Algorithmic poetry - comparing two forms of computer generated haiku

GROUP Awesome-o

Vanya Avramova	Rémi Domingues	Ahmed Kachkach	Johan Wikström
19760827	19920604	19920321	19920815
avramova@kth.se	remido@kth.se	kachkach@kth.se	jwikst@kth.se




Abstract

In this project, we investigated how to automatically generate haiku - a form of very short Japanese poetry. We compared two methods for haiku generation by utilizing two data sources that provide related words with different semantic metric for "relatedness": Word Association Norms (WAN) and WordNet [8]. A prototype haiku generator was created, using the Natural Language Toolkit (nltk) [7], WordNet [8] and Word Association Norms (WAN) for Python [6]. In the pre-processing phase of the haiku generation process, a crawler was implemented that collects existing haiku poems from the web, and converted them to json [13] format. Then, the lexical class of each word was identified, and common grammatical structures for haiku lines were extracted. In the second phase, a haiku generator was implemented that can output a haiku verse given a seed word as the target haiku theme. Since haiku poetry relies heavily on lexical associations for content [1], two providers for related words were implemented: one based on WAN, and another based on WordNet.

Previous work in this area had indicated that haiku generators based on WAN would perform notably better than generators based on WordNet [1]. We designed a survey to test this claim, in which human created haiku were compared side by side with our haiku generated using WAN and WordNet. The haiku were rated on a scale of 1-5, and their origin designated as "human" or "artificial". While the human-written haiku were liked notably better than the artificially generated ones (avg. rating 3.38 out of 5), we found that the difference between the WAN and WordNet generated haiku was not that pronounced, with an average rating of 2.80 and 2.59, respectively. Additionally, 43% of the WAN generated haiku and 36% of the WordNet generated haiku included in the poll were marked as written by humans.

1. Introduction

Commenté [JW1]: Introduction: provides helicopter view which puts the work into context

Commenté [JW2]: Introduction: describes what contribution(s) is/are made at a conceptual level.

Commenté [JW3]: Related Work: There is at least relevant 1 scientific paper being referred to.

A Haiku in English follows a set of structural and grammatical rules. Traditionally, it consists of 17 syllables on 3 lines according to a 5-7-5 pattern. Modern haiku, however, usually do not follow this constraint. It also usually is in present tense. Semantically, a Haiku usually contains a seasonal reference, but contemporary Haiku authors may choose to forego this rule. The most characteristic aspect of the Haiku is its power to generate a resonating image within the reader, usually by employing a juxtaposition between two concepts. The first concept is introduced on the first line, the second on the last line, and the second line must tie the two together. Conveying this imagery and some coherent meaning is one of the more challenging aspects of the project.

Computer generated haiku is generally a multistage process including selecting a subject, selecting suitable content related to that subject and deciding a grammatical structure for the haiku. It may also include over-generation and purging of generated haiku to increase quality of remaining haiku. This requires some heuristic to measure quality of haiku.

In this paper, we have chosen to focus on the content selection phase since our primary source [1] asserts that content selection based on Word Association Norms (WAN) may be superior to other common means of associating words, such as WordNet. Word Association Norms is a tool and dataset from psychology that has been used for several different research projects in varying areas [4]. It consists of a large dataset containing associations between stimulus words and responses gathered by asking more than 6000 test subjects to write down what word that came to mind when given a certain stimulus word. These response words are then recorded and stored as a mapping from stimulus to response word. WordNet, on the other hand, is a lexical database for English [5] containing definitions, synonyms and more for some 155000 nouns, verbs, adjectives and adverbs.

1.1. Contribution

In [1], the authors claim that WAN may be superior to WordNet for haiku content generation and provide some evidence to support this claim. They do not, however, implement both a WordNet-based and WAN-based haiku generator to compare their outputs.

In this work, we have implemented two different means of generating content, one based on WordNet and one based on WAN. These content generators are included in a multi stage haiku generation process which is then evaluated by letting human test subjects examine the resulting haiku. The results from these experiments show that while haiku generated from WAN were liked better than haiku generated from WordNet (supporting what is claimed by [1]), the differences were not that pronounced, and over 35% of both types of haiku passed as generated by a human in the survey.

1.2. Outline

This paper describes how we implemented a haiku generator loosely following the idea described in [1]. It starts with a review of related work, followed by a detailed description of our method. Generating haiku is a multistage process and each subsection of our method description describes one stage of the process. Then we introduce and explain the experimental setup of the project followed by graphs, tables and calculations that investigates our data. Lastly, we summarize the results and also discuss what would be needed to improve our results.

