Leif Christiansen

Detecting Style in Ancient Latin Authors

For all pertinent code, data, and figures see
https://github.iu.edu/leifchri/B565_Project/tree/master

Background

Stylometry is a field that uses statistical and computational techniques to study the style of authors. Stylometry is used to address questions of authenticity, authorship, and chronology, among others. Most famously, Mosteller and Wallace used statistical techniques to determine the disputed authorship of the Federalist papers, valuable documents from the founding of the United States of America.[1] More recently in 2015, stylometric techniques have been used to argue for the authenticity of a hitherto unaccepted work by Shakespeare.[2] Stylometry has also proven useful when applied to more ancient texts in Latin and Greek. Thanks to initiatives like the Perseus Digital Library[3] and the Thesaurus Linguae Graecae,[4] a large number of ancient Latin and Greek texts are available in digital form. Yet despite this, Classicists have remained reticent to adopt these new techniques as readily as other scholars. The purpose of this project is to further demonstrate the potential benefits of statistical and computation techniques when applied to ancient, Western texts. Specifically, I will be performing stylistic analysis of 8 ancient Latin authors, building a classifier to determine the author of a given text.

Data Selection and Acquisition

For the sake of simplicity, Latin was chosen over ancient Greek as Greek requires special encodings for the text. Texts were downloaded from the thelatinlibrary.com, a collection of Latin works in simple HTML. It is important to note that these texts are not intended for research purposes, they have not been rigorously checked for quality and accuracy. However, these are the most complete and easily available (i.e. free and without licensing restrictions) versions of the texts in digital form. Thus, for the scope of this project, the quality of the texts was deemed acceptable.

The complete works of 8 ancient Latin authors were downloaded:

Ammianus, Caesar, Cicero, Frontinus, Horace, Juvenal, Livy, and Justin

Authors with at least 10 works were chosen arbitrarily. More than 8 authors met this criterion but only 8 were selected, again for the sake of simplicity.

Data Pre-Processing

Texts downloaded from thelatinlibrary.com were in HTML format and without standardized naming conventions. Texts were stripped of HTML code, renamed to include the name of the author, and then written to text files using LatinHTMLtoTxt.py. The spacing was also standardized by removing tabs and double spaces. This was done to assist the word counting performed in the classifier.

[1] See *Inference and Disputed Authorship: The Federalist* by Frederick Mosteller and David L. Wallace.

[2] See http://www.latimes.com/science/sciencenow/la-sci-sn-shakespeare-play-linguistic-analysis-20150410-story.html.

[3] http://www.perseus.tufts.edu/hopper/

[4] http://www.tlg.uci.edu/demoinfo/demo.php

Feature Selection

The problem addressed here is one of text classification. The standard approach to text classification is to represent texts as “bags-of-words” with each vocabulary word its own feature. Even though our extant Latin texts have a small core vocabulary compared to modern English, 4,000 in Latin compared to 10,000 for an English college student, there are still far fewer texts than vocabulary words.[5] Thus, in order to make the problem more tractable we must perform some form of feature reduction.

Function words have been shown to be effective for distinguishing writers’ styles. In fact, Mosteller and Wallace used function words in the aforementioned study of the *Federa*list papers. Mosteller and Wallace provide the following definition for function words: "the filler words of a language, such as a, an, by, to and that. Generally they include prepositions, conjunctions, pronouns, and certain adverbs, adjectives, and auxiliary verbs.''[6] Using function words as our features, we may drastically reduce the dimensionality of our data.

Function words have been specifically applied to Latin texts by Bernard Frischer in his study of Horaces’ *Ars Poetica*.[7] Frischer identified 51 Latin function words. See Table 1 for a complete list.

a	cur	mox	post	sine
ab	de	nam	quia	sive
ac	donec	ne	quidem	sub
ad	dum	nec	quodsi	tam
an	enim	neque	quoque	tamen
at	et	neu	saepe	tandem
atque	etiam	nisi	sed	ubi
aut	iam	non	seu	unde
autem	in	nunc	si	ut
cum	inter	per	sic	vel
				velut

Table 1: 51 Latin function words identified by Frischer

According to Frischer, a function word exhibits one of three patterns in a writer’s corpus: 1. The word usage varies randomly from work to work, 2. The word usage varies according to a pattern, or 3. The word usage remains fairly constant. In establishing a chronology, Frischer was interested in case 2. For our purposes of distinguishing authors, one would expect case 3 to prove the most informative. However, unlike Frischer we will not be manually examining these patterns. Rather, we will rely on a supervised algorithm to select the most useful features for the

[5] See http://www.orbilat.com/Languages/Latin/Alternative_Grammars/Harris_Grammar/Latin-Harris_16.html for the relevant excerpt from *The Intelligent Person’s Guide to the Latin Language.*

[6] *Inference and Disputed Authorship: The Federalist* by Frederick Mosteller and David L p. 17.

