Poetic Variation

Adam Flaherty

May 5, 2012

Abstract

The purpose of this project is to determine if factor analysis can uncover correlations between parts of speech within different works of poetry.

1 Introduction

The first recorded frequency analysis on language is from the ninth century by the "philospher of the Arabs" Abu al-Kindi (Singh, 8). In hopes of deciphering encrypted messages, he studied the frequency each letter occured in Arabic and found that the distribution of letters was very stable. Cryptographers have since applied such frequency analysis to other languages in order to crack substitution ciphers based on the stability of letter frequency. Such analyses have noted that some letters appear more often than others in a broad glance at the English language (Hoffstein, 6). Figure 1 presents the overall English letter frequency on a scale of 40,000 to show when deciphering a large message, the most frequently used letter would most likely represent e or t.

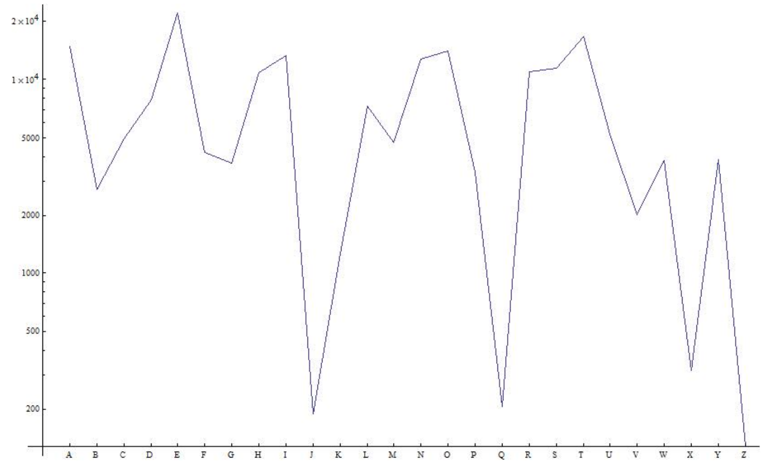

Figure 1: Letter Frequency of the English Language (Hoffstein, 6)

The purpose of this research is to verify if particular authors, or poetry as a genre, contain stable patterns not at the level of letter frequency, but rather at the level of parts of speech. By finding the frequencies and correlations between parts of speech in various works of poetry, we will be able to verify these results by using a statistical method known as factor analysis.

2 Factor Analysis

Factor analysis did not originate in a mathematical environment. Rather, factor analysis formed within the field of psychology to help find correlation between factors that cannot be directly observed (Mardia, 255), by starting with a set of measurable variables. These measurable variables will be represented by a linear combination of a fewer number of the unobservable factors (Mardia, 255; Rencher, 408). Such linear combinations will resemble the following:

$$x_i = \Lambda_{i1} f_1 + ... + \Lambda_{ik} f_k + \epsilon_i \text{ , for } k < i$$

In this linear combination, Λ_{ik} represents the loadings of the kth factor f_k, or how each variable x_i is dependent on the kth factor, and ϵ_i represents an independent random error term for x_i.

It is best to use normalized data in a factor analysis to keep all the variables comparable and to make the results more accurate (Rencher, 419). A normalized data set is a modified set of initial data scaled with respect to one of the variables to remove its presence and so the rest of the variables can be compared. In this study, the frequency of a given part of speech will be normalized with respect to word count; the effects of word count, which varies with each poem, will be removed. Once the variables are comparable and the data is normalized, a correlation matrix can be constructed. The correlation matrix is an $p \times p$ matrix where p is the number of variables in the data set. The nondiagonal elements of this matrix represent correlation coeffiecients between each pair of variables. To find the correlation matrix, we will start with the covariance matrix, an $p \times p$ matrix containing the covariance between the ith and jth variables, denoted $E[(v_i - \mu_i)(v_j - \mu_j)]$. For the ith variable v_i with mean μ_i,