2. Related work

There are several attempts of creating haiku generators on the web and strategies include generating random words focusing on the syllable constraint [9], choose random lines from existing haiku [10], scanning texts for sentences that match the syllable constraint [11] or replacing some key words in existing haiku [12].

But there are also some attempts from the scientific community to generate haiku from scratch. Netzer et al. [1], focus on the generation of whole haiku from just a few key words. In [1] the authors investigate WAN as a way to generate creative texts, in particular haiku. They also divide the Haiku generation process into five phases: theme selection, syntactic planning, content generation, filtered over-generation and re-ranking. Our implementation is a multistage process very similar to the one in [1], with the addition of a special handling for prepositions using Markov Chains.

3. Our method

To generate a haiku we POS-tag (Part of Speech-tag) a haiku corpus, extract common grammatical structures, extract common stop words and prepare Markov Chains for prepositions (pre-processing phase, Fig. 1).

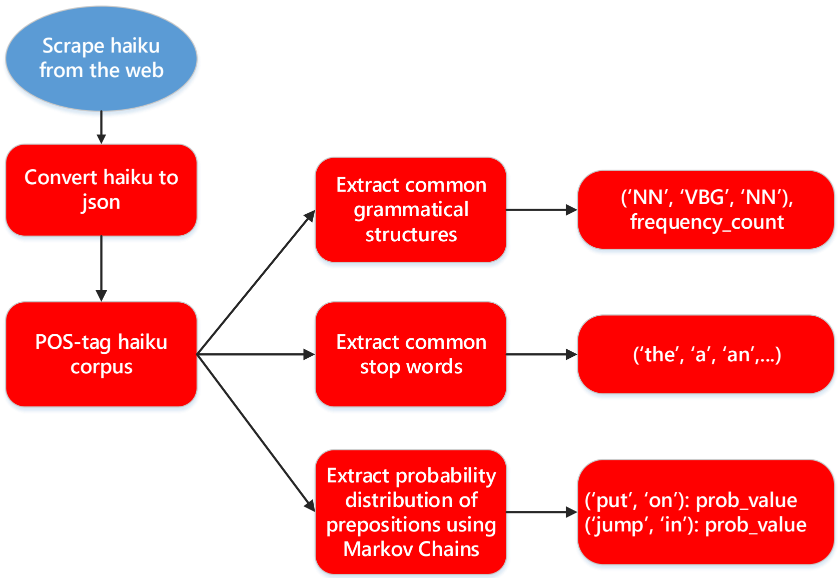

Fig. 1 Pre-processing phase of the Haiku Generation process

Then for each row in the haiku to generate, we sample a grammatical structure. For each POS-tag in that structure we insert a word using either a random stop word, our Markov Chains or WAN/WordNet depending on the POS tag. Then the pluralization, conjugation of verbs and repetition avoidance are applied to the resulting words (Fig. 2). The two-phase steps are described in detail in the following sections.

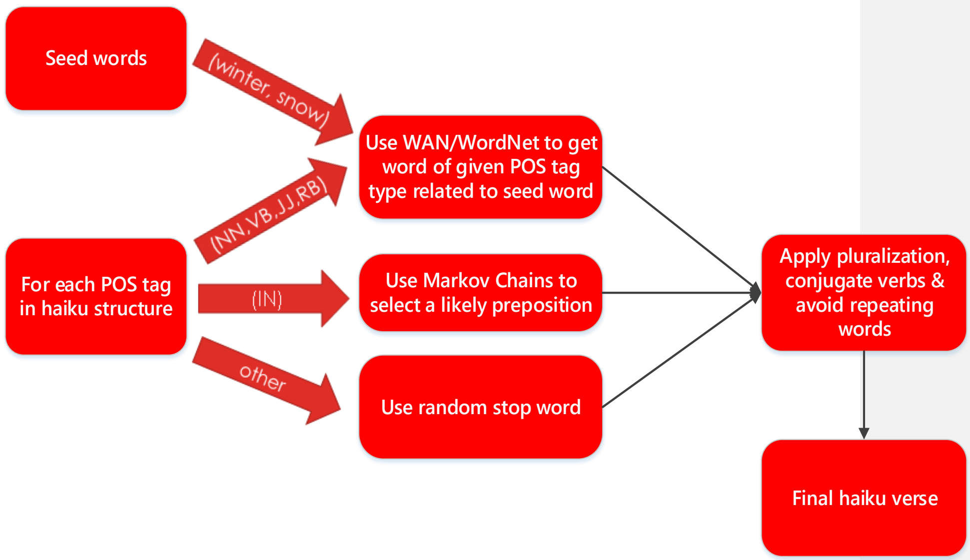

Fig.2 Haiku generation phase

3.1. Compile a Haiku corpus

The first step was to create a Haiku verses corpus that can be used to extract the necessary information in the pre-processing phase of our generator. This step is vital since we needed a large dataset to learn the most common grammatical structures and combinations of words (with their probability distribution, etc.) for haiku. We discovered Haiku from contemporary English authors that are organized by season, and within each season, by mood, events, plants and animals that are traditionally associated with the season. We crawled two websites and scraped valuable datasets from them and converted them to a structured JSON format:

