Insert a cool title here

Alham Fikri Aji
University of Edinburgh
s1447728@sms.ed.ac.uk

Abstract

ABSTRACT

1. Introduction

Poetry is one of natural human method to communicate by encapsulating the message in beautiful and artistic manner. William Wordsworth said that poetry is "the spontaneous overflow of powerful feelings: it takes its origin from emotion recollected in tranquillity" [1]. More non-philosophic and restricted description defined by Manurung, poem is a text that is meaningful, grammatically correct, and poetic [2].

Automatic poetry generation one challenge in artificial intelligence and computational creativity. It is considerably interesting as it works on human creativity, emotional and intelligence domain [3]. The difficulties of this task comes in various factors. One major issue lies in the poem evaluation. Evaluating poems and creative arts tends to be subjective, thus automated poetry generation faces difficulty to obtain objective result [4]. Another challenge in poetry generation is massive requirement of the resources. In order to create poem that satisfies diverse constraints such as phonetic, syntax, grammar or semantics requires wide array of resources [5].

The objective of this work is to explore further about poetry generation. This work tries to answer the challenge in computational creativity on automatically builds poetry as one of human artifact. This work will be focused into one specific genre of poem: Haiku. Although specialized, we are hoping that this work can be generalized in some aspects for broader and more generic poetry generation.

Lastly, poetry generation could be used in various creativity related domain, such as helping tool for song composers, video game company, greeting cards industry, or be used by poet themselves. It also could be used to enhance computer to human interaction system. Personal assistant systems could embed poetry generation to make it capable to talk in poetic manner, thus making it more friendly towards the user.

The papers are organized as follows. The section 1 delivers the motivation and the objective of research in automated poem generation. The section 2 of this paper explains necessary background information related to this topic. This section also covers recent related works. The section 3 lists some useful resources to aid the research. The section 4 states the specification of this project. The section 5 describes detailed information of the proposed methodology. Finally, the section 6 is the last section that discuss about evaluation method that will be used in this research.

2. Background

2.1. Haiku

Haiku is traditional Japanese poetry . Rules of writing Haiku are clear in Japan. However, problem occurs when Haiku is adopted into different language [6]. The issue comes from different linguistic concepts and designs from Japanese to foreign language. Some Japanese language concepts simply does not exists in English. Cultural difference is also one reason of variety rules in foreign Haiku. Therefore, there is no strict rule of writing Haiku in English [7].

Haiku is three-parts short poem consists of 17 syllables. The syllables are further distributed so that the first and the last part have 5 syllables and the second part has 7 syllables. However, converting this rule from Japanese language which has monosyllabic phonetic system into English which has different phonetic system can sometimes be challenging [6]. Thus, 5,7,5 syllables distribution rule is still important in English Haiku, but not completely mandatory [7].

Romaji:

yuku haru ***ya***
tori naki uo no
me wa namida
(Bash, tr. Shirane)

Translation:

spring going–
birds crying and tears
in the eyes of fish

Haiku consists of two different sub-ideas, separated by *kireji*, or "cutting word". In Japanese, *kireji* is used to extend emotional context. It may be used as punctuation. As *kireji* does not exists in English, it is replaced by punctuation mark, such as dash or question mark. In some cases, *kireji* is simply unmarked and understood as implied delay [8]. In example above, *kireji* "ya" is often used to emphasizes the previous word. In English, it is translated with long dash.

There is no specific genre in Haiku. Any subjects can be written in Haiku [7]. Haiku poem often delivers the content physically. This expression can be achieved by choosing words that relate with one of human senses such as visual, hearing, or touch. Moreover, Haiku uses *kigo*, or 'seasonal words' to refer and visualise the poem into one of the seasons in Japan. However, modern Haiku may do not contains any seasonal word. [7]. It is also important to collaborate the five senses perception with correct seasonal theme [9].

2.2. *Automated Poetry Generation*

Before moving into automated poetry generation, we need to discuss about automated text generation in general. Computer generated text is ...

