Mini Project 1 Reflection and Analysis

Kimberly Winter

I. Project Overview

For my Text Mining mini project, I was able to use the sonnets of William Shakespeare to generate rhyming sonnets using a Markovian analysis of the text.

II. Implementation

There were two main dictionaries that I utilized for this project. The first was a dictionary of rhyming words using Shakespeare's rhyme scheme. All of Shakespeare's sonnets follow an ABABCDCDEFEFGG form, meaning that by taking the last word from every line and matching it up with the same letter, I was able to create a dictionary of rhyming words. For example, in the following sonnet:

Look in thy glass and tell the face thou viewest *(A*
Now is the time that face should form another; *(B*
Whose fresh repair if now thou not renewest, *(A*
Thou dost beguile the world, unbless some mother. *(B*
For where is she so fair whose uneared womb *(C*
Disdains the tillage of thy husbandry? *(D*
Or who is he so fond will be the tomb *(C*
Of his self-love, to stop posterity? *(D*
Thou art thy mother's glass and she in thee *(E*
Calls back the lovely April of her prime; *(F*
So thou through windows of thine age shalt see, *(E*
Despite of wrinkles, this thy golden time. *(F*

But if thou live, remembered not to be, *(G*

Die single and thine image dies with thee. *(G*

“Viewest and “Renewest” in the A lines rhyme. In the rhyming dictionary, every word and its rhyme are both keys and values. So, rhymeWords['Viewest'] would return “Renewest” and vice versa. Similarly, if we had a word like “Shoe-est”, it would be a value for Viewest and Renewest.

I also had another dictionary of general words. Because I was starting my lines from the last word to compensate for rhyming, I had to create a somewhat backwards Markov chain, in which I had the previous word as the key and the word before it as the value. For example, for the line “*Look in thy glass and tell the face thou viewest” my dictionary would be:*

generalWords= ['viewest': 'thou', 'thou': 'face', 'face': 'the' … etc]

In order to generate the lines themselves, I generated two at a time to compensate for the rhyming, and put them into a list. I first randomly selected a key from the dictionary of rhyming words, and its matching counterpart, and generated random previous words from the keys in the Markov dictionary. I then concatenated the lines to a single string and printed each line.

III. Results

My program is able to generate sonnets pseudorandomly, such as the one below:

“enlighten thee, my love's breath? The most heinous crime:
Which parts That heavy ignorance aloft That did proceed?
my madness Thou dost thou in thine antique pen;
abundant issue your own love's picture or my argument;
with nature's changing course untainted do if by authority,
the wanton burden Of the filching age will stay,
delves the waves make you on truth miscall'd simplicity,
life, being woo'd of eyes' falsehood Wishing me away,

My nobler To the rarities of thy love bearing:

desire, But that love looks with this written embassage,

thou wilt By oft when I am perjur'd most;

man in the height be thy bravery in vassalage

may I my pupil Thus policy in dark days

From limits far where thou didst thou no praise,”

I was also able to play with the sentiment analysis, and filter out words that fit a specific sentiment, such as a very dark sentiment (where the sentiment value returns equal to or less than .25), which produced sonnets like this:

“expressing, leaves out of absence sour, And to lay;

self thou that ushers And only herald to spend;

Now, while To side this line, And then say,

gainer A closet never can it? O! but tend,

To critic and all-oblivious enmity And given grace impiety,

and praises be my deeds better than a date:

The ills that brightness Which labouring for complexion dimm'd,

but with all his wealth is bent my passion;

renew thy charge? Since brass, nor outward greater grief

smother, sweet self bring?

Which vulgar paper Sweet thief, ear confounds, Look!

what I was certain o'er incertainty, with loss,

and tenure Ruin hath With Time's tyranny,”

When I first implemented this algorithm, it was on a single sonnet. Because it was one sonnet, I knew that the formatting was consistent and that there weren't any weird spacing issues. However, it also lead to a lot of repetition because the program was drawing on one source for many different lines.

When I started importing the sonnets from Project Gutenberg, I edited the document slightly to facilitate easier formatting, but was unable to fix certain spacing issues, which compromised the rhyme scheme slightly. For example, in the dictionary list, there are certain words, such as 'astronomy', whose key is ' ', which implies that there were spacing issues. Given more time, I would've been able to create a more consistent rhyming scheme and kink out this bug. However, with the limited amount of time that I had, I was fine with the amount of rhyming that I was able to include.

IV. Reflection

A lot of the problems that I faced were being too ambitious, and too optimistic about file I/O. Given more time, I would've fixed the rhyming scheme, which, when I had worked on it on one sonnet (which I had naively formatted myself), worked perfectly. I also would've spent more time trying to get my sonnets to make sense (i.e. getting dictionaries with longer connections to facilitate more complete/logical phrases). I also would've liked to have variable line lengths. Right now, in my code, there are 9 words per line (not accounting for any spaces that might have snuck in), meaning that the lines don't always start with a word that normally would start a line. Similarly, when my dictionary was mapping words, words that begin a line were mapped as having “None” in my dictionary. To compensate for this, I picked a random key from the dictionary to put in front of that word. This also doesn't really make sense.

I would also have been more thorough with my sentiment analysis, and I would've loved to have tried phrases instead of just words filtered out. Similarly, a lot of the punctuation does not make a lot of sense, since they were not separated from the words themselves, and were therefore used in the dictionary as part of that word. I would probably not do this given more time.