$$\begin{aligned}
S &= E[(v-\mu)(v-\mu)^T] \\
&= E\begin{pmatrix} v_1 - \mu_1 \\ v_2 - \mu_2 \\ \vdots \\ v_p - \mu_p \end{pmatrix} (v_1 - \mu_1, v_2 - \mu_2, \ldots, v_p - \mu_p) \\
&= E\begin{pmatrix} (v_1-\mu_1)^2 & (v_1-\mu_1)(v_2-\mu_2) & \ldots & (v_1-\mu_1)(v_p-\mu_p) \\ (v_2-\mu_2)(v_1-\mu_1) & (v_2-\mu_2)^2 & \ldots & (v_2-\mu_2)(v_p-\mu_p) \\ \vdots & \vdots & & \vdots \\ (v_p-\mu_p)(v_1-\mu_1) & (v_p-\mu_p)(v_2-\mu_2) & \ldots & (v_p-\mu_p)^2 \end{pmatrix} \\
&= \begin{pmatrix} E(v_1-\mu_1)^2 & E(v_1-\mu_1)(v_2-\mu_2) & \ldots & E(v_1-\mu_1)(v_p-\mu_p) \\ E(v_2-\mu_2)(v_1-\mu_1) & E(v_2-\mu_2)^2 & \ldots & E(v_2-\mu_2)(v_p-\mu_p) \\ \vdots & \vdots & & \vdots \\ E(v_p-\mu_p)(v_1-\mu_1) & E(v_p-\mu_p)(v_2-\mu_2) & \ldots & E(v_p-\mu_p)^2 \end{pmatrix} \\
&= \begin{pmatrix} v_{1,1} & v_{1,2} & \ldots & v_{1,p} \\ v_{2,1} & v_{2,2} & \ldots & v_{2,p} \\ \vdots & \vdots & & \vdots \\ v_{p,1} & v_{p,2} & \ldots & v_{p,p} \end{pmatrix}
\end{aligned}$$

Now that we have constructed the covariance matrix, we can construct the correlation matrix. First, let $s_i = \sqrt{v_{ii}}$, or the standard deviation of the ith variable, then let D be the diagonal matrix defined by:

$$D = \begin{pmatrix} s_1 & 0 & \dots & 0 \\ 0 & s_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & s_p \end{pmatrix}$$

The correlation matrix, R will be

$$\begin{aligned} R &= D^{-1}SD^{-1} \\ R &= \begin{pmatrix} 1 & r_{1,2} & \dots & r_{1,p-1} & r_{1,p} \\ r_{2,1} & 1 & & & r_{2,p} \\ \vdots & & \ddots & & \vdots \\ r_{p-1,1} & & & 1 & r_{p-1,p} \\ r_{p,1} & r_{p,2} & \dots & r_{p,p-1} & 1 \end{pmatrix} \end{aligned}$$

Where $r_{i,j} = \dfrac{s_{i,j}}{s_i s_j}$, which represents the correlation between each pair of variables.

Now that the correlation matrix has been found, we need to find the eigenvalues of R to determine how many factors need to be extracted.

$$\det[\lambda I - R] = 0$$

Solving for λ in the above equation will yield a set of n eigenvalues, $\lambda_1, \lambda_2, \dots, \lambda_n$. With these eigenvalues now determined, we will look at two ways to find the number of factors to extract. The number of factors chosen is very important, as choosing too few factors will result in very high loadings, "[but] with too many factors, factors may be fragmented and difficult to interpret convincingly"(Everitt, 69). In the first method, we study the plot of the eigenvalues, known as a scree plot. A scree plot resembles a reciprocal function, $f(x) = \frac{1}{x}$, with a sharp drop in the graph before leveling out completely. The number of factors extracted using this method is n, where the nth factor is located on the scree plot directly before the drop. Using the second method, we look at the values of the eigenvalues. The sum of the eigenvalues for the number of factors chosen will amount to "a predetermined percentage, [usually] 80%," of the total sum of all the eigenvalues (Rencher, 427). Once we have determined the number of factors to use, we can find the loadings using the corresponding eigenvectors by solving the equation $R\vec{x}_n = \lambda_n \vec{x}_n$ for $\vec{x}_n$. The loadings of each factor, the vector $\vec{c}_n$, can be obtained by multiplying the eigenvectors by the square root of the corresponding eigenvalue.

$$\vec{c}_n = \sqrt{\lambda_n}\vec{x}_n$$

Once the factor loadings are found, they can be interpreted based on the model from where they were evaluated. We can also find the factor scores for each factor to help with the interpretation across the different variables.

