Randomness vs. Restriction in Markov Models of Natural Language: An Examination of haiku-text generation based off the works of Richard Wright

Navraj N. Narula
Computer Science Department
Tufts University – Medford, MA USA
navrajnarula@gmail.com / navraj.narula@tufts.edu

Abstract

The notion behind text generation is one that is seemingly playful, yet its role in natural language processing surpasses this thought. Instead it paves the way for content production that is not only cohesive, but also satisfies a set of goals. Mine were to examine whether or not generating text at random versus restricting generation by means of grammatical rules would impact the overall formulation of haikus based off the works of Richard Wright. Upon examination of the output, variation is clearly detected and success in imitating natural language was more so closely related with restriction. As expected, though, the human mind still takes its throne as the ultimate "text generator," according to the collective thoughts of 74 survey participants.

Introduction

Albert Einstein once stated: "Computers are incredibly fast, accurate, and stupid; humans are incredibly slow, inaccurate, and brilliant; together they are powerful beyond imagination." Natural language processing, a subset within the artificial intelligence field of computer science, is arguably the cherry-on-top that cannot be accomplished with one attribute minus the other.

My project involves the close examination of natural language itself, particularly in association with the Markov model. A Markov model utilizes a Markov chain to create a "statistical model" of a piece of text [1]. This notion implies that transitions from a given state are random and only dependent on the current state itself. In devising a Markov text generator, this was my premonition: the next word in my sequence would be determined based on the current word.

The initial algorithm that I devised was based on randomness. I then chose to restrict arbitration in informing word sequences by introducing tags in efforts of insinuating natural language as it would be spoken or written based on grammatical rules. The text that I have chosen to base my model off of are the haikus of Richard Wright, an African-American poet of the mid-20th century whom I encountered in a comparative literature class at Boston University. Not only is his content engaging in the manner in which he depicts man to nature, but the strict 17 syllable structure of the haiku poses the generation of it as a challenge—which is essentially my motivation for attempting the task of Markov generation as it relates to natural language processing.

Claude E. Shannon, also known as the father of "information theory," spoke of a communication system [2] similar to the one I am attempting to imitate in the form of a generator: a system that uses an information source and algorithm to create a sound production.

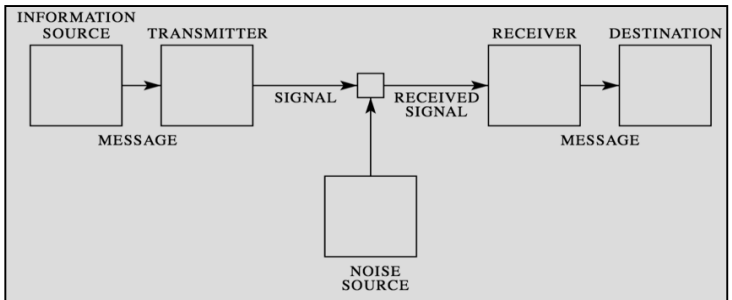


Fig. 1 — Schematic diagram of a general communication system.

Its applications, of course, extend further and is put to use in order to recognize speech, retrieve information, understand data, and filter out what is needed to eventually solve problems.

Datasets

The dataset that I am using to inform my Markov text generator was retrieved from Terebess Asia Online (TAO) [3]. The site itself contains 130 haikus written by Wright. Each are three lines long with the first line containing five syllables, the second line containing seven syllables, and the third line containing five

syllables. Below is a haiku written by Wright:

Standing patiently,
The horse grants the snowflakes
A home on his back.

I manually obtained the data and saved the haikus in a .txt file. This makes up the first instance of my dataset, which I used in my baseline algorithm.

The second instance of my dataset involves labels in the form of grammar tags. I used the NLTK library in Python to map the universal tagger [4] to the standard tagger. Below is the same haiku with tags:

(u'Standing', u'VERB')(u'patiently', u'ADV')(u',', u'.')
(u'The', u'DET')(u'horse', u'NOUN')(u'grants',
u'VERB')(u'the', u'DET')(u'snowflakes', u'NOUN')
(u'A', u'DET')(u'home', u'NOUN')(u'on', u'ADP')(u'his',
u'PRON')(u'back', u'NOUN')(u'.', u'.')

This is the dataset that I used to inform a more restricted algorithm that works to generate text based on grammatical rules rather than pure randomness.

Both instances of the dataset serve as my training set. The test set will be the generated haikus themselves.

