Mini-Project 3: Text Mining

Kaitlyn Keil

February 2016

1 Project Overview

I chose to analyze the Project Gutenberg text of *The Mabinogion*, as translated by Lady Charlotte Guest. Using the natural language functions of Pattern, I wanted to compare the trend of sentiment through the different sections. I hoped to discover a pattern that the old Welsh stories followed, such as happy beginnings and tragic ends. Time constraints due to other projects and a FWOP performance mandated a simple project for this week.

2 Implementation

Going from a plain text web page to a set of graphs has several major steps. Once the text had been downloaded through Pattern and saved to a file, the header and endnotes needed to be stripped away, which was easiest to do by slicing the string. Keeping the text as a string for the first few steps (this initial parring, finding the table of contents, and then sending it to be cut into sections) let me easily perform methods such as find() and strip(), as well as replace(), which I used a number of times to attempt to fix some unicode issues due to odd Welsh words and an atypical set of double quotation marks. SLicing was especially nice when I was trying to preserve the entire text while still inserting a breakpoint. While lists would likely have worked as well, immutable strings let me alter them more easily without being concerned about where else I was changing a list.

Once I had the stories broken into major sections, I began to work with lists so that I could conveniently process sections in chunks with for loops. I often chose to use indexes in my for loops (such as 'for i in range(len(mylist))') rather than something like 'for section in mylist' because I wanted an easy way to skip over certain sections (such as the table of contents and everything before 'INTRODUCTION') or to reference another list, as I did when saving files under their section names. As I was not counting words or sentences, dictionaries didn't seem necessary.

This grew into lists of lists, which was good for keeping related sections together while still having them split by sentence. I was then able to quickly join them back into strings to use the sentiment analyzer, which gave me tuples to place into another list. I could then extract them, using the index as a second variable, to produce the graphs I needed.

3 Results

Out of all this code, I produced the polarity of the sections and looked at the average over time. I expected to see trends, such as stories often have, with positive beginnings, tense middles, and then happy endings. Or, since they are old tales, I expected pretty negative endings.

The longer ones, such as the Lady of the Fountain and Peredur, the Son of Evrawc, were extraordinarily noisy. This is probably less indicative of more dramatic polarity and more the fact that it condenses more information onto a single graph.

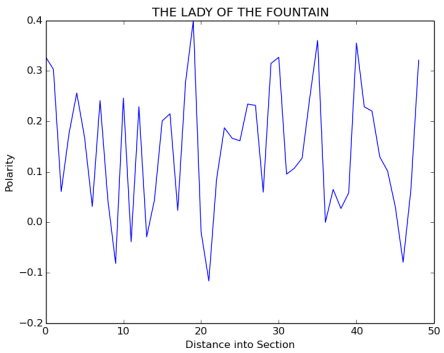

Figure 1: The Lady of the Fountain, which tends to stay between 0.0 and 0.2.

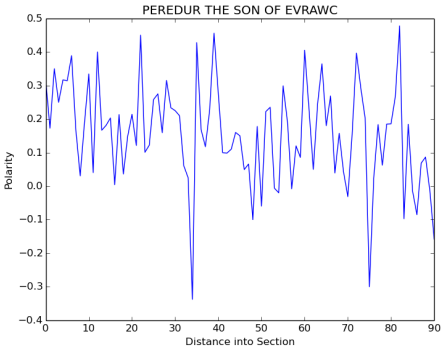

Figure 2: This one varies even more than the Lady of the Fountain, but despite some significant plunges, it tends to be a higher polarity than the Lady.

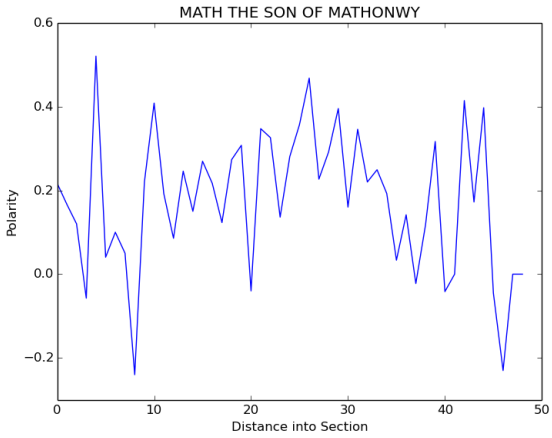

Figure 3: The story of Math, Son of Mathonwy, is closer to following the pattern I expected. Though it too has quite a bit of noise, it has a general arc... in the opposite direction of what I anticipated, but nevertheless.

I had hoped to be able to display a few that followed a similar pattern, but the other 10 sections follow no particular pattern, save that they tend to be more positive than negative. Even this, however, varies between "The Dream of Maxen Wledig," which never drops below 0, to "Manawyddan, the Son of Llyr," which nearly always jumps between positive and negative (though the average is still positive).

4 Reflection

I wasn't particularly ambitious with this project. Initially, I intended to process it quite differently, and compare *The Mabinogion* with emails or other sources, and find who wrote in a similar fashion; comparing it internally was a choice born mostly of necessity. However, I think I likely misinterpreted exactly what 'polarity' meant, as negative polarity isn't the same as dark or dangerous. However, I'm more satisfied with my comments and docstrings on this project, and I think I broke things into fairly good, small functions to build the full program. I could probably improve the versatility of the program, so that it isn't purely focused on just the one file, but could analyze any book with sections, though with graphs as noisy and virtually useless, I'm not sure why this would be all that useful.

I can definitely see using pickling and pattern in the future, however, and I appreciated the chance to experiment with both, as well as dictionaries and becoming more familiar with lists and string methods.