Final Project - Progress Report
Literature in the Age of Artificial Intelligence

1. Overview

I began earnestly working on this project once I received confirmation to move forward on May 5, 2020. What follows is a progress report. There were items learned during this process that require further investigation when time permits. Realizations that, for someone of my nature, require to be experienced instead of read.

2. Artifact

The artifact I had chosen to work with was Quirinus Kuhlmann's Love-Kiss XLI and I was successfully able to exploit it. As a quick refresher, below was the method suggested in the attachment to the poem by Kuhlmann:

a. Begin a sentence with the given first word.
b. Select one of thirteen words for a given line at random, and use this as the next word in the sentence.
c. Concatenate the first two words from a & b with the last two words of the sentence which are given on the respective line.

Example for Steps a-c using the first line in *Love-Kiss XLI*: *From Fog and Plagues*

d. Construct 11 more sentences following steps a-c with each of the 12 lines in the poem.
e. Attach Kuhlmann's given final two sentences, one of which contains a combinatorial element.

According to Kuhlmann, By step e, one will have created 1 of 622,720,800 variations that are possible.[1] Unfortunately, Kuhlmann miscounted and there are actually 12,454,041,600 possible permutations of this method counting the combinatorial element in step e.

$$\binom{13}{12} \times 2 = \frac{13!}{(13-12)!} \times 2 = \frac{13!}{1!} \times 2 = 13! \times 2 = 12,454,041,600$$

With this tiny mistake, if I was not doing a strict search (as described in the proposal section 3.2.1), the search space would have expanded nearly twenty fold.

It is important to note, that Kuhlmann was kind of close to the number I arrived upon. His number, 622,720,800, was not far different from the result of 13! which is 6,227,020,800 (not counting step e). He was missing a 0 between 6227 and 20800.

2.1. Artifact - Punctuation & Line 7

In order to proceed, there were two matters I needed to address. For the search to have meaning outside of this poem, it was required that all punctuation be removed. Punctuation is usually left out in a search, but I wanted to ensure it was not the cause for a limitation in search results.

On line 7, the final word is noted as G-d. As there is no such word in all the dictionaries (Cambridge, Merriam Webster, etc) I could check, I had to replace this term with "God". This word was chosen as it made the most sense within the context of Kuhlmann's era and the poem itself. Many unique results situations presented themselves due to this correction from my behalf, therefore I made it a point to note it early within this log.

[1] KUHLMANN, QUIRINUS, and Richard Sieburth. "Love-Kiss XLI." *Poetry* 194, no. 1 (2009): 13-16. Accessed April 22, 2020. www.jstor.org/stable/25706521.

3. Evaluation Function

I began the coding side of this project by designing the reinforcement learning environment as provided by the TensorFlow Agents documentation.[2]

Intended operation:

Action Space: Generate Poem or Save State

State Space: A 2x13 embedding of integers.

First row: Embedding for words selected for step b from Section 2.

Second row: Embedding for the corresponding "Total Results" from Google by completing Steps a-c from Section 2 using the word selected for the first row.

Reward: The sum of all integers in the second row representing the cumulative search results from every line in the poem.

Observation Space: An agent would be able to view the entire state space of the previous poem generated and the reward received.

3.1. Input - Word Embedding

First word: A 1x12 array consisting of:

['from', 'come', 'from', 'come', 'the', 'love', 'the', 'seek', 'what', 'with', 'lest', 'where']

The index of the array represents the (line number - 1) from the poem, and the value of the index represents the first word from the poem.

Example: first_word[1] would yield 'come' from the 2^{nd} line in the poem.

Second word: A 12x13 array. The first two rows are shown below:

[['night', 'fog', 'clash', 'frost', 'wind', 'sea', 'heat', 'south', 'east', 'west', 'north', 'sun', 'fire'],
['day', 'blaze', 'bloom', 'snow', 'peace', 'land', 'flash', 'warmth', 'heat', 'joy', 'cool', 'light', 'flames']]

The first index in this 12x13 array represents the (line number - 1) from the poem, the second index represents one of thirteen possible words as given by Kuhlmann.

Example: second_word[1][4] would yield 'peace' from the 2^{nd} line and 5^{th} word above.

Third word: A 1x12 array consisting of:

['and plagues', 'and dread', 'and fraud', 'morning rays', 'at play', 'and bread', 'and god', 'to say', 'be named', 'to avoid', 'be destroyed', 'the day']

The index of the array represents the (line number - 1) from the poem, and the value of the index represents the last two words from the poem.

Example: third_word[1] would yield 'and dread' from the 2^{nd} line in the poem.

Following step c from Section 2 would yield the phrase **'come peace and dread'** from the above examples.

3.2. Output - State Space Embedding & Rewards

After completing steps a-c from Section 2 for each line, the index of the chosen word for step b would be saved (The number 4 in the example from section 3.1) to the first row of the State Space array. By doing this, we would also be able to reconstruct a poem only using 12 integers.