Experiments and Algorithms

My baseline algorithm in implementing an initial Markov text generator involved the creation of dictionaries and generating text based on knowledge contained in those dictionaries. Upon examining a piece of text, the dictionary will be populated with key-value pairs in which each key is a word encountered in the text file and each value associated with the key is a list of words that follow that key word within the text file. If a particular word is followed by another word multiple times, it will also be included in that list as many times as it appears to follow the word.

Though perhaps not as relevant to poetical forms such as the haiku, words that served to end sentences (i.e. words that included a ".", "!", or "?" as their final character) were not included as keys. Furthermore, words that started sentences (i.e. lines of each haiku) were listed as values under the start key which I indicated by a dollar sign, "$." Values associated with the "$" key in the aforementioned haiku would be:

"$": ["Standing," "The," "A"]

No third party tools were used to generate the haiku other than the random library made available by Python. The first word is chosen arbitrarily from the list whose key is "$." The following word is then chosen randomly as well from the list of words that may be associated with the first word. The process will once again repeat itself when the current word includes an ending character.

Below are five instances of haikus that I generated using this algorithm.

Like a dead body.
Crying and so silent train.
As the cherry tree

My cold lake in my name away.
The sudden thunder
Startles a sunlit branch

In this autumn morning
Leaving the dew-wet grass
A pretty princess

High above the steeple.
My shadow Hovers in the roses.
A tolling church

I took it is taken By whirling snowflakes.
In the lake.
In an

Not all haikus are of 17 syllables. Though they are each between 10 and 15 words, mimicking the length of haikus.

My latter algorithm also utilized the random library; however, it served to restrict randomness by generating text based on tag patterns rather than word patterns. Using the universal tagger, ten total tags were used to label each word. These tags included:

NOUN → nouns
VERB → verbs
PRON → pronouns
DET → determiners
ADJ → adjectives
ADV → adverbs
PRT → particles
ADP → prepositions and postpositions
CONJ → conjunctions
NUM → numerals

Though not as specific as the standard tagger, universal tags permit for a "more reasonable" comparison of accuracy [4], which speaks to my aim in generating haikus to a more so accurate degree—or at least above the baseline. As opposed to my former algorithm which used the untagged dataset, this algorithm considers word-tag pairs. The NLTK library was crucial to accomplishing the task of mapping the universal tag to the standard tag for each word in each haiku. Below are two versions of the same of haiku. The first is tagged standardly while the second is tagged universally.

(u'A', 'DT')(u'nude', 'JJ')(u'fat', 'JJ')(u'woman', 'NN')
(u'Stands', 'VBZ')(u'over', 'RP')(u'a', 'DT')(u'kitchen', 'NN')(u'stove', 'NN')(u',', ',')
(u'Tasting', 'VBG')(u'applesauce', 'NN')(u'.', '.')

(u'A', u'DET')(u'nude', u'ADJ')(u'fat', u'ADJ')(u'woman', u'NOUN')
(u'Stands', u'VERB')(u'over', u'PRT')(u'a', u'DET')(u'kitchen', u'NOUN')(u'stove', u'NOUN')(u',', u'.')
(u'Tasting', u'VERB')(u'applesauce', u'NOUN')(u'.', u'.')

The Counter feature from the collections library was then used to assist me in counting the number of times a previous tag was followed by another tag. The most common patterns are indicated below.

```
In [47]: sortCommonCounts(commonCounts)
Out[47]:
[(('DET', 'NOUN'), 70821),
 (('NOUN', '.'), 69660),
 (('NOUN', 'NOUN'), 65790),
 (('ADJ', 'NOUN'), 42570),
 (('ADP', 'DET'), 42183),
 (('NOUN', 'ADP'), 41796),
 (('.', 'DET'), 35604),
 (('DET', 'ADJ'), 31347),
 (('NOUN', 'VERB'), 28638),
 (('ADP', 'NOUN'), 17415),
 (('.', 'ADP'), 15093),
 (('VERB', 'DET'), 14706),
 (('PRON', 'VERB'), 13932),
 (('VERB', 'VERB'), 13158),
 (('PRON', 'NOUN'), 12384),
```

As indicated by the picture above, a determiner is most often followed by a noun in Wright's haikus.