- Jane Reichhold's Haiku dictionary [2]: a list of 4821 haiku, classified by 29 season words.
- A set of essential Japanese season words [3], classified by season/period (e.g: late winter)

3.2. Grammar extraction phase

From our POS-tagged haiku corpus, we extracted common grammatical structures to use for our haiku. A grammatical structure is defined as a sequence of POS-tags that form one line in any of the haiku in our corpus. We store each such sequence along with its frequency in the corpus. This frequency is then used to generate random rows for each new haiku, where the probability of getting a grammatical structure is proportional to the frequency of it in the haiku corpus.
An example output of this phase looks like the following:

```
['DT', 'JJ', 'NNS', '\n', 'VBD', 'NNS', '\n', 'NNS', 'VBD']
```

Or, with the POS tags converted back to their original terms, the produced haiku structure is given below:

```
Line 1: determiner adjective noun
Line 2: verb noun
Line 3: noun verb
```

3.3. Extracting stop words

A stop word is a word that carries little information about the meaning of a poem, but is important for the grammatical structure. Common stop words are 'a', 'an', 'the', 'to' etc. These words belong to specific POS-tags and to generate a stop word we simply randomize a word with the correct POS-tag from our list of stop words. This list was in turn extracted from our POS-tagged haiku corpus.

3.4. Preposition Selection

We handle prepositions as a special case, since using the randomization approach from the previous subsection produced poor results. We used Markov Chains of order one and our haiku corpus to estimate the conditional probability of seeing a certain preposition after a certain word. We then used these conditional probabilities to randomize a preposition given the previous word. As a fallback, the method from the previous subsection is used if the previous word is not in the haiku corpus.

3.5. Content selection phase

Content selection consists of selecting suitable words to give the haiku a sense of meaning. At each POS-tag, we either pick as our "inspiration" the haiku's theme (a seed word like "winter") or the last generated adjective/verb/noun (if any), with equal probabilities. This way we keep a common semantic thread through our haiku, while we avoid looping back on the same concepts. Once an inspiration is picked, our two content selection mechanisms generate a word of the given POS-tag and associated with the given word. This association is performed in different ways in the WAN and WordNet implementations.

3.5.1. WAN implementation

The WAN implementation loads the dataset from [4] into a map from stimulus word to a list of responses. When given a stimulus word and POS tag, it lists all possible responses to the stimulus word, excludes all words with the wrong POS-tag and returns a random word from those that are left.

3.5.2. WordNet implementation

The WordNet implementation uses the synonyms, hyponyms and hypernyms of the given word, and each word found in their corresponding lexical definition, to compile a list of related words. This list is then filtered by POS tag, and used by the haiku generator for next word selection based on POS tag.

3.6. Assembling and post-processing stage

Each time we generate a word, we post-process it to make sure it fits well in the haiku as a whole:

- Nouns are correctly pluralized
- Verbs are correctly conjugated
- We make sure words aren't duplicated in the haiku since looping on the same words is likely to happen when looking for similar terms
- We check that some connections between words make sense (no plural after "the", no noun that begins with a vowel after "a", ...)

This phase is vital as it gives coherence to the haiku and avoids robotic-looking sentences where each term looks like it was forced into the haiku.

3.7. Over-generation and purging step

In this phase, we generate a big number of haiku (100 in our case) and rank them to only pick the top 10-20. This is usually done by using a heuristic function that rates factors like grammatical correctness, themes present in the haiku, etc. Since such a heuristic was out of the scope of our project, we simulated this phase manually by generating one hundred haiku and having each member of the team manually pick the first 20 haiku he/she considered to be grammatically correct and sensible. Then, we intersected these sets to find the best haiku (in our case 5 generated using WAN and 5 generated using WordNet).

3.8. Technical implementation details

The haiku generator was implemented in Python [7]. We utilized the Natural Language Toolkit (nltk) [8] for POS tagging, and access to word synonyms, hyponyms, hypernyms and their definitions (WordNet) [9].
The dataset collection phase was done by scraping HTML content with the *lxml* XML/HTML library.