We define automatic poetry text generation simply as automatic text generation with additional poetic constraint. The . Some poetic measurements have been described and will be discussed in next part of this chapter.

Automated poetry generation is not something new. Previous attempts have been done to develop poetry generator.

2.3. *Measuring Poetry*

As mentioned in the first chapter, one major issue of poetry generation is how to objectively evaluate the result. Hence, a method to answer "how good your poetry" should be defined formally. In this part, we some definitions and methods to achieve such goal.

Manurung et al. mentioned that there are three aspects of poem that can be scored. The first one is the phonetics. This information can be evaluated through the poem's phonetic structure such as rhyme, metre, alliteration, etc. The second measurement is linguistic. Word and syntax choices is measured in this point. The last part of evaluation is semantics, where a poem is scored based on its semantics structure relative to the desired semantics[5].

Colton et al. suggest more statistical approach on measuring a poem. Evaluation done by analyzing each word and assign some scores based on several criterion. A flamboyance is defined on word frequency, where a word obtained less score based how frequent it used in the constructed poem. It also computes the relevancy of each word from given baseline article. Lastly, appropriateness is a score defined by its average word's sentiment distance towards a given sentiment level [10].

3. Resources

3.1. *WordNet*

WordNet is huge English lexical database that focuses on relationships between the meaning and the concept of words [11][12]. WordNet groups nouns, verb, adjectives, and adverbs based on their meaning and usage into set called sysnet. Words in a sysnet shares same concept and can be substituted by each other in most cases. WordNet has about 117.000 sysnets covering broad topics[13].

Relationship between sysnets are defined by linking them with conceptual relational attribute. Some sysnet relations examples in the wordNet are:

- Hypernym

  Relates a concept into its superordinate concepts. In verb, it connects the verb into the superordinate events. For example: book → publication, write → create.

- Hyponym

  Simply the inversion of hypernym. It relates the connects subtype concept relations. For example: book → catalogue, textbook, story book, etc.

- Troponym

  Similar with hyponym. Troponym is a verb relationship that captures subordinate event of another event. For example: travel → fly, ride, etc

- Antonym

  Relates the opposite of word senses. For example: happy → infelicitous, travel → stay in place.

- Derivationally Related Form

  Connects lemma with the same root. For example: burn → fire.

- Entailment

  A verb to verb relations where one verb is involved with the other. For example: snore → sleep

WordNet is useful resource for text generation. It can be used to paraphrase generated text. Its relational concept is helpful to build the knowledge base of the system. WordNet can be used to enrich the system's dictionary by replacing some words with other words that shares similar usage and characteristics.

3.2. Standford PoS Tagger

Standford Part-of-Speech (PoS) tagger is an application with purpose to assigns part of speech label verb, noun, adjective, etc. to each input text word [14]. For English language, Stanford PoS tagger uses labels in Penn Treebank tag set. This tag set has a total of 36 different labels with more specified and extended labels such as verb-past, verb-present, adjective-superlative, etc[15].

Input

A frog jumps into the pond

↓

Output

A/DT frog/NN jumps/VBZ into/IN the/DT pond/NN

Figure 1: Example of PoS tagger

Figure 1 illustrates an use example of Standford PoS tagger. An input sentence is tokenized and given a label for each token. The output of PoS tagger will be list of paired token and its speech tag.

3.3. CMU Pronouncing Dictionary

Carnegie Mellon University Pronouncing Dictionary (CMUPD) provides machine-readable pronunciation dictionary for over 133.000 words. It uses 39 set of phonemes based on ARPA-bet symbol set. Each vowel contains lexical stress pattern information, further categorized into three different levels: 0 for no stress, 1 for primary stress, and 2 for secondary stress[16].

Word	Pronunciation
Haiku	HH AY1 - K UW0
Poetry	P OW1 - AH0 - T R IY0
Thesis	TH IY1 - S AH0 S0

Table 1: Examples of Syllabified CMU Pronunciation Dictionary

The stress pattern and pronunciation are invaluable resources that can be used to analyze the beautifulness of a poetry.

