An Artificial Intelligence Approach to Poetry Generation

Shashank Rao
Guide: Prof. Amitabha Mukerjee

Abstract

Poetry is a unique artifact of Human Natural Language Production whose creation requires mastery over linguistic knowledge and creativity. Thus, generating poetry by a computer is certainly a testament to the progress of Artificial Intelligence. This report describes the past work in this area and also presents a working algorithm which generates poetry using semantic similarity based on WordNet[1].

1 Introduction

Poetry is an art form that dates back to time immemorial. Since the dawn of time, man has been composing poetry and even to this day we still are appreciating works from long ago. So now the obvious question is how long to go before machine can compete along with man at attempts at some astounding poetry? The answer, sadly, is that it would certainly take some time for it to be reasonably as able as man. But the good news is that, its progress is rapid.

Generation of Poetry artificially falls under the sub category of Natural Language Generation. However, it goes slightly beyond the traditional NLG problem. This is because, here, not only we have a creative component to deal with but we also have unity of form and content and a lack of clear communicative goal which makes it all the more harder to program.

2 Methodology

The programming language, Python[2] has been used to do the entire coding owing to its extra-ordinary flexibility and portability. First a wordlist has been used as input which is having basically two fields, namely, word and class. Here word is the word itself while the class is indicating which set it belongs to: nouns, verbs, adjectives, prepositions, etc. This wordlist has then been converted into a dictionary using pickle[3].

From the pickled dictionary, each of the words have been inducted into the data structures of our choice, which here are lists. After a lot of experimenting
lists came out to be the best, as they are very easy to iterate over and the speed at which the results are generated are much faster than using other data structures. Thus each class such as noun, verbs, etc have their own list structure which basically consist of the entire list of words which have been given as input through the wordlist. Now, for each time we want to get any word, functions have been defined which from each list generate a random word of the respective class. This has been done using the random module of python. To have grammatically correct sentences template based grammar has been used in which each line is a randomly generated grammatically correct sentence. The randomness is basically to add to the unpredictability of the program.

The semantic similarity is achieved using WordNet. WordNet is a Semantic Lexicon for the English Language. It groups English words into sets of synonyms, provides short, general definitions, and records the various semantic relations between these synonym sets. The way that it helps our purpose is that it supports automatic text analysis and Artificial Intelligence applications.

The similarity algorithm used is Wu-Palmer Similarity[4]. This algorithm returns a score denoting how similar two word senses are, based on the depth of the two senses in the taxonomy and that of their Least Common Subsumer (most specific ancestor node). This has been implemented in the Natural Language ToolKit[5] (NLTK) module.

Thus the program based on an input title keyword(noun) generated semantically similar statements by running the aforementioned similarity algorithm on the nouns in the poem.

3 Results

3.1 Without Semantic Similarity

The results have been divided into two subsections. To see the difference that the implementation of the semantic similarity makes first the results a very primitive base implementation of the algorithm will be shown. In the following subsection the actual present implementation results will be shown. Thus a qualitative can be made:

- Artificially Intelligent Poetry Generator
- straightway pithy at thy reredos
  his crepe wants to cry their dressy glue
  with our turbojet
  at our stadium
- Artificially Intelligent Poetry Generator
- oft urban – or, rather, your conservation
  one brand should puff my explicit biosphere
- his thrush
into your lump
-
Artificially Intelligent Poetry Generator
-
there Latin for thy gourmet
my paddock should deprave their negligent talon
as per their pyridine
- for example, their raster
-
Artificially Intelligent Poetry Generator
-
hereinbelow detonable during her handbag
our twentieth will plunge our snappish rosemary
like thy thrush
into their plateau

3.2 With Semantic Similarity
This subsection, as stated above shows the results of the algorithm with the semantic similarity algorithm implemented. It can be noted that this algorithm prompts for a user input.

Artificially Intelligent Poetry Generator
-
Give keyword(noun):butterfly
-
Accepted, Thank you!
-
fully exotic just maybe, our butterfly
her firefly might install my sensory butterfly
as per your larva
wherefore, your beast shall walk
-
Artificially Intelligent Poetry Generator
-
Give keyword(noun):city
-
Accepted, Thank you!
-
shyly other per his territory
his prefecture might conclude our federal state
for your state
hither, my city could exhibit
-
Artificially Intelligent Poetry Generator
-
Give keyword(noun):parent
-
Accepted, Thank you!
-
though monogamous as their parent
her sibling might integrate their imperceivable guardian
at their father
truly, her Aries might dress
-
Artificially Intelligent Poetry Generator
-
Give keyword(noun):citizen
-
Accepted, Thank you!
-
maybe transposable for thy prodigy
her friend might surcharge one felonious airman
like one prophet
upwind, his celebrant could advise

4 Conclusion

Thus we can see that our algorithm generates poetry which atleast once in
a while produces poetry which is more meaningful than random collection of
words. Also it can be seen that the implementation of the Semantic Similarity
Algorithm certainly produces more meaningful poems.

5 Future Work

A major problem that I have encountered is the lack of a very good word list.
I can surely attest to the fact that given a better input of a wordlist the poems
that will be produced will be much more tighter in meaning. I plan to get the
entire WordNet word list or some other equivalent and pickle it to form the
dictionary giving a very rich collection of words.

Also, as of now the grammar is a randomized template based. I plan to move
to a PCFG for a much stronger and better form.

With these two improvements the poems that will be generated will be having
better semantic similarity across them and stronger form.

6 Acknowledgement

I wish to express my sincerest gratitude to Prof. Amitabha Mukerjee for making
me believe in myself and giving me this opportunity to work on this project and
learn and develop my skills in the field of Artificial Intelligence. I am also
thankful to him again for letting me choose a project which involves Poetry which has long since been one of my passions.

7 Bibliography

[5] NLTK, nltk.org