Once a Google search had been performed on a given phrase, the total number of results would be saved to the second row of the State Space array, the index of which would correspond to the line number.

The reward the agent would receive would be a sum of the second row of the State Space. This number would be separate from the State Space so the agent may view which lines in the poem yielded in the largest results through the Observation Space.

[2] TensorFlow is a product of Google and helps with parallelization of the machine learning problem.

4. Resource Problems

As noted in Section 4 of the project proposal, an initial problem I foresaw with my problem was the number of requests through the API. I began to code this section of my problem as its own method and immediately started running into problems. Google, through its direct API, enforces a 50 search maximum on a daily basis. This means that I could only conduct 50 searches a day through their API for free. After this, I would have to pay to conduct searches, the amount which went up as the number of searches went up. I immediately began looking for an alternate as I do not have the funds to run the size of this experiment. I came about two alternates, one of which required payment (SerpAPI: 50,0000 searches for $50), and the second of which became unstable due to the amount of searches being conducted (A combination of the requests library by Django and BeautifulSoup, an assistant for a webscraper).

I could not proceed with SerpAPI and the last of my hopes were dashed when Google began using CAPTCHA to stop my http GET requests through the requests library. At this point I began to look for corpora that I could download and parse in its entirety. There are many available, however choosing corpora means the search space is now limited to the source materials. If a Wikipedia corpora is chosen, only items within Wikipedia could be searched. The larger problem was space requirements. Using Google Colab it is very difficult to work with only 65gb of space. A meaningful corpus would requires terabytes, therefore this avenue was abandoned.

4.1. An antiquated solution to a new problem.

The severity of my lack of progress was a big blow to me after having spent the time to formulate the problem in a manner that I thought would yield interesting results. I went back to the drawing board and my proposal to see if there was something I could do to at least have some results to reflect upon.I could not automate searches but I could manually conduct searches, so I formulated the following solution:

1. Each line had thirteen variations completed via Steps a-c from Section 2. For example, line 7 (See section 2.1 for special notes regarding this line), would yield the following thirteen combinations:

"the guard and god"
"the man and god"
"the work and god"
"the strain and god"
"the art and god"
"the play and god"
"the ship and god"
"the mouth and god"
"the prince and god"
"the rage and god"
"the care and god"
"the greed and god"
"the faith and god"

2. For each strict search, note the total results in google.

By manually processing the poem, I searched 156 lines (12 lines * 13 unique phrases per line) and it resulted in the following matrix where each row represents the line from Kuhlmann's poem, the index of a given row represents the word from step b in Section 2, and the number listed in the matrix represents the number of search results:

$$\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 24500 & 5 & 5 & 0 & 1 & 0 & 0 & 0 & 4 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
27800 & 0 & 1 & 1 & 22400 & 0 & 2 & 38100 & 1 & 6 & 6 & 0 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 15900 & 0 & 0 & 0 & 0 & 5 \\
121000 & 6710000 & 6840000 & 56300 & 2080000 & 289000 & 308000 & 2 & 278000 & 86000 & 7 & 46100 & 6290000 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1550000 \\
4980000 & 4 & 0 & 19000 & 0 & 1 & 0 & 389000 & 4 & 1 & 194000 & 1 & 4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Example: The number 121000 in the 7^{th} row represents 121,000 total strict search results for the phrase **'the guard and god'** derived from the 7^{th} line of the poem and the 1^{st} of 13 words.

5. Analysis

The following observations were made with this data collected.

1. Lines 2 & 4 yielded no search results for any phrases.
2. Lines 1, 8, 12, & 13 yielded less than 10,000 results.
3. If one considers meaningful results to be over 10,000 results: lines 1, 2, 4, 8, 11, & 12 provide no meaning.

Below are three poems generated using the maximum number of search results for each phrase. Lines 2 & 4 (The only lines with variation in the poems below) have been selected at random.

from fire and plagues
come blaze and dread
from fear and fraud
come joy morning rays
.
the tree at play
love fruit and bread
the work and god
seek praise to say
.
what should be named
with care to avoid
lest peace be destroyed
where fear the day
.
all things do change; all things do love; all things are locked in hate
whoever fully thinks this through holds the key to man's estate

---

from fire and plagues
come joy and dread
from fear and fraud
come peace morning rays
.
the tree at play
love fruit and bread
the work and god
seek praise to say
.
what should be named
with care to avoid
lest peace be destroyed
where fear the day
.
all things do change; all things do love; all things are locked in hate
holds the key to man's estate whoever fully thinks this through

---

from fire and plagues
come light and dread
from fear and fraud
come wins morning rays
.
the tree at play
love fruit and bread
the work and god
seek praise to say
.
what should be named
with care to avoid
lest peace be destroyed
where fear the day
.
all things do change; all things do love; all things are locked in hate
whoever fully thinks this through holds the key to man's estate

Below are three poems where lines 1, 2, 4, 8, 11, & 12 have been generated at random, and the most meaningful phrases (in terms of search results) are kept from the other lines. These lines are said to have no meaning according to point 3 of the Analysis section therefore randomness is applied. The 1x12 array following the poem represents the index of the word chosen for a given line.

