University of Nottingham
UK | CHINA | MALAYSIA

Implementation of better Synonyms and Hyponyms for Poetic Writing.

Submitted by:

18025577

Abstract

Blackboard systems used for generating poems are inspired by how experts successfully alter the blackboard given a set of inputs. As each expert's contribution revolves around a specific task in the poetry generation process, which is then presented on the blackboard for further alteration by the experts until a problem is solved. This study covers the design and implementation of additional experts into a blackboard system, attempting to produce more poetic speeches.

We have three distinct experiments for the generating process: Based on a part of the text from the corpora, the system (1) replaces a word with one of it's synonyms according to its part of the speech (i.e. noun with noun, adjective with adjective, etc.) (2) replaces words with synonyms that have the same sound as the word before, and (3) replaces a word with one of its hyponyms. Finally, we evaluate the fluency of the generation poem.

Introduction

Poetry generation is a challenging task because both linguistic and literary aspects need to be taken into account. However, for poetic writing the text does not follow specific characteristics [1], as there are rules that define a text as poetic that need to be broken down when it comes to the production of natural language. [2]. Moreover, poems include a repeated occurrence of interdependent verbal spectacles where rhyme, existence of metaphoric language, and meter are features that play an important role.

This topic did not begin to receive notice from the Computer Science society until the turn of the millennium, mostly from scientists in the field of Artificial Intelligence (AI) and, especially, Computational Imagination [3]. Since [1] and [2] contributions, there has been a significant growth in the development of poetry generating systems as nowadays intelligent systems are developed to be not only limited to rewriting text for poems. These systems contain some kind of knowledge that deals with several areas of language in order to produce aesthetically pleasing text with a sense of creativity. Poetry generators are useful for educational purposes, such as coming up with scientific poems that boosts the student's level of recalling information.

Related work regarding poetry-generating systems

There are several systems that employ human-generated text corpora as source material creating poems. Particularly, [4] and [5] both revolve around a small piece of text from poetic human written texts. [6] gathers information from blogs and search engines; the Google n-grams corpus and Project Gutenberg and were used by [7]. All of these methods count on user-supplied strophes to initiate a search for fundamental materials and kick-start the poem production procedure.

Electronic Text Composition (ETC) poems generating engine [8] is a generator that utilizes a huge corpus of ordinary language to generate meaningful poetry. Its knowledge base is made up of the British National

Corpus' 85 million parsed words, which have been converted to a dictionary of 560,000 terms and 49,000,000 lists of phrase connections. ETC creates specific poetry patterns, and its sheer size promotes unexpected, grammatically valid results. A dozen of the sonnets have been published anonymously.

The WASP system, which deploys a cultural meta-level generating method in which a writing is well-thought-out and advanced by a group of different expert sub-systems, they act like a cooperating group of readers, critics, editors, and writers that could consider performing concept formation in terms of features of computational poems at a more valid level than just generating examples. [5] In a similar fashion to [4] and [9], the iterative assessment findings are not shown together with the last output, and the esthetics that the system evaluates are heavily influenced by the collected resources used to train the text generator.

To my knowledge, no poetry generating systems offer an appealing context in which to evaluate the poems they generate. Furthermore, none of the current systems present their poetry any context. Generally, the environment in which the poems can be considered is either intentionally obscured allowing objective evaluation, as in Turing-style experiments, or given by the computer scientist in the form of a scientific paper, preface to a compendium, or a web page.

A few websites that produce poetry in a simple manner and then ask the viewer to read them. Poem Generator website (www.poem-generator.org.uk/sonnet/), for example, claims that a large piece of poetry baffles its readers: It might sound lovely, but it makes you wonder “what the heck was that meant to mean?” before extolling the virtues of spontaneously producing strange sounding poems. This overlooks the fact that poets employ their brains when writing rhymes that may require revision with the purpose of better communicating a message, emotion, or style.