3 Literature Review

Studying poetry from a numerical standpoint has been in effect for decades. In Josephine Miles' *Renaissance, Eighteenth-Century, and Modern Language in English Poetry* from 1960, around 200 different poets are studied for the frequency of adjectives, nouns, and verbs in their works (Miles, 1). Miles' work contains four tables, each illustrating a different aspect of study, but only tables 1 and 3 are relevant to this study. The first table depicts the raw frequencies of each part of speech from the first 1000 lines of a given anthology from each author. Table 3 shows a basic relationship between the authors through time based on adjectives per words. While the mathematics only involves simple counting, Miles was able to illustrate in table 3 an interesting relationship between adjectives and verbs over time. In the late 1400's and early 1500's, there were very few adjectives per verb, but around the 1700's, the reationship shifted so there were more adjectives per verb. As the table approached 1900, the number of adjectives per verbs became much more balanced.

Factor analysis plays an important role in *Variation Across Speech and Writing*, 1988, Douglas Biber's well regarded in-depth analysis of the English language across both written and spoken works. In his work, Biber utilizes factor analysis to identify underlying co-occurance patterns of English. After studying previous research in the field, he gathered a sample of 481 works from 23 genres (15 written, 6 spoken, and 2 types of letters) that span the "full range of situational variation" in the English language (Biber, 65). Biber performed a preliminary analysis based on 67 linguistic features. Appendix A lists the different linguistic features used in Biber's study along with the results from his analysis based on the normalized frequencies from sample of texts he used. Each of the frequencies was normalized to a text of 1000 words as to keep the results of his study accurate. That is, the data gathered was put into terms as if the data were taken from a sample of 1000 words using the normalization equation $v_{norm} = \dfrac{v \cdot 1000}{l}$ where v is the variable to be normalized and l is the word count of the specific sample.

With this data, Biber used a principal factor analysis, because he explains that "the use of a factor analysis in linguistics is usually exploratory rather than confirmatory"(Biber, 81-82). In other words, since there are no previous models or theories to draw hypotheses from, factor analysis in linguistics is exploratory. Thus, Biber decided to use the most common form of factor analysis, the principal factor analysis described in Section 2. In his study, Biber extracted seven significant factors, in which he was able to categorize each of the genres he used. This categorization displays the overall use of particular linguistic features in each genre. By observing these patterns, Biber is able to recognize the differences and similarities between each genre.

4 The Code

The first thing required for a factor analysis is data. Whereas Biber identified 67 linguistic characteristics over several genres, this study will focus on the basic parts of speech in the genre of poetry. In order to avoid manual counting, we will implement *Mathematica* to make the process easier.

By studying code provided by Dr. Patrick Bahls of UNCA, we created *Mathematica* code that could take an input of words and sort them by parts of speech. The code starts by changing the input string into all lowercase letters, and removing all numbers, symbols, and punctuation. Next, the code imports lists of parts of speech from text files garnered from the *Corpus of Contemporary American English* (Davies, 2008) and finds the intersection between all these lists to create a set of all words with multiple parts of speech. The *Corpus* is a compilation of over 425 million words from various genres of works dating back to 1990 created by Mark Davies of Brigham Young University. The *Corpus* offers a free list of nearly 500,000 words that can be sorted by parts of speech. After the lists are finalized, the code sorts the original string into subsets of parts of speech, and identifies any words within the string with multiple parts of speech or any words not identified within the master lists. In the case of the latter, the words are manually added to the appropriate lists before the code is run again with the more inclusive lists. The following is a piece of the code that creates the sublist of verbs:

```
VerbList=Flatten[Table[Intersection[{stringlist[[i]]},verb],{i, Length[stringlist]}]]
```

A simple intersection between the input string and the list of all verbs does not suffice is because the intersection only pulls unique elements. Any duplicate words in the text would be passed over and the resulting sublist would be incomplete. This line of code looks at each element of the input string seperately, cross-referencing them with the parts of speech lists. If there is an intersection, the word is put into the table, and if there is not, an empty set is inserted. The `Flatten` command removes all the empty sets leaving all the relevant elements to be counted. The entire code used can be found in Appendix C.