from wind and plagues
come joy and dread
from fear and fraud
come jokes morning rays
.
the tree at play
love fruit and bread
the work and god
seek port to say
.
what should be named
with care to avoid
lest praise be destroyed
where meat the day
.
all things do change; all things do love; all things are locked in hate
holds the key to man's estate whoever fully thinks this through
[4, 9, 3, 8, 7, 7, 2, 6, 12, 0, 12, 4]

---

from heat and plagues
come warmth and dread
from fear and fraud
come ease morning rays
.
the tree at play
love fruit and bread
the work and god
seek luck to say
.
what should be named
with care to avoid
lest peace be destroyed
where flesh the day
.
all things do change; all things do love; all things are locked in hate
whoever fully thinks this through holds the key to man's estate
[6, 7, 3, 3, 7, 7, 2, 4, 12, 0, 10, 5]

---

from frost and plagues
come light and dread
from fear and fraud
come fame morning rays
.
the tree at play
love fruit and bread
the work and god
seek praise to say
.
what should be named
with care to avoid
lest friend be destroyed
where woe the day
.
all things do change; all things do love; all things are locked in hate
holds the key to man's estate whoever fully thinks this through
[3, 11, 3, 2, 7, 7, 2, 2, 12, 0, 6, 7]

6. Conclusion

Here I must stop further testing and begin wrapping up this report. The experiment was not successful in its entirety. I was not able to create an agent to carry out the task I stated in my project proposal. However, taking the onus upon myself and acting as an agent of purpose, I sought to complete parts of my proposal. The most important of my goals was to find some deeper meaning, as I called it "inherent truths", within Kuhlmann's permutations and I believe to have found at least one.

The seventh line in the poem, referenced many times within this report, contained the largest amount of strict search results. Even though the line was modified as described in Section 2.1, it was done so with care and calculation. Arriving at the sheer volume of results within this line I drew the following inferences.

1. Khulmann must have chosen these thirteen words in line seven because they were linked strongly in his time. Consequently, nearly 400 years later, these words do have some symbolic links as we see they're still being used in phrases today.
2. The word God and its surrounding words are sorts of "keystones" where knowledge is built around.
3. If God is said to be True, the words surrounding or supporting God have had their strength and correlation increased over time.

The second strongest line in the poem was the 12^{th}. Here, the keystone appears to be the word avoid. The total search results do not come close to line two, but it is still meaningful in its own right. The only inference I can draw is that phrases that give warnings are still just as meaningful as in Kuhlmann's time.

6.1. Links to Readings

While working on this project, my mind kept returning to Alan Turing's paper, Computing Machinery and Intelligence. Turing started this paper with a simple question, "Can machines think?". What followed were not strict engineering schematics or code as we witnessed with the ELIZA paper, but rather a logical progression of ideas and resulting questions, some of which contained concrete answers and some which did not.

It seemed to me Turing would conduct an experiment (a thought or physical exercise) investigate his results, and expand upon them. The areas that his paper touches upon such as philosophy and psychology were completely outside the realm of computer science. With his paper, Turing proved to me that these categories or genres that we have defined as separate entities are not all that far apart.

As we continued on our journey past Turning, this became even clearer as we saw three distinct approaches to natural language processing come into view; the schema based approach, the grammar based approach, and the statistical approach. Having only briefly studied these in the class and after applying some context obtained by studying Chomsky in my Computer Science Theory courses, it seems to me that these three approaches have a possibility of being combined through the computation power available today.

With this being said, I am a person who needs to "get their feet wet". I need to run experiments next to any theoretical work that I do, and with that comes a required knowledge base.

7. Future Work

I believe this to be a successful beginning of my exploration of Natural Language Processing. I had held off on taking this course for several reasons. First, even though English is now my primary language, this was not always the case as I am an immigrant from India. The rules of English grammar have always been very difficult for me. By studying four semesters of Spanish at Columbia, I was able to gain a more complete insight into English as a language as all translations went back and forth between the two languages. Second, I have audited several individual lectures of Natural Language Processing a few semesters ago, however, due to the first reason, some sections fell flat. Third, I needed to gain an appreciation for literature in a college level atmosphere. By completing this course and journey through time from Aristotle to GPT-2, I believe I've resolved any reservations I have left in moving forward.

For me, this experiment has been successful in tapping my innate curiosity. I am eager to learn why such strong correlations exist around the word God and much less so for the other words in the poem. There are a number of unsupervised learning methods such as t-SNE that I would like to use along with a large sized corpus to see the neighborhood of the embeddings created through Kuhlmann's permutations. I would also like to use the 1.5 billion parameter GPT-2 model to run some experiments on the phrases from these poems.

At the end of the day, nothing might come of it, or hopefully, with a little luck, some unique structure might come into view.

8. Code for Project

All code is located on my personal Github: https://github.com/tmatrixhy/lit_ai