors are mapped to the closest word vectors in the embedding. Our architecture is implemented using Keras (Chollet and others, 2015), a high level python deep learning library.

3.3 Word Embedding for Associations

We consider two ways of modeling word associations. The first way treats each haiku as one text document (FullText). The second way treats each pair of consecutive lines as one text document (LinePair). In both case, we run GloVe with the window size, ω, set to a large number to include all words in each document, and the dimensionality of the vectors, to 50. Infrequent words (occurring less than 5 times) are dropped. To increase the vocabulary size, we also experiment on adding the limerick corpus from (Ballas, 2015) to the training (FullText+ and LinePair+).

After training, we evaluate the average S_3 score (from Equation 3 of the top $n = 3$ reconstructed outlines across different haiku resources for different word embeddings, including the pre-trained embeddings with Twitter (Twitter, 1M vocabulary words) and Wikipedia 2014 and Gigaword 5 (Wiki+Giga, 400K vocabulary words) corpora from (Pennington et al., 2014), both with 50 dimensions (see Figure 1). Due to the difference in training data and window size, one cannot directly compare the S_3 score between different embeddings, as for example the neighbouring distribution of a word in poetic texts can be much more uniform-like than in Twitter tweets. However, we observe that Twitter and Wiki+Giga embeddings appear to give smaller range for the average S_3 score of online published haikus compared to the embeddings that we trained, suggesting that those embeddings are significantly different. Indeed, the haiku quality from online publications are often better than those on social media.

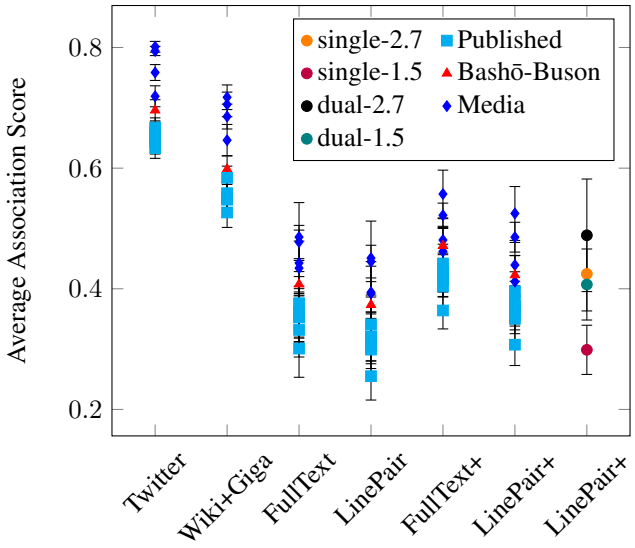

Figure 1: Average association score of the top 3 reconstructed outlines using different word embeddings and different haiku resources (*Published* for online publications; *Bashō-Buson*; from Bashō and Buson; *Media*, from social media; and the generated haikus with our method: *single* for single word query; *dual* for dual word query, and 1.5 and 2.7 for α).

Interestingly, although we have more than twice as many haikus from online publication than from social media, our trained embeddings almost seem to give an even range of average S_3 score for both of them. Moreover, Bashō and Buson's haikus appear to have an average S_3 score falling between the other two for all embeddings.

Pairing the lines duplicates the middle line in each haiku, thus increases the vocabulary size. Although this can causes a bias in the co-occurence matrix, qualitatively, we find that all our trained word embeddings give equally appealing associations and outlines. We therefore take the one with the largest vocabulary size, and compute that $\mu_1 = 0.187$, $\mu_2 = 0.238$, $\sigma_1 = 0.256$ and $\sigma_2 = 0.238$ for LinePair+. We observe that high creative range indeed generate more common associations while low creative range generates associations that are less intuitive, yet arguably with some potential in *meaningfulness* (see examples in Table 1).

3.4 Haiku Generation

We notice an improvement in haiku quality when the creative range is high. We then generate 50 haikus

Input	Completion	α, β
sun	lump, boyhood	-0.3, 0.4
sun	cabin, history	0.0, 0.2
sun	sky, arab	1.2, 0.1
sun	morning, light	2.4, 0.3
wind, freeze	farewell	-0.3, 0.4
wind, freeze	baltic	0.0, 0.4
wind, freeze	torrential	1.2, 0.2
wind, freeze	rain	2.4, 0.5

Table 1: Examples of generated outlines with different creative ranges.

