**Name**: Gabriel Pleimann  
gdp7zt@mst.edu   
   
**Course**: CS 5402   
**Assignment**: Semester Project  
**Repository**:   
**Date**: 2022-05-05  
   
   
**Concept Description:**In this project, I will be performing author attribution to hopefully determine who the real author of the Modern Prometheus (Frankenstein) is. There has been some debate as to whether it was Mary Shelley or Percy Shelley that actually wrote it. The current accepted author is Mary Shelley; however, a member of Mensa, a high IQ society, has made the claim for years that Percy Shelley is the real author and uses the claim that the preface is so similar to only Percy Shelley’s writing that he must be the real author. The point of this project is to figure out if that claim has any truth to it with actual computational evidence. I will be using three different data mining techniques to perform an author attribution of the works of Percy Shelley, Mary Shelley, and a couple of other contemporary authors that will be included to make sure the model is accurate.

**Data Collection:**The data that is being used for this project is collected from gutenburg.org which is a website that provides many different books and poetry for free download into different file formats. Since we are going to be performing author attribution on a variety of these works, there is going to be a lot of data wrangling that must be done to the data before we can actually create any models. This will be discussed in the future section Data Import and Wrangling.  
   
**Example Description:**Throughout the various techniques I will be performing to attribute an author to the Frankenstein story, there will be a variety of different attributes used. These will be entirely quantitative attributes such as word length, phrase length, sentence length, word frequencies, and words of different lengths. These attributes listed above are all ratio data – they are quantitative and have a meaningful zero. For example, if a word frequency is 0, we know that word was not used in one story while it may be used very often in all other works. This would suggest there is a good chance that the author of the story we are examining does not match the author of the other stories. However, we would not be able to conclude that with certainty since all of the other stories could be part of a series following the same characters while the other story may be by the same author in an entirely different universe where they don’t speak English. For this reason, we need to combine the results from all or many of the attributes listed above. The names of these attributes explain very well what they are, no further explanation is needed.

**Data Import and Wrangling:**Data Import and Wrangling was a big part of this project – we must take many works from a variety of different authors and import them into our code to perform analysis on them. Gutenberg.org was used to gain access to all the different stories used. I was not able to find an easy way to automatically download and import all the stories from a specific author into my data folder, so I was forced to manually select stories and import them by hand. Therefore, I searched Gutenberg for the authors I was interested and took each of their most popular stories to use in my analysis. Graphical user interface, text, application

Description automatically generated

I created the following two functions to manipulate the standard Gutenberg format. All Gutenberg stories begin with a prologue and ending about copyrights, supporting the website, and some other useless data for our use. The first function takes each story and cuts all of that information out so that we only have the authors original words and there is no extra data skewing our results. The second function encodes each story into a txt file type that can be manipulated without error in python as well as separates all the stories out and applies the previous function to each one. Once all of this is run, we are left with an array of documents each one being a separate story and we are ready to start performing analyses on them.  
   
**Exploratory Data Analysis:**

Chart, histogram

Description automatically generated

I created a histogram plot to visualize the length of the stories and how many were in each length bracket. The shortest of stories were 33715-104,112 words long while the longest of stories were over 1 million words. We will be analyzing 5 different authors from the time – Lord Byron (5 works), Mary Shelley (5 works), Percy Shelley (5 works), Polidori (2 works), and William Godwin (3 works). In the Mensa member’s article, these are the names he mentioned that were who he believed it was possible to have written the Frankenstein story. Obviously we can see that none of these besides Mary Shelley are female, so that is something to note when determining how sexist this claim may be.

Chart, histogram

Description automatically generated

This visual represents the frequency of words that are used across all of these stories. It extracted all of the words from the story and created counts and a graph for us to visualize these counts. Obviously words like “the” and “and” will have the highest peaks and less used words will be what is causing the smaller peaks.   
   
**Mining or Analytics:**   
**K-Means clustering**:

For the purposes of this model, only Percy and Mary were considered possibilities. This is because the K-means works well when there are 2 possible classes and Mary and Percy are the two biggest authors up for debate. For this model, we compared lexical features, punctuation, and syntactic features to 5 different papers written by both Mary and Percy. We also ran the Frankenstein story through some data wrangling functions and then compared it to both Percy and Mary’s known works. The model predicted that Frankenstein is most similar to Mary’s punctuation usage in her other works; however, it predicted that in the lexical and syntactic features category, it is closest to Percy’s other works. We can see the code below:

Graphical user interface, text, application

Description automatically generated

The nltk package was used to tokenize the texts into their individual words/sentences. Sklearn was used for a lot of the comparisons. The overall conclusion from the K-Means clustering model is that Percy is actually the author of Frankenstein. However, it is also important to note that we only compared 3 attributes, it is entirely possible that Mary had a different writing style at the time she wrote this since she was younger and Percy died when he was young, so his writing never got a chance to evolve. There are a wide variety of different things that could go into this result, so it can not be taken as definitive.