5 Input

A total of 30 poems were selected to be analyzed. These poems were authored by three American authors and three British authors, each from a different time period of their respective countries, and from these six authors, five poems were chosen at random. The following is the list of chosen poems organized by author.

1. From Lord Byron (1788-1824)
   - "Epitaph to a Dog"; "My Soul is Dark"; "So We'll Go No More A-Roving"; "The Cornelian"; "The First Kiss of Love"
2. From Emily Dickinson (1830-1886)
   - "Awake Ye Muses Nine"; "Because I Could Not Stop for Death"; "Because That You are Going"; "I Stepped from Plank to Plank"; "The Saddest Noise, The Sweetest Noise"
3. From Robert Frost (1874-1963)
   - "Gathering Leaves"; "On Looking Up by Chance at the Constellations"; "Storm Fear"; "The Telephone"; "The Wood Pile"
4. From Edgar Allan Poe (1809-1849)
   - "Conqueror Worm"; "Dream Within a Dream"; "Eulalie"; "Lenore"; "The Raven"
5. From Jonathan Swift (1667-1745)

- “A Description of a City Shower”; “A Satirical Elegy”; “The Beasts’ Confession to the Priest”; “The Day of Judgement”; “The Puppet Show”

6. From Oscar Wilde (1854-1900)

    - “Athanasia”; “Requiescat”; “The Dole of the King’s Daughter”; “The Grave of Shelley”; “The Harlot’s House”

6 Output

Table 1 shows the normalized frequencies of the nine variables (verb, noun, pronoun, adjective, adverb, preposition, conjunction, interjection, and article) from each sample obtained from the *Mathematica* code. The data were normalized to a string of 100 words. Word count has also been included but has not been normalized, because the other data were normalized with respect to it.

At this point we use the statistical software `R` to perform the factor analysis. Before the actual analysis, we must first decide how many factors to extract. To do this, we must look at the scree plot of the eigenvalues of the correlation matrix of the normalized data, which is obtained using the methods in Section 2. Table 2 shows the correlation matrix of the normalized data.

Now that we have the correlation matrix, we need the eigenvalues of the matrix so we can figure out how many factors to extract. Again, using `R`, we obtain the eigenvalues and use them to create a scree plot that will be used to determine the number of significant factors. By studying the scree plot in Figure 2, we can determine the number of factors to extract is about five or six, the number of eigenvalues before the drop. To determine if five factors would be sufficient, we rely on the second method of determining the number of factors as