Outline	Haiku	α
woman men homeless	the boy baby dead guests desolate nests	2.7
clean **beach** home	smoke smoke mound trail guests ride	2.7
account portals humanoid	the mankind roadways clogged playfulness discomfort	1.5
may **drizzle** breeze	going sight cervix osmotic stratosphere shower	1.5

Table 2: Selected examples of generated haikus, where input topic words are in bold.

for each $\alpha \in \{1.5, 2.7\}$ and $\beta = 1$, and using a single or dual word query (see examples in Table 2). We use the 1000 most common words (Heise, 1965) for single word queries, and first two words in the reconstructed outlines of corpus haikus for dual word queries.

We observe that the majority of the generated haikus contain repeated words, and topic words are usually absent, which may be due to the small amount of data and the mapping between the output of the RNN and the least distant word vector.

Using the procedure for reconstructing outline (see Section 3.2.1), we compute the average S_3 score for the generated haikus. Figure 1 shows that the range of the S_3 score matches closely between the generated haikus and the haiku from our corpus. In particular, high creative range (with $\alpha = 2.7$) is shown to result high S_3 scores. Additionally, dual word query appears to give higher S_3 scores compared to single word query, which is expected as we took the query words from the haiku corpus.

4 Discussion and Future Work

Qualitatively, our method can successfully generate outlines that contains common and creative associations, and 3-line haikus that exhibit some degree of *grammaticality* and *meaningfulness*. One advantage in our RNN construction is that our method is guaranteed to generate real words. However, it suffers from two drawbacks: (1) the presence of many endline markers in the training process biases the generation to prefer short lines (i.e.: an average of 2 to 3 words), resulting in short haikus; and (2) structural coherence is lost as the order of the lines is ignored, forbidding us from generating traditional 5-7-5 haikus for example.

The usage of word embeddings, although improves the efficiency and the result of our method, poses a constraint on the input (i.e.: topic words must exists in the word embedding) and on the RNN's output as different embeddings, depending on the training procedure, have words with different neighborhood in the embedded space. For example, the last generated haiku on Table 2 contains "cervix", "osmotic" and "stratosphere", which are scientific words present in Wikipedia pages, but not in our haiku corpus. One solution to explore is to use a word embedding trained on haikus, or poems for the surface realization. The training corpus must be large enough to ensure that all words in the haiku corpus for training the RNN are captured.

Another concern is the heuristics used for outline reconstruction and planning, which may not be optimal for the downstream tasks. For instance, humans are not restricted to construct outlines of one word per line. Future work includes a stronger model for the planning stage, as well as improvement on the RNN architecture, more quantitative analysis on the results, and enlargement of the training corpus.

5 Statement of Contributions

Yi Tian Xu collected the haikus and formatted the corpus, constructed the word association model (for outline reconstruction and planning). Karim Koreitem set up the RNN architecture and experimented with the embedding models for representing the data in the neural network. Both worked on the report and shared ideas throughout the entire process.

References

Sam Ballas. 2015. Generating poetry with poetrnn. http://sballas8.github.io/2015/08/11/Poet-RNN.html.

Mathias Benedek, Yoed N Kenett, Konstantin Umdasch, David Anaki, Miriam Faust, and Aljoscha C Neubauer. 2017. How semantic memory structure and intelligence contribute to creative thought: a network science approach. *Thinking & Reasoning*, 23(2):158–183.

François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

Simon De Deyne and Gert Storms. 2008. Word associations: Norms for 1,424 dutch words in a continuous task. *Behavior Research Methods*, 40(1):198–205.

David R Heise. 1965. Semantic differential profiles for 1,000 most frequent english words. *Psychological Monographs: General and Applied*, 79(8):1.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Hisar Manurung. 2004. An evolutionary algorithm approach to poetry generation.

Sarnoff Mednick. 1962. The associative basis of the creative process. *Psychological review*, 69(3):220.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

George Miller, Christiane Fellbaum, Randee Tengi, P Wakefield, H Langone, and BR Haskell. 1998. *WordNet*. MIT Press Cambridge.

DL Nelson, CL McEvoy, and TA Schreiber. 1998. The university of south florida word association, rhyme, and word fragment norms.

Douglas L Nelson, Cathy L McEvoy, and Th A Schreiber. 2011. University of south florida free association norms.

Yael Netzer, David Gabay, Yoav Goldberg, and Michael Elhadad. 2009. Gaiku: Generating haiku with word associations norms. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 32–39. Association for Computational Linguistics.

Hugo Gonçalo Oliveira. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 11–20.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Daniel Rodriguez. 2015. Python interface to google word2vec. https://github.com/danielfrg/word2vec.

YA Rotmistrov. 2014. Word associations network. https://wordassociations.net/en.

Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. 2016. Chinese poetry generation with planning based neural network. *arXiv preprint arXiv:1610.09889*.

Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *EMNLP*, pages 670–680.