[7] See *Shifting Paradigms: New Approaches to Horaces’ Ars Poetica* by Bernard Frischer.

classification problem.

In order to standardize the features, function words were measured per 1000 words. Each text was split into 1000-word components with the leftover words discarded, e.g. in a text with 2403 words, the last 403 words would not be used. A 51-dimensional feature vector was generated for each 1000-word segment, each feature corresponding to the number of occurrences of each function word. This approach had the added benefit of generating additional observations from the data set. However, even with these added observations, further dimensionality reduction was still required to perform the classification, as we will see shortly.

Visualization

The data generation process resulted in a total of 891 observations distributed as follows:

Author	# of Observations
Ammianus	39
Caesar	49
Cicero	214
Frontius	22
Horace	41
Juvenal	19
Livy	475
Justin	32

Table 2: Number of observations in data set per author.

Clearly, some authors were far more verbose than others resulting in a highly skewed class distribution. This is an issue to which we will return later.

First, let us examine the class means as star plots. The R star plot requires that each column have values [0,1]. Columns were scaled independently to have std=1 and then normalized to [0,1].

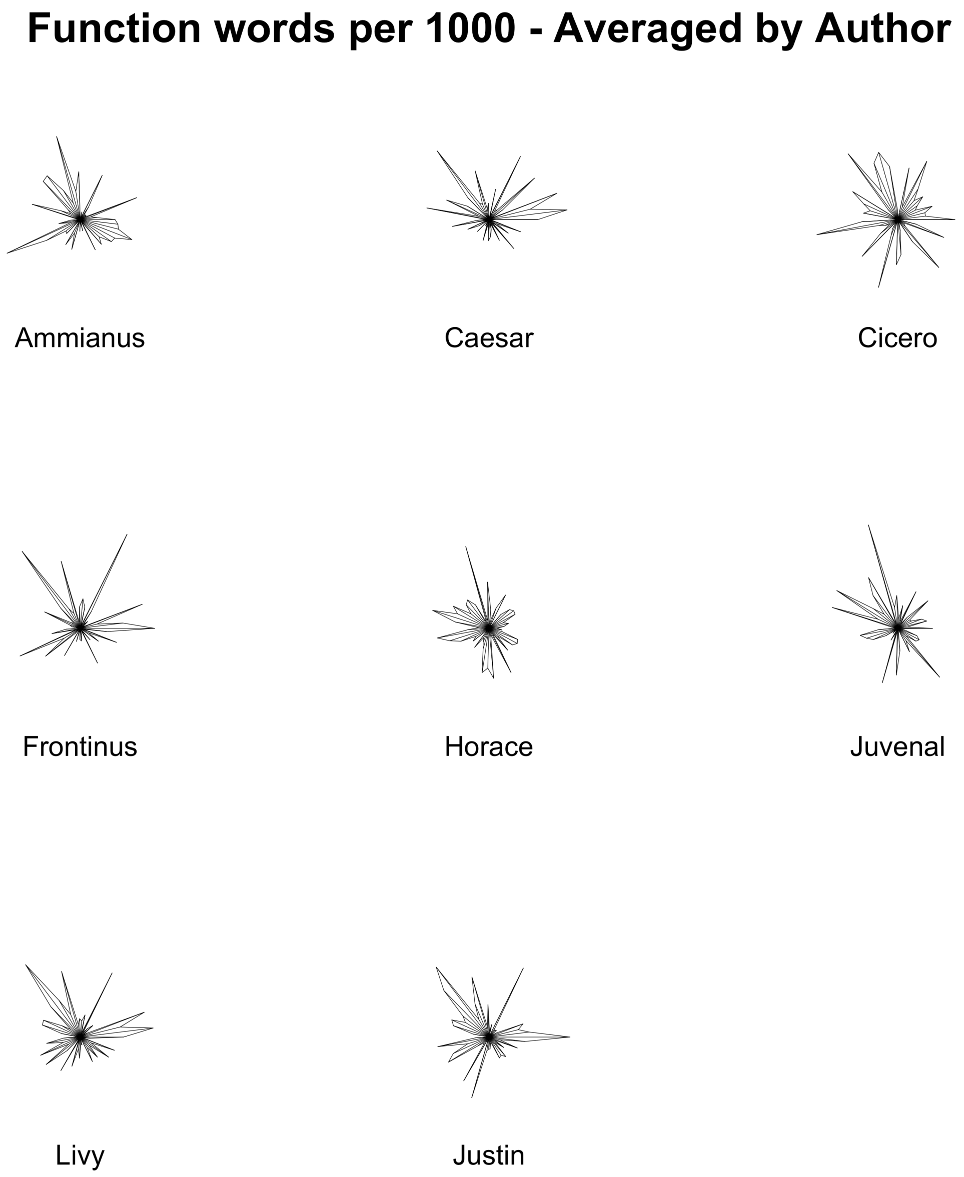

Figure 1[8]

On visual inspection the class means appear to be fairly distinct, although there are some interesting similarities. Cicero appears the most balanced and verbose, having the largest number of features with high values. This matches well with historical accounts of Cicero as a supremely long winded orator. Frontinus, Livy, Caesar, and Justin all exhibit a similar v shape in the upper half of the plot with another protrusion out to the right. However, as Simpson's Paradox teaches us, we should be careful to rely on patterns observed in summed data.

[8] Sorry the plots are so small; I couldn't figure out how to change their size in R. If the pdf image is not high enough resolution to zoom, a high resolution version of this graphic is available in the Github repo under figures.

In order to investigate the potential impact of Simpsons paradox, star plots were generated for nine randomly selected observations for each author. The plot for Horace offers an elucidating example.