Background and Implementation

A blackboard system is a learning system that consists of a collection of information (the blackboard) and a group of agents that may read and write to the blackboard. One of the most essential characteristics of such a system is its ability to easily grow by adding new agents. Because the agents do not interact directly, there is no need for a complicated coordination mechanism as they simply read from and write to the board in any sequence, and the information on the board becomes more relevant to the job if the system is effectively built. When the initial problem is printed on the blackboard, the experts begin to contribute at random while keeping an eye on the blackboard. Until the challenge is solved, the experts continue to add their contributions to the blackboard.

The blackboard model enables the implementation of a wide range of modules. Each expert is regarded as a black box, with all internal implementation details hidden. The interface between the module and the blackboard is the only thing that needs to be standardized. The data on the blackboard can be structured at various degrees of abstraction and in various forms (as long as the chosen modules can interface with it).

Because a blackboard system's solution is constructed incrementally, it's good for situations that involve a lot of steps or a lot of different sources of knowledge. The blackboard architecture also makes it easier to explore through a vast number of potential solutions, including multiple pathways of reasoning based on incomplete or inaccurate information. The search approach can be tailored to different stages of issue solving, allowing it to work at several levels of abstraction and investigate multiple lines of reasoning.

Experts in word generation have lexical knowledge: they develop words related to the issue and add them to the pool of ideas sections, as well as a repairing function that puts a and a back in their proper positions (i.e. an apple instead of a apple). Based on the WordNet lexical repository, the WordNet Experts produce synonyms, hypernyms, and antonyms for nouns and adjectives.

The NLTK Python package is utilized by the majority of language-processing tools. The use of the widely recommended useful WordNet lexical database, which contains a hierarchical structure of syntactically related nouns, verbs, adjectives, and adverbs grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept, for defining word associations and finding related expressions.

Similar part of speech synonyms:

1. Reads and tags each word with its part of the speech as shown below:
   - Once there was a dog and a cat
   - ('Once', 'RB'), ('there', 'EX'), ('was', 'VBD'), ('a', 'DT'), ('dog', 'NN'), (‘and’, 'CC'), ('a', 'DT'), ('cat', 'NN')
2. Selects a word at random and represents its synonyms.
3. Checks the synonyms pos_tag and appends similar part-of-speech synonyms into a dictionary to select o ne at random.
4. If the word doesn’t have similar part of speech synonyms, then the word selected at first is returned in it s original place in the sentence otherwise the similar part of speech synonym is appended instead.

Replace words with their hyponyms:

1. Reads and selects a word from the user input text.
2. Checks if it is a stop_word as stop words lack hyponyms.
3. Appends the hyponyms into a dictionary and replaces the word selected by a random hyponym.
4. Returns the selected hyponym to the original sentence.

Replace words with synonyms that have the sound as the word before:

1. Selects a word at random to be replaced with a synonym that sounds like the word before.
2. This is done by getting the syllables of the word before and observing the ending sound.
3. Then the word-to-be-replaced synonyms that rhyme with the word before are appended to a dictionary, by getting the intersection of both dictionaries the synonyms of the word-to-be-replaced and the syllables of the word-before.

4. Returning the replaced word to the location of the initially selected word.

The system's algorithm works as shown below.

Algorithm: BlackBoard Poetry Generation Algorithm

```
Data: Input speech text
Result: Poetic Text
Select expert randomly
Experts alter the text accordingly
IF user is happy with the blackboard output:
  Exit the system
Else: Continue generation process
```

Results and Analysis

An innovative system ought to be able to produce both innovative and meaningful outputs. The poems generated for the same input parameters should not necessarily be identical, but they should have a link to those parameters at the same time. If an aspiring artist produces something that resembles that of a previous creator, he is considered as useless. This makes developing a trial set-up for human being assessment of computer created poems that is not theoretically incorrect from the beginning is extremely challenging. If the outcomes mirrors past effort by human writers, critics who assess them as human output will give them a lower score.

The focus is around two crucial elements of a poem to analyze the literary characteristics of the blackboard system outputs: the sum of syllables per line and the existence of rhymes. However, other aspects such as theme, language, structure, context, sound, and rhythm, which are computationally difficult to evaluate, need be addressed in order to completely analyze a poem.