A selection of some of the best haiku generated using our method are given in table 2 below.

WAN	**WordNet**
Cat life spouses— A hug Through every kindness	The atomic snows — Had solidifications Songbirds formed
An excessive desire— Fornicate wives Under theology exegeses	Over some winter— Metropolis died This life time
Big brontosaurus— Keep At the movie	No study understanding— Worlds Of no mathematics

Table 2 – some of the best haiku created by our generator

4. Experimental results

4.1. Poll setup

We wanted to make our results comparable to the results described in [1], so we created a poll to allow for human test subjects to rate the haiku produced by the generator.

We constructed a poll containing 15 haiku, 10 generated by a computer and 5 written by humans. 50 haiku were generated from the generator using WAN, and 50 others when using WordNet, both using random seed words. 10 haiku were selected by our team as mentioned in section 3.8, 5 from each set. The 5 human-written haiku were randomly picked from our original haiku dataset.

We published this poll online, and asked our readers to rate every haiku, based on grammatical correctness and coherent meaning, from 1 to 5 stars, and asked them to decide whether the haiku was human-written or computer-generated.

Commenté [In6]: We have to justify or explain that part

4.2. Results

Our results were obtained by aggregating the 28 answers from the poll described in 4.1, and are summarized on figures 3, 4 and table 3 below:

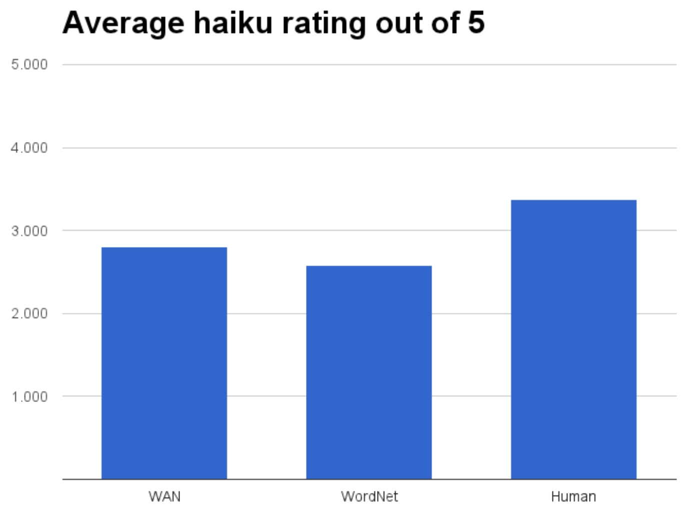

Figure 3. Average rating for each haiku source

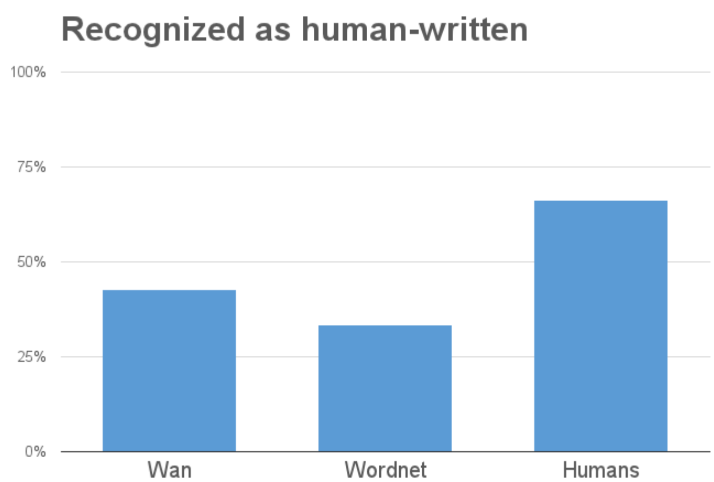

Figure 4. Average percentage of haiku recognized as human-written for each haiku type

Rating(/5)				Human written (%)		
	Avg	Min avg	Max avg	Avg	Min avg	Max avg
WAN	2.80	2.33	3.25	43%	19%	64%
WordNet	2.59	2.30	2.75	35%	30%	54%
Humans	3.38	3.18	3.46	67%	54%	75%

Table 3: Averaged haiku poll results, based on responses from 28 survey participants
Avg: average
Max avg: average of the max scores/ratings for each category
Min avg. average of the min scores/ratings for each category

As expected, we found that human-written haiku have the best rating and human-written recognition (Fig. 3 and 4). However, it can also be observed that haiku generated using both WAN and WordNet have a human-written recognition rate higher than 35% (Fig. 4). These results are better than what we expected.