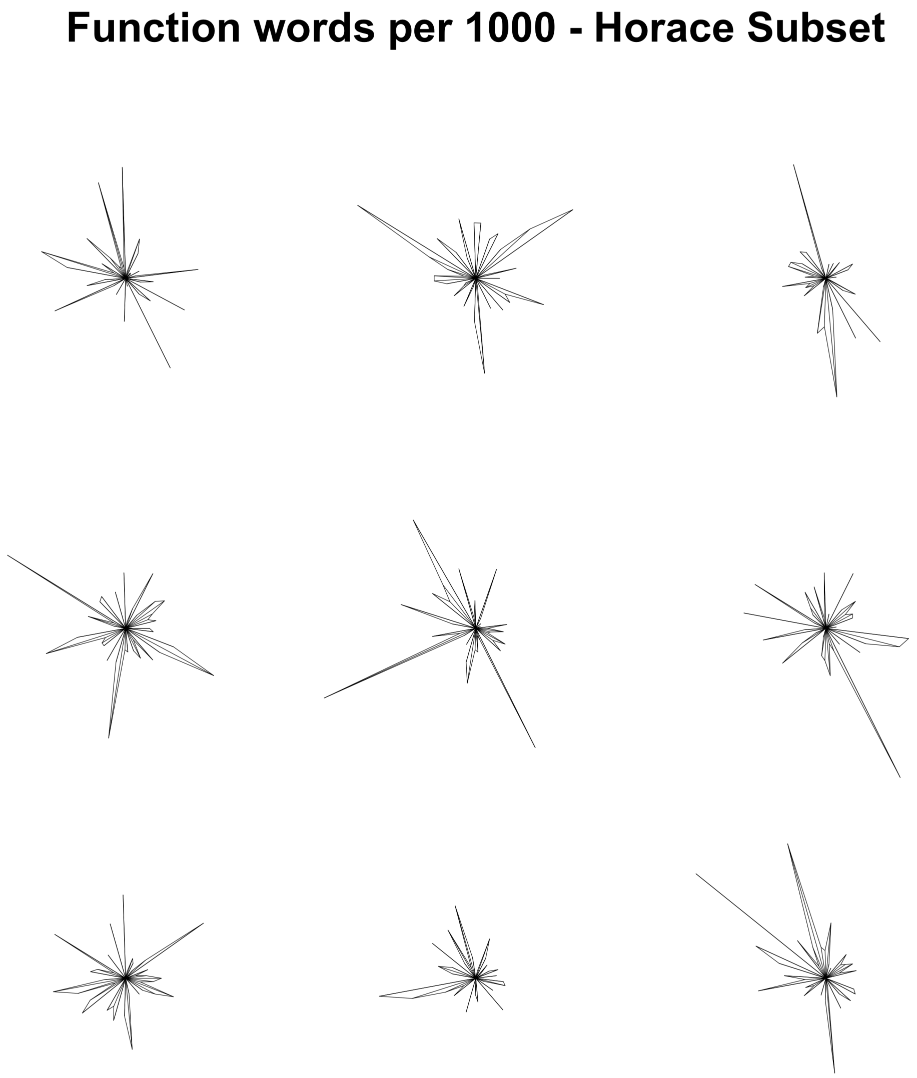

Figure 2: Star plots for a random subset of 9 observations.

Just in this random sample, there appears to be a decent amount of variation. Almost every observation has a different feature with the highest value. This is not to say that the mean was misrepresentative, likely it did in fact capture the function words with fairly consistent usage throughout an author's entire works. However, Figure 2 does demonstrate that there are some other qualities of our data set. Possibly, the variety between observations is a byproduct of our bin size of 1000 words. Since most of these works are only a few thousand words, one would expect 1000 words to be fairly representative. But perhaps 1000 words is too small a bin to capture a representative sample. Regardless, let us see if our classifier is able to detect the patterns that seem to be present in the star plots.

Classifier

For the classifier, Naïve Bayes using a Guassian mixture model to approximate probability densities was useed. Naïve Bayes was chosen for two primary reasons. First, as has been discussed in class, Naïve Bayes is a robust classifier that often performs well. Second, in this case the independence assumption seems valid. In studying Horace, Frischer used chi-squared tests to demonstrate the independence of his chosen function words. Since function words by nature allow for limited variation in their functional use, it seems reasonable to assume that the independence observed in the works of Horace generalize to the entire Latin corpus.[9]

However, in this particular case there are a couple drawbacks to using Naïve Bayes. First, Naïve Bayes takes into account the *a priori* class distributions. As was seen earlier, for our sample this distribution is heavily skewed. Furthermore, it is even dubious to assume that the distribution of authors among extant Latin works is at all similar to the ancient distribution. It does seem reasonable that authors more popular in ancient times were more likely to be preserved due to wider dispersal, and that is certainly the case with authors like Livy. However, much of what has been preserved has also been filtered through the preferences of generations and generations of other individuals from a variety of cultures. Many of