**SVM:**

For the SVM model, our data exploration is quite different – we first select 3500 random sentenced from each author’s work to include in our data set (3500 since the smallest # of sentences we have is 3600 from one author). We can find the following statistics about our data once we have it wrangled:

Text

Description automatically generated

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

We can also see the average length of words across all works.

Next we do some data cleaning by removing blocks of white space, removing funky characters that occur from the import process, but make sure to leave characters that are from a foreign language as those can be helpful indicators in our model. Then we split the data into text and author train and test sets of size 80/10 or 13400 and 3350 datapoints.

We then ran a hyper parameterization to find the best value of C:

A picture containing scatter chart

Description automatically generated

And we were able to conclude that C=1 gives the best score overall.

Text

Description automatically generated

After running the bag of words (SVM) model on the Frankenstein dataset, we get a 55.5% accuracy. This means that the model classified over 50% of the text to be written by Mary Shelley based off of her other works. The other 4 authors had less than 20% each giving a clear suggestion to the true author.

**CNN:**

For that model that we didn’t discuss in class, I chose to use a CNN. I am taking advanced topics in AI with Dr. Tripathy this semester, so I figured it would be fitting to apply what I have learned in that class to this project as well.

Table

Description automatically generated

I first ran a 1-gram CNN and got the results above. We see that the accuracy came out to 56% after 10 epochs. Next, I ran a 2-gram CNN and got the following results:

Table

Description automatically generated

Finally, the 3-gram CNN which trained with 96% accuracy after 10 epochs:

Table

Description automatically generated

I started to run a 4-gram CNN, however it was clear that it was not going to reach anywhere near the same accuracy as our 3-gram, so I decided to stop that training process. The final step I did for the CNN was I imported the entire Frankenstein story as its own test set and ran it through our trained 3-gram CNN to see who it would classify the data to be written by. The results:

A picture containing text

Description automatically generated

After running the Frankenstein data through, we are able to conclude that 80% of the data was written by Mary Shelley at the least. Therefore, according to the CNN, Mary Shelley is the true author of Frankenstein.

**Evaluation**:   
A lot of the evaluation of the model was shown above in results snippets, so I will do a recap here. For the K-means clustering, I used a package through nltk to do a lot of the evaluation, considering that our goal was to predict who wrote one paper, we didn’t really need to evaluate the model as much as evaluate what the model believed about the author of Frankenstein. As for the SVM model, we had a precision of 1, an accuracy of 55%, and ran a hyperparameterization to ensure we used the right values of our parameters. We did an extensive data exploration and cleansing to ensure that our data was ready to be evaluated. The model that we didn’t discuss in class, the n-gram CNN, we ran the most evaluation on. We first ran training models on 1-gram, 2-gram, 3-gram and 4-gram models and figured out that the 3-gram model had the best accuracy. Once we figured out that the 3-gram model would work best, we ran the test data (Frankenstein) through the model and found that the model classified it at 80% accuracy as Mary’s work.   
   
**Results**:   
The results of our K-means clustering model suggested that Percy may have been the author of Frankenstein. However, as mentioned in that section, there are many factors that could play into this. Percy died at a young age and Frankenstein is one of Mary’s earliest works, so her writing could have evolved over time whereas Percy’s never had the opportunity to evolve. We also only ran the clustering techniques on 3 different attributes, if there had been more run, it is entirely possible we would have gotten a different conclusion.

The results of the SVM model suggest more strongly that Mary is the true author of Frankenstein. With 55% of the text being classified as her writing whereas no other author included got more than 20% of the text classified as their own writing. This is a huge gap and heavily suggests that Mary is the true author of Frankenstein. Our F-score was also 0.72 for this method.

Finally, the results of the 3-gram CNN also suggest that Mary is the true author of Frankenstein. After training at a 96% accuracy, the network classified more than 80% of the Frankenstein text as being Mary Shelley’s work. This is the method that I personally trust the most to be accurate and have reliable results. Due to the very high probability of this being accurate, it is safe to say that this model strongly suggests that Mary was the true author.

All in all, since 2 of the 3 methods used suggest that Mary is the true author of Frankenstein, I think it is safe to make that assumption. Obviously, we would need to run more evaluation to be certain, but for the limited analysis that we did, the results highly suggest this fact!  
   
**Reference**:

For the implementation of CNN and SVM, I followed very closely to the following github project: <https://github.com/gkhayes/author_attribution>

<https://www.gutenberg.org/ebooks>

<https://aicbt.com/authorship-attribution/>

<https://github.com/PacktPublishing/Learning-Data-Mining-with-Python-Second-Edition>

<https://arxiv.org/pdf/1810.02054v2.pdf>