We can finally observe that WAN (2.8/5 and 43% for human-written recognition) performs slightly better than WordNet (2.57/5 and 35%). Those results confirm the ones described in [1], where WAN is expected to perform better. However, the difference between WAN and WordNet performance is arguably not that pronounced. From the min. and max. rating averages in Table 3, we can also deduce that low-rated haiku from WordNet and WAN score similarly (2.8 and 2.6, respectively), but highly-rated haiku generated using WAN are liked more than those generated using WordNet (with scores of 3.25 and 2.75, respectively).

4.2.1. WAN and WordNet statistical comparison

We can compare the WAN and WordNet results statistically. The ratings from WAN and WordNet (10 haiku with 28 responses) combined comprise 280 observations, 140 from each generator. From these observations we can estimate a mean rating u = 2.694 and standard deviation $\sigma = 1.220$.

Now we introduce the null hypothesis H0: There is no difference in the distribution of ratings between WAN and WordNet.

The mean u1 of the n=140 samples generated by WAN is 2.799. What is the probability of this deviation according to our null hypothesis? We assume that u1 is normally distributed $N(u,(\sigma/\sqrt{n})^2)$ according to the central limit theorem. Then we find that the probability is:

$$P(x > 2.799),\ x \in N(u,(\sigma/\sqrt{n})^2) = 0.1520$$

Commenté [In7]: What does that mean exactly ?

Commenté [JW8R7]: It's the probability that we get a mean score higher than or equal to what we actually got for WAN. Under the assumption that there is no difference between WAN and WordNet. If this prob is really small it means that our premise(WAN and WordNet are equal) is probably wrong.

So we cannot prove at a significance level of 0.9 that there is a difference, but we are not very far off either. We get similar answers for WordNet, but more answers to our poll would have given increased granularity.

5. Summary and Conclusions

The contribution of this paper resides both in the generation process presented above, and in the comparison of the WAN and WordNet data sources.
The results from our poll may not be representative enough to have statistical significance, but preliminary results are indicative that both haiku types (generated using WAN and WordNet) have relatively similar performance, and over 35% of the haiku from each type in the poll passed as created by a human being.
By using Markov chains for the preposition selection, we suggest a new way to improve haiku quality.
As a future improvement, the over-generation and purging step could be automated by designing an efficient heuristic function to remove the manual step (and any bias it causes), thus resulting in haiku generated with a consistent quality and style.

References

[1] Netzer, Yael, et al. "Gaiku: Generating haiku with word association norms." *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*. Association for Computational Linguistics, 2009.

[2] "A Dictionary of Haiku: Classified by Season Words with Traditional and Modern Methods", Jane Reichhold, http://www.ahapoetry.com/aadoh/h_dictionary.htm, Accessed 2014-11-01

[3] "The five hundred essential Japanese season words", Kenkichi Yamamoto, translated by Kris Young Kondo and William J. Higginson, http://www.2hweb.net/haikai/renku/500ESWd.html, Accessed 2014-11-01

[4] Nelson, Douglas L., Cathy L. McEvoy, and Thomas A. Schreiber. "The University of South Florida free association, rhyme, and word fragment norms." *Behavior Research Methods, Instruments, & Computers* 36.3 (2004): 402-407.

[5] Fellbaum, Christiane. *WordNet*. Blackwell Publishing Ltd, 1998.

[6] "Python Software Foundation", https://www.python.org/, Accessed 2014-11-01

[7] "Natural Language Toolkit", http://www.nltk.org/, Accessed 2014-11-01

[8] "WordNet – A Lexical Database for English", http://wordnet.princeton.edu, Accessed 2014-11-01

[9] "The Genuine Haiku Generator." Selendy Communications, 2000. Web. 29 Oct. 2014. http://www.everypoet.com/haiku/explanation.htm, Accessed 2014-11-01

[10] "Haiku-O-Matic." Smalltime Industries, n.d. Web. 29 Oct. 2014. http://www.smalltime.com/haiku.html, Accessed 2014-11-01.

[11] Harris, Jacob. "About Times Haiku." *Times Haiku*. New York Times, 2013. Web. 29 Oct. 2014. http://haiku.nytimes.com/about, Accessed 2014-11-01.

[12] "Interactive Haiku Generator." *Language Is a Virus*. N.p., n.d. Web. 29 Oct. 2014. http://www.languageisavirus.com/interactive-haiku-generator.html, Accessed 2014-11-01.

[13] "JavaScript Object Notation", http://www.json.org/, Accessed 2014-11-01