The syllables per lines are compared against the threshold of a sonnet, which includes 10 syllables per line, any possible value of syllables obtained per line that is abnormal to the threshold is negatively marked accordingly. Moreover, analyzing the rhythms per line is accounted to the fluency per sentence, asking the question “Does it sound natural?” recognizes the acoustic value of the generated poem.

Poem Generation Trial	**Line 1**		**Line 2**	
	Number of Syllables	Is it naturally spoken?	Number of Syllables	Is it naturally spoken?
1	20	No	17	Yes
2	15	No	15	Yes
3	18	Yes	8	Yes
4	18	No	17	No
5	11	No	13	Yes
6	16	Yes	12	Yes
7	16	No	13	Yes
8	18	No	12	Yes
9	9	Yes	13	No
10	13	Yes	11	Yes
11	21	No	16	Yes
12	12	No	10	Yes
13	16	Yes	15	Yes
14	13	Yes	9	Yes
15	18	Yes	9	Yes
Mean	15	7/15 Yes, 8/15 No	12	13/15 Yes, 2/15 No

Table 1. Experimental results of agents interacting with the Blackboard.

The average number of syllables for sentence two is considered less abnormal compared to the one in the first sentence, alongside more naturally spoken sentences that are more likely to be used in day-to-day spoken English conservations. Some studies use a Turing test comparing human-like poems with computer generated poems. The Turing test for poetry, according to Alison Pease and Simon Colton [10], is not a trustworthy measure of its inventiveness. They point out that current versions of such tests only measure human-style originality, and that the evaluation is prone to subjective biases, making it inappropriate for evaluating a computer system's creativity. In addition, 4 out of 15 poems contained syllables that were close enough to the threshold, and only one of them is considered speech fluent when considering the first line of the speech. For line 2, as shown in Table 1, the majority of the generated lines did not lose the fluency variable. One can argue that line 1 contained more verbs than line 2, which influenced the alteration process.

Conclusion

In this experiment we implement a blackboard system for poetry generation that focuses on the alteration of words to produce more meaningful sentences. In the results section the generated poems' syllables are compared with the number of syllables a sonnet should have per line. In the future, implementation of other experts will allow further alteration of the blackboard system by adding experts that makes sure both sentences carry the same number of words and replaces longer phrases with shorter meaningful ones that reduce the sonnet's syllables up-to 10 syllables.

References:

[1] P. Gervás, "WASP: Evaluation of Different Strategies for the Automatic Generation of Spanish Verse." Accessed: Jun. 17, 2021. [Online].

[2] H. M. Manurung, "An evolutionary algorithm approach to poetry generation," 2003. Accessed: Jun. 17, 2021. [Online].

[3] S. Colton and G. A. Wiggins, "Computational Creativity: The Final Frontier?," 2012, doi: 10.3233/978-1-61499-098-7-21.

[4] E. Greene, T. Bodrumlu, and K. Knight, "Automatic Analysis of Rhythmic Poetry with Applications to Generation and Translation," Association for Computational Linguistics, 2010. Accessed: Jun. 17, 2021. [Online].

[5] P. Gervás, "Engineering Linguistic Creativity: Bird Flight and Jet Planes." Accessed: Jun. 17, 2021. [Online].

[6] "(PDF) Automatic Haiku generation using vsm." https://www.researchgate.net/publication/229028573_Automatic_Haiku_generation_using_vsm (accessed Jun. 17, 2021).

[7] Y. Netzer, D. Gabay, Y. Goldberg, and M. Elhadad, "Gaiku : Generating Haiku with Word Associations Norms." Accessed: Jun. 17, 2021. [Online].

[8] J. Carpenter, "Electronic Text Composition Project." Accessed: Jun. 17, 2021. [Online]. Available: www.aliencentral.com.

[9] "Poetry Generation in COLIBRI | Proceedings of the 6th European Conference on Advances in Case-Based Reasoning." https://dl.acm.org/doi/10.5555/646180.758710 (accessed Jun. 17, 2021).

[10] S. Colton, J. Charnley, and A. Pease, "Computational Creativity Theory: The FACE and IDEA Descriptive Models." Accessed: Jun. 17, 2021. [Online].