The generated haikus followed a similar pattern to the sorted list of tag patterns displayed above. Aside from the start of each line—which could be any random word associated with any random tag—the words that followed each other were based on the tag of the previous word. For instance, if the preceding word was an adjective, I would look to see what the next most common tag followed by an adjective would be by reordering the above counts alphabetically.

```
('ADJ', '.'),
('ADJ', 'ADJ'),
('ADJ', 'ADP'),
('ADJ', 'ADV'),
('ADJ', 'CONJ'),
('ADJ', 'DET'),
('ADJ', 'NOUN'),
('ADJ', 'PRON'),
('ADJ', 'VERB'),
```

Disregarding punctuation characters, the next word following the adjective will be an adjective or a noun. This makes sense grammatically to the human mind as well.

Below are five instances of haikus that I generated using this thought process.

To snowflakes tree candle
The scarecrow was from cock
A light creeping holding

To autumn princess arriving
A trunk back of right
A light sickbed leap

Each burning sill window
To pauses fails beneath night
The horn town stumbles

The spring flowers limping
Out barbershop doll in vanishing
The sparrow sky look

The tulip spider returns
To leg blazing upon face
Rouse hair circus spanks

The challenging aspect in generating haikus were determining the correct number of syllables in total per haiku and per line. Though the above poems do not all obey the general 5-7-5 rule, they again resemble the original haikus themselves. The Carnegie Mellon

University (CMU) Pronunciation dictionary makes it possible to count syllables in words and sentences [5]; however, due to its predefined nature, there are certain words that may be encountered in certain texts that are not taken into consideration within the dictionary. During iteration over these words, key errors may be encountered if the CMU dictionary is being utilized. This occurred in my case while analyzing words in Wright's haikus.

Both the baseline and this latter algorithm are constructed in a way that it could potentially scale to large amounts of data. If textual patterns as it relates to poetry were not of concern, its accuracy in terms of sentence construction may also increase.

Evaluation

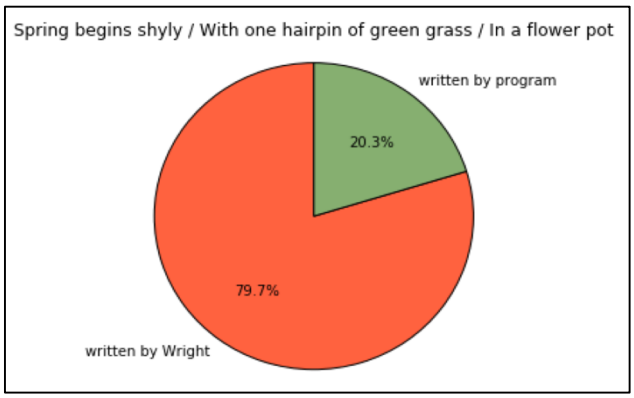
To allow for an objective evaluation of my test set (i.e. the generated haikus from both algorithms), I constructed a survey [6] and encouraged the public to participate in it over social media avenues such as Facebook, Twitter, and Piazza.

The survey asked participants to identify both their age and their gender. Following that were ten questions that asked them to determine whether or not an indicated haiku was written by Wright or written by a program.

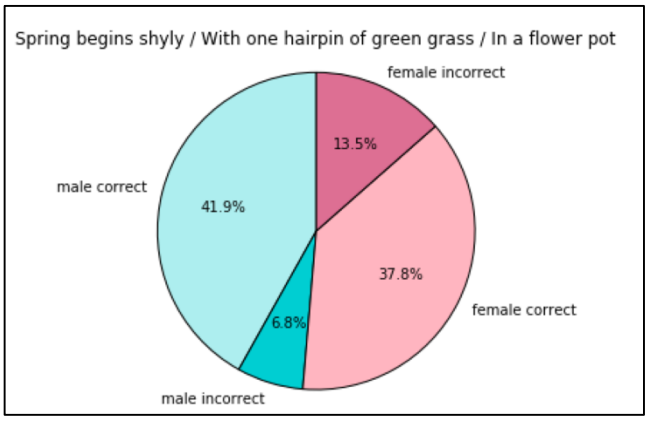
A total of 74 people participated in the survey. 48.6 percent of people identified as male and 51.4 percent identified as female. Ages ranged from 13 to 38 years old. The overall survey results indicated discrepancies in age and gender, but a large majority—roughly 70.2 percent—of people were able to determine whether or not it was a human or machine that produced each haiku. 29.8 percent overall indicated incorrect answers.