Sample	Verb	Noun	Pronoun	Adj.	Adv.	Prep.	Conj.	Interj.	Article	Word Count
Epitaph to a Dog	23.08	35.38	12.31	4.62	1.54	20.00	6.15	0.00	7.69	65
My Soul is Dark	31.09	21.85	13.45	10.08	3.36	10.08	17.65	3.36	4.20	119
So We'll Go No More A-Roving	21.52	17.72	7.59	10.13	5.06	11.39	18.99	2.53	16.46	79
The Cornelian	25.11	19.28	18.83	10.31	3.59	11.21	13.00	0.45	7.17	223
The First Kiss of Love	28.16	28.57	13.47	8.57	2.45	15.92	7.76	2.45	9.39	245
Awake Ye Muses Nine	24.04	25.24	10.82	10.58	3.13	8.17	13.70	1.68	11.54	416
Because I Could Not Stop for Death	19.69	23.62	15.75	4.72	7.87	11.02	18.11	0.79	11.02	127
Because That You are Going	27.27	17.11	25.13	5.88	4.81	11.76	13.90	0.53	4.81	187
I Stepped from Plank to Plank	20.93	23.26	23.26	11.63	2.33	13.95	6.98	0.00	11.63	43
The Saddest Noise, The Sweetest Noise	19.09	21.82	19.09	9.09	8.18	10.91	10.00	3.64	10.91	110
Gathering Leaves	22.43	22.43	14.02	10.28	3.74	22.43	16.82	1.87	5.61	107
On Looking Up by Chance at the Constellations	27.33	21.33	11.33	12.67	4.00	20.00	17.33	0.67	6.67	150
Storm Fear	21.15	19.23	8.65	13.46	4.81	17.31	15.38	2.88	10.58	104
The Telephone	33.83	14.29	28.57	3.01	5.26	9.77	12.03	3.01	5.26	133
The Wood Pile	21.33	16.07	19.67	15.51	3.60	16.90	14.68	2.22	6.09	361
Conqueror Worm	22.61	26.09	9.13	7.39	3.48	16.52	13.04	0.87	12.61	230
Dream Within a Dream	24.65	13.38	19.72	5.63	4.93	14.08	16.90	2.11	11.27	142
Eulalie	15.67	21.64	10.45	17.16	4.48	11.94	13.43	2.24	9.70	134
Lenore	18.77	22.33	12.30	12.94	4.21	14.24	13.92	2.59	10.68	309
The Raven	21.85	21.39	17.82	11.33	5.94	11.70	11.43	2.47	7.13	1094
A Description of a City Shower	26.77	26.77	10.04	9.45	3.15	14.96	11.81	1.38	7.09	508
A Satirical Elegy	23.35	17.62	19.82	10.57	3.08	12.33	13.66	3.08	5.29	227
The Beasts' Confession to the Priest	24.98	20.73	16.95	10.09	3.91	11.94	11.39	1.24	7.00	1457
The Day of Judgement	23.33	25.33	18.67	10.00	2.67	11.33	8.00	2.67	5.33	150
The Puppet Show	26.98	29.55	7.49	8.35	2.57	13.92	12.21	1.28	10.06	467
Athanasia	21.82	26.06	9.49	14.55	1.41	16.36	8.89	2.02	10.71	495
Requiescat	25.00	26.19	20.24	13.10	4.76	10.71	5.95	0.00	3.57	84
The Dole of the King's Daughter	18.43	25.35	9.68	16.59	0.00	13.36	11.06	5.53	11.52	217
The Grave of Shelley	14.78	29.57	6.09	20.00	3.48	17.39	9.57	1.74	13.92	115
The Harlot's House	25.70	24.30	7.01	14.02	2.80	15.42	7.01	0.00	14.95	214

Table 1: Initial Analysis of Selected Works

described in Section 2. The values of the eigenvalues in decreasing order are as follows:

$$(2.94, 1.87, 1.22, 1.04, .93, .81, .50, .41, .25, .04)$$

By the common practice described in Section 2, the number of factors to be extracted should make up 80% of the variance. In the case of this analysis, five factors satisfy this condition. Table 3 shows the loadings of each of the variables to obtain each of the five factors. These loadings represent the coeffecients for the linear combination of each part of speech.

	Verb	Noun	Pron	Adj.	Adv.	Prep.	Conj.	Interj.	Art.	WrdCnt
Verb	1.00	-0.18	0.35	-0.59	-0.07	-0.17	0.04	-0.16	-0.50	0.08
Noun	-0.18	1.00	-0.54	0.13	-0.46	0.31	-0.58	-0.25	0.17	-0.02
Pron	0.35	-0.54	1.00	-0.46	0.33	-0.39	-0.08	-0.05	-0.60	-0.01
Adj.	-0.59	0.13	-0.46	1.00	-0.31	0.20	-0.15	0.20	0.24	0.04
Adv.	-0.07	-0.46	0.33	-0.31	1.00	-0.34	0.40	-0.02	-0.02	-0.01
Prep.	-0.17	0.31	-0.39	0.20	-0.34	1.00	-0.03	-0.21	0.08	-0.15
Conj.	0.04	-0.58	-0.08	-0.15	0.40	-0.03	1.00	0.24	0.01	-0.08
Interj.	-0.16	-0.26	-0.05	0.20	-0.02	-0.21	0.24	1.00	0.02	0.00
Art.	-0.50	0.17	-0.60	0.24	-0.02	0.08	0.01	0.02	1.00	-0.13
WrdCnt	0.08	-0.02	-0.01	0.04	-0.01	-0.15	-0.08	0.00	-0.13	1.00

Table 2: Correlation Matrix

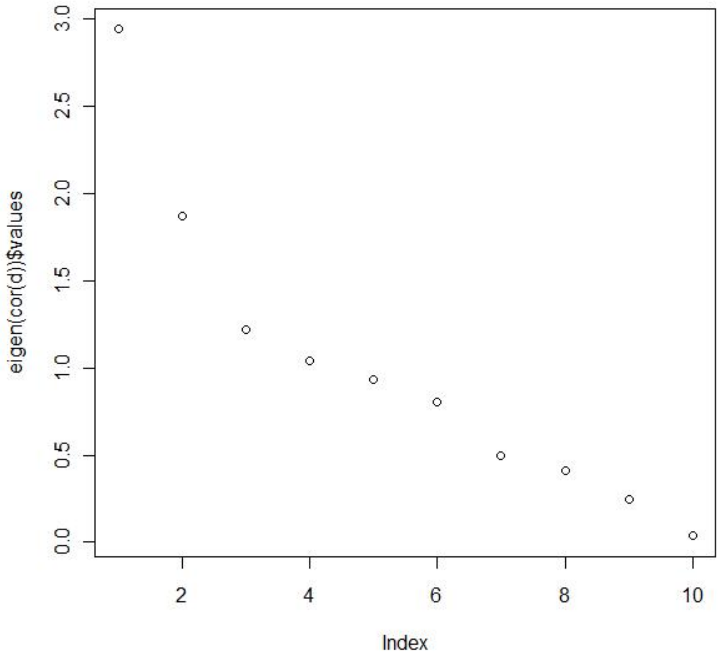

Figure 2: Scree Plot of Eigenvalues of Correlation Matrix

Now that we have the loadings, we can figure out what each factor represents. In the following discussions of each factor, the less prevalent loadings of each factor have been removed to observe the actions of the more prevalent loadings. We consider a non-salient (small) loading

	Factor1	Factor2	Factor3	Factor4	Factor5
Verb	-0.202		-0.923	-0.221	0.233
Noun	0.687	-0.716			
Pron	-0.850		-0.126	-0.470	-0.182
Adj	0.235		0.691	0.145	0.565
Adv	-0.305	0.404			-0.442
Prep	0.427		0.111		
Conj		0.901	-0.115		-0.161
Interj		0.316	0.210		0.143
Art	0.186		0.229	0.937	-0.171
WrdCnt				-0.107	0.176

Table 3: Factor Loadings

as any λ where $|\lambda| < .35$ (Biber, 87). In the tables for each factor, any field with a "plus" or "minus" indicates a non-salient loading that is positive or negative, respectively. Any field left blank indicates the loading of the variable is negligible. Each factor is given a name based on the defining salient loadings.

Factor 1 - Noun Set

	Verb	Noun	Pron	Adj	Adv	Prep	Conj	Interj	Art	WrdCnt
Factor 1	-	0.687	-0.850	+	-	0.427			+	

The most prevalent factor is rather intuitive. What it tells us is that there is a high negative correlation between the presence of nouns and pronouns. Considering the role of a pronoun is to replace a noun, this result makes sense in a poem that uses very few unique nouns, pronouns will be used often to prevent repetition. On the other hand, a poem that is rich in unique nouns, pronouns are not used as often to keep from confusing the reader. The positive loading of the prepositions indicates that prepositions are associated more with nouns rather than pronouns.

Factor 2 - Continuation

	Verb	Noun	Pron	Adj	Adv	Prep	Conj	Interj	Art	WrdCnt
Factor 2		-0.716			0.404		0.901	+		

Factor 2 indicates a negative correlation between conjunctions and nouns but a positive correlation between conjunctions and adverbs. One purpose of conjunctions is to combine phrases or clauses, essentially continuing a sentence. By continuing a sentence with a conjunction, the subject noun has already been established and therefore need not be repeated. In this case, the conjunction could prevent another full sentence from forming and could also prevent the use of another noun. The positive correlation between conjunctions and adverbs could indicate the reason a sentence is extended is to better describe the action.

Factor 3 - Action and Adjectives

	Verb	Noun	Pron	Adj	Adv	Prep	Conj	Interj	Art	WrdCnt
Factor 3	-0.923		-	0.691		+	-	+	+	

Factor 3 shows that if a poem has very few verbs, there will be better presence of adjectives. This factor seems to split poems into two categories: those that tell a story and those that "paint a picture." Many more adjectives are used in more descriptive poems but there is little action, so there is less of a need for verbs. However, a poem that tells more of a story requires more verbs but lacks description of the subjects.

Factor 4 - Common Noun Replacement

	Verb	Noun	Pron	Adj	Adv	Prep	Conj	Interj	Art	WrdCnt
Factor 4	-		-0.470	+					0.937	-

Factor 4 has been named "Common Noun Replacement" for the implications of its loadings and is another somewhat intuitive factor. The purpose of an article is to identify a noun, for instance: "the box." However, when "the box" is replaced by "it," the identifying article is lost. Thus, it seems the more articles in a particular piece, the less likely nouns are replaced by pronouns.

Factor 5 - Description

	Verb	Noun	Pron	Adj	Adv	Prep	Conj	Interj	Art	WrdCnt
Factor 5	+		-	.565	-0.442		-	+	-	+

Factor 5 shows a negative correlation between adjectives and adverbs. This could refer back to Factor 3 with poems being split between action and description. The lack of description of action in a poem could lead to a better physical description of what something's characteristics. A poem with a high presence of factor 5 would probably be very descriptive with location and characters but without embellishing any action.

7 Results

Appendix B gives gives the factor scores for each author by factor. A poem with a high score will have a higher . Studying the graphs, there does not seem to be much variance between any of the authors with the exception of Oscar Wilde. Based on the graphs, his works exhibit a higher presence of each factor as as compared to the other authors, only with the exception of factor 2. Also, in each of the graphs, Oscar Wilde displays a significant variation with at least one author. This could be explained by stylistic differences or even the time period in which he wrote along with his nationality. To ascertain such conjectures, further analysis on a broader scale would have to be done.

8 Conclusion and Continuation

Based on the results, nothing conclusive can be drawn from the analysis. All of the factors seem intuitive from the standpoint of using the language in a normal fashion. As it stands, this factor analysis has uncovered nothing concrete about poetry as a whole.

The *Mathematica* code written for this study streamlined the counting and cataloguing of each word. Even the longest text in this study of about 1500 words took less than a minute to run through the code. The utility of this code could make an expanded analyis tractable.

In particular, an expansion what has already been established in this study would include more poets and more works from each poet to create a base. It would then look at particular types of poetry to compare against the base to find out how each type of poetry contributes to the overall results. From there, this analysis could be applied to another genre.

Poetry is a highly individualized art; however, it is subject to poetic constraints. An expanded factor analysis could give a window into the functionality of the tools poets use to ply their craft.

A Descriptive Statistics for the Corpus as a Whole (Biber, 77-78)

Linguistic Feature	Mean	Minimum Value	Maximum Value	Range	Standard Deviation
past tense	40.1	0.0	119.0	119.0	30.4
perfect aspect verbs	8.6	0.0	40.0	40.0	5.2
present tense	77.7	12.0	182.0	170.0	34.3
place adverbials	3.1	0.0	24.0	24.0	3.4
time adverbials	5.2	0.0	24.0	24.0	3.5
first person pronouns	27.2	0.0	122.0	122.0	26.1
second person pronouns	9.9	0.0	72.0	72.0	13.8
third person pronouns	29.9	0.0	124.0	124.0	22.5
pronoun IT	10.3	0.0	47.0	47.0	7.1
deomonstrative pronouns	4.6	0.0	30.0	30.0	4.8
indefinite pronouns	1.4	0.0	13.0	13.0	2.0
DO as pro-verb	3.0	0.0	22.0	22.0	3.5
WH question	0.2	0.0