these works were copied and preserved by Christian monks and Arabic scholars. Certainly the tastes of these groups, culturally distinct from the ancient Latin readership, influenced what they sought to preserve and what they did not. It is unclear what *a priori* distribution would be desired. Do we want to learn the distribution of extant works with the assumption that all extant works had similar factors determining their preservation? Or do we want to learn the ancient distribution? Second, Naïve Bayes applied to continuous variables is especially susceptible to high dimensional data. When the ratio of observations to features is high, there is a high likelihood that the covariance matrix is singular, i.e. features are correlated. Unfortunately, this is the case with our data set. For all but two of the classes, the number of observations is less than our total number of features.

Ultimately, the pros of Naïve Bayes were determined to outweigh the cons. It is difficult in this case to make any clear determinations as to the *a priori* distribution. But theoretically, similar pressures would act on the transmission of the majority of these Latin works so considering the *a priori* distribution does not seem an egregious assumption to make. Additionally, the observation to feature ratio may be easily addressed using dimensionality reduction techniques. Principal Components Analysis (PCA) was used to select 5 new features with greater predictive power.[10]

The classifier was trained on a training set consisting of half of all samples with the *a priori* distribution maintained, half of class one was assigned to the training set, half of class two, etc. The test set was the complement of the training set. See the error of the

[9] This would be a claim worth testing in the future. Similar chi-squared tests could be run on the entire data set.

[10] There are certainly more intelligent ways for selecting the subset of new features but due to the constraints of this project a simple threshold was used.

classifier on the test and training sets in Table 3. Values were rounded to three significant digits.

Run	Error on Testing	Error on Training
1	0.109	0.093
2	0.125	0.074
3	0.116	0.074
4	0.123	0.068
5	0.112	0.068

Table 3: Error rates for 5 runs of the classifier.

As can be seen in Table 3, the classifier performs quite well on the data set. Furthermore, the error on the test and error on the train are fairly close, potentially indicating good generalizability for our model. Let us examine some of the errors in more detail. For the sake of reproducibility, the following was generated using a seed of 12.

Table 4 shows the total error rate on the test set as well as the error rate per class.

	Error Rate
Total	0.11607
Ammianus	0.2
Caesar	0.28
Cicero	0.01869
Frontinus	1.0
Horace	0.19047
Juvenal	0.2
Livy	0.03781
Justin	0.8125

Table 4: Error rate break down for one run.

The error rates for Cicero and Livy are substantially lower than the total error rate. For Ammianus, Caesar, Horace, and Juvenal the error rates are a decent amount higher. But for Frontinus and Justin the error rates are atrocious. Not a single case of Frontinus was classified correctly. Clearly, the skew of the *a priori* distribution has had a profound effect on the performance of the classifier. There is a heavy preference for Cicero and Livy. What is interesting is that the two classes on which the classifier performed worst are part of the group of Livy, Frontinus, and Justin identified in Figure 1. To further investigate this, we may examine the most frequent incorrect predictions. See Table 5 for the most often misclassifications of each class.

Actual	Most Frequent Incorrect Prediction(s) (Frequency)	# of Errors / # of Obs (Error Rate)
Ammianus	Livy (0.5)	4/20 (0.2)
Caesar	Livy (0.571)	7/25 (0.28)
Cicero	Caesaer (0.00) Livy (0.00)	2/107 (0.01869)
Frontinus	Livy (0.8182)	11/11 (1.0)
Horace	Juvenal (1.00)	4/21 (0.19047)
Juvenal	Livy (0.5) Horace (0.5)	2/10 (0.2)
Livy	Ammianus (0.333) Justin (0.333)	9/238 (0.03781)
Justin	Livy (1.00)	13/16 (0.8125)

Table 5

Interestingly, the grouping of Frontinus, Justin, and Livy does seem to be represented in our per class error. The classifier is clearly biased towards predicting Livy. But even so, Livy is predicted much more often for Frontinus and Justin than the other classes. Also interesting is the fact that every incorrect prediction of Horace was Juvenal. Returning to Figure 1, one may see that the star plots for these two classes do appear strikingly similar.

In order to potentially lessen the bias towards Livy and Cicero, a new data set was created with less skew by removing a number of the texts of Cicero and Livy.[11] See the new distribution in Table 6.

Author	Number of Observations
Ammianus	39
Caesar	49
Cicero	72
Frontius	22
Horace	41
Juvenal	19
Livy	62
Justin	32

Table 6

Then the classifier was trained and tested using the same method as before. Table 7 shows the new error results per class.

[11] This new dataset is in /Works-data-fix_dist in the Github repo.

Actual	Most Frequent Incorrect Prediction(s) (Frequency)	# of Errors / # of Obs (Error Rate)
Ammianus	Livy (1)	2/20 (0.1)
Caesar	Livy (0.5)	4/25 (0.16)
Cicero	Caesaer (0.33) Frontinus (0.33) Livy (0.33)	3/39 (0.0769)
Frontinus	Livy (0.8571)	7/11 (0.6364)
Horace	Juvenal (0.5) Livy (0.5)	2/21 (0.0952)
Juvenal	Horace (1)	2/10 (0.2)
Livy	Frontinus (0.5714)	7/31 (0.2258)
Justin	Ammianus (0.5) Livy (0.5)	2/16 (0.125)

As expected the error rate on all classes expect Livy and Cicero improved. The error rate for Justin improved the most drastically. However, the difficulties distinguishing between Frontinus, Livy, and Justin and between Horace and Juvenal are still present. Surprisingly, even with the drastic reduction in the number of Cicero observations, Cicero maintains the lowest error per class. This would seem to validate our initial observation that Cicero appears the most unique in Figure 1.

Conclusion

This work has demonstrated the potential for using modern statistical and computational techniques to study ancient Latin texts. However, in order to convince conservative classicists more direct answers to classical questions is likely necessary. Regardless, using function words and a Naïve Bayes classifier some interesting patterns in the selected ancient authors were discovered. This patterns are worth further investigation both through extending the techniques used here and consulting existing scholarship on the ancient authors. If the similarity detected by our classifier parallels similarity determined by Classicists, this would certainly serve to further validate the method.