The haiku that people were able to most correctly identify was:

Spring begins shyly
With one hairpin of green grass
In a flower pot



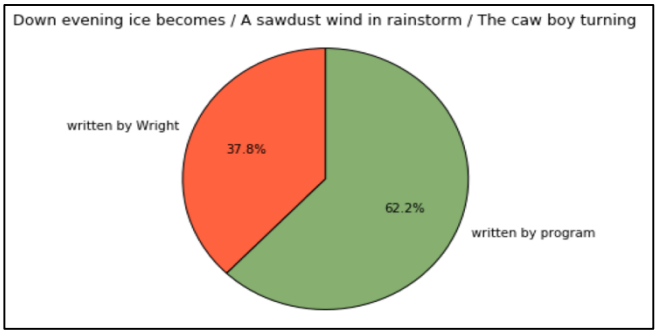
79.9 percent of people accurately indicated that the above poem was written by Wright. Though more women than men participated in the survey, an equal amount contributed to the large slice indicated by the chart above. More specifically, these statistics are indicated in the following graph:



The range of men who participated in answering this question were between 18 and 38 years old. The range of women who were also involved were between ages 13 and 31. This includes a total of 36 men and a total of 38 women. Both men and women of age 21 were most accurate in answering this particular question.

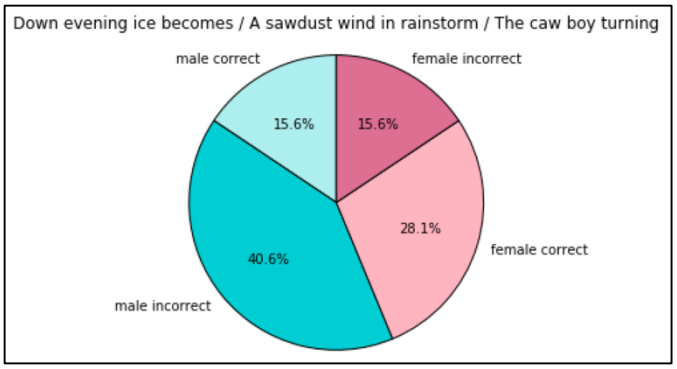
The haiku that people were able to least correctly identify was:

Down evening ice becomes
A sawdust wind in rainstorm
The caw boy turning



This haiku was generated by the algorithm that involved tag restriction. Haikus produced via the baseline algorithm were categorized more so accurately.

Roughly 40 percent of people inaccurately classified the above poem as one that was written by Wright. While this graph does indicate a slight sway in opinion, the majority of people—as expected—are quite adept in recognizing natural language and telling it apart from simulated speech. As indicated below, more men were able to correctly identify this haiku.



The age range of both men and women who participated in answering this question were of the same range as indicated previously. Once again, both men and women of age 21 most accurately identified the correctness of this haiku.

Conclusion

Both text generated based on randomness and restriction of tags can be recognized by the human mind as jargon; that is, not in congruence with the definition of natural—or even creative—language. While my project has succeeded in showcasing this intelligence, an extension of it is needed to reinforce the benefits that such Markov models of natural language could have: detecting verbal abuse on social media platforms, pushing the case for a criminal conviction reversal, or furthering the learning process for students in special education classrooms.

My extensions for this project include:

- Utilizing the CMU dictionary to account for syllables in haikus in efforts to produce an accurate 5-7-5 piece of 17 syllables

- Combining the baseline algorithm and restriction algorithm to enhance the readability and accuracy of the generated poem (i.e. creating a new algorithm altogether)

- Advertising my survey to an audience beyond that mainly consisting of college undergraduates

References

[1] "Markov Model of Natural Language." Markov Model of Natural Language. Web. 06 May 2016. <http://www.cs.princeton.edu/courses/archive/spr05/cos126/assignments/markov.html>.

[2] Shannon, C.E. “A Mathematical Theory of Communication.” Worry Dream. Web. 06 May 2016. <http://worrydream.com/refs/Shannon%20-%20A%20Mathematical%20Theory%20of%20Communication.pdf>.

[3] "Haiku Poems by Richard Wright, Terebess Asia Online (TAO)." *Haiku Poems by Richard Wright, Terebess Asia Online (TAO)*. Web. 13 May 2016. <http://terebess.hu/english/haiku/wright.html>.

[4] Petrov, Slav and et. al., “A Universal Part-of-Speech Tagset.” Google Research. Web. 13 May 13, 2016. <http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/37072.pdf>

[5] "The CMU Pronouncing Dictionary." *The CMU Pronouncing Dictionary*. Web. 11 May 2016. <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>.

[6] Link to Survey: https://docs.google.com/forms/d/1gFJoyH5MmeUkHRZTmixFD6SgORdm8JR0FQU0pznc9Hc/alreadyresponded?c=0&w=1