Bartlett et al. further improve the CMUPD and add additional syllables information. This syllabified CMUPD provides syllable boundaries that splits between one syllable to another [17]. This information can be used to compute the total syllables of given word which is invaluable resource for fitting the syllable constraints in Haiku. Table 1 shows some examples of words and their corresponding pronunciation data.

4. Project Specification

In this project, we aimed to build automated Haiku generator. This Haiku generator will takes input of one or some topic keywords from user, and randomly generates Haiku that related with the keyword.This keyword is not mandatory, topic will be randomly chosen in case when user does not provide the input.

The generated Haiku will follow English Haiku rules as follow:

1. It consist of three lines with 5,7,5 syllables distribution
2. Using *kireji* is not necessary
3. Using seasonal words is preferable, but not required.
4. Does not consider about rhymes

However, as there is no strict rule of writing English Haiku, we allow user to customize the output Haiku structure. Therefore, we put the rules above as default rule and user may change it preferences as desired.

This project will be implemented in Java, thus we expect that it should run in most standard computers. Moreover, Java is operating system independent, hence the application will run in most operating systems. A Java applet of the project will be developed and further deployed on web platform.

5. Methodology

The brief concept of implementing Haiku generator is by learning the grammatical pattern and rules of existing Haiku poems and build a new grammar skeleton based on it. The generator will select words to fill-in the grammar skeleton based on statistical approach. Other resources such as CMUPD and WordNet will be supporting the decision making of words selection.

5.1. PoS Tag Extraction

In this phase we apply PoS tagger algorithm for every collected Haiku poems. Therefore, in this phase we aim to acquire list of grammar tag pattern. We are only interested in the grammar tag, therefore we can remove the Haiku after tagging. One simple illustration of this process is provided in table 2:

Haiku	Extracted PoS Tag Information
Three strokes of lightning	CD NNS IN NN
One hit mountain frightening	CD NN NN JJ
Dark clouds thunder loud	JJ NNS NN RB

Table 2: Example of extracted PoS tag information of the Haiku

The reason of choosing this method over using grammar tree or context free grammar is because sometimes poetry does not follow standard grammar rules. By using formal grammar rule, we might achieve less poetic result. By learning from existing Haiku, we hope that we are able to capture the grammatical pattern used in Haiku.

5.2. Creating Haiku: Grammar Skeleton

Grammar skeleton is sequence of tags that will be used to create a Haiku. The generated Haiku should follow the grammar tag defined by the skeleton. In this phase, grammar skeleton is created by combining several tag patterns in previous part.

Pattern 1	Pattern 2	Pattern 3
CD NNS IN NN	DT NNS IN NNP	**JJ JJ NN**
CD NN NN JJ	JJ NN CC JJ NNS	VBG IN DT NN IN NN
JJ NNS NN RB	**VBG DT NN**	NN IN

↓

Generated Skeleton
JJ JJ NN
JJ NNS NN RB
VBG DT NN

Illustration of creating a grammar skeleton based on tag data is shown above. Some basis grammar patterns are randomly chosen as parents. The grammar skeleton is generated by applying crossover from the parents. In this example, each parent inherits one line of grammar pattern. In the implementation, the rule is not strictly forced the generation made by three parents. Therefore more or less parents is possible.

5.3. Creating Haiku: Word Filling

Haiku will be constructed with previously defined grammar skeleton as its basis. For each speech tag, we look up word with exactly same label and fill in the slot with the chosen word. Our word resource comes from various corpus. Although some corpus such as Brown corpus has already annotated with speech tagging, we need to apply identical PoS tagger algorithm that used in constructing the skeleton for speech label consistency. To enrich our dictionary, WordNet will be used to paraphrase and replace some words in corpus with different but related terms.

In some situation user will provide the system with some keywords. In this case, the keywords will be tagged as well before further processing. Our tagged keyword will be placed in the skeleton first, before applying general word filling. Again, WordNet can be used to transform user keywords into another similar term.

Word filling is not randomly choose any words with required speech tag. It follows haiku rule constraints and meaning constraints. The word must be chosen in particular so that it satisfy the syllables rule. In case of strict use of *kiregi* and/or *kigo*, those words must exists in generated haiku. Meaning constrains is on how to chose correlated words and have actual meaning. We want to make the resource usage remains minimum, therefore we choose a word if it is statistically correlated with one of some previously placed words.

5.4. Creating Haiku: Scoring

In this part, generated haiku will be scored to determine its quality. We use the restricted definition of poetry by Manurung where a poem should be meaningful, grammatically correct, and poetic. As the haiku is generated based on previously defined grammar, we can remove the second factor as a score. Therefore, our scoring consists of two parts: Meaningfulness and beautifulness.

Meaningfulness score M is calculated by statistical approach. The haiku will be scored based on statistical relationship of each word, and word to word relationship. Beautifulness score B is defined by some phonetic features that could be extracted from the haiku such as rhymes, stress pattern, etc.

$$S = w_m M + w_b B$$

The score will be a weighted sum of both components, as defined in formula above. Therefore, higher w_m leads to more 'meaningful' Haiku and higher w_b leads to more 'poetic' Haiku. The composition of the weights is another important research part. The challenge is how to determine the weight so that the generate haiku is not too statistically meaningful so that it only produces common sentence, but not too 'creative' so that it does not make sense at all.

This score can be used as quality threshold to make sure our generated haiku result good. The system will reject any generated haiku lower than some defined threshold and keep generating until its score requirement is satisfied.

6. Evaluation

We propose a Turing test to evaluate our system and the research outcome in general. In this test, we create several automated generated Haiku and several human created Haiku. Those Haiku will be mixed into a single set. Some respondents will then decide which one is computer generated and which one is created by real human. This test will observe how good our Haiku in terms of its similarity to real human creation. Another evaluation can be achieved by asking the respondents some question and score on various aspects of the Haiku, such as "Do you understand this Haiku?" or "How beautiful this Haiku".

In engineering aspect, we could perform some test on how fast the system build the Haiku. Some further observation regarding the relation between score weighting and threshold towards the execution time can be performed to find out the best configuration so that the resulting Haiku is not rubbish, yet still takes reasonable amount of computational time. Another aspect that can be observed is the correlation between the resource used towards the generated Haiku. In this evaluation, we want to find out what is the best text resources for Haiku generation and automated poetry in general.

References

[1] William Wordsworth. Preface to lyrical ballads (1802). *Romantic Prose and Poetry*, 1990.

[2] Hisar Manurung. An evolutionary algorithm approach to poetry generation. 2004.

[3] Simon Colton, Geraint A Wiggins, et al. Computational creativity: the final frontier? In *ECAI*, pages 21–26, 2012.

[4] Kim Binsted. Machine humour: An implemented model of puns. 1996.

[5] Hisar Manurung, Graeme Ritchie, and Henry Thompson. Towards a computational model of poetry generation. Technical report, The University of Edinburgh, 2000.

[6] Kei Grieg Toyomasu. Haiku for people. URL `http://www.toyomasu.com/haiku/`.

[7] Haiku in english. URL `https://www.ryukoku.ac.jp/haiku/haiku_en.html`.

[8] Ashley Capes. An introduction to haiku - form and structure: Kireji and kigo. URL `http://ashleycapes.com/2013/08/26/an-introduction-to-haiku-form-structure-kireji-kigo/`.

[9] Michael Dylan Welch. Becoming a haiku poet. URL `http://www.haikuworld.org/begin/mdwelch.apr2003.html`.

[10] Simon Colton, Jacob Goodwin, and Tony Veale. Full face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, pages 95–102, 2012.

[11] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[12] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

[13] Princeton University. About wordnet, 2010. URL `http://wordnet.princeton.edu`.

[14] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[15] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[16] Carnegie Mellon University. The cmu pronouncing dictionary. URL `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`.

[17] Susan Bartlett, Grzegorz Kondrak, and Colin Cherry. On the syllabification of phonemes. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 308–316. Association for Computational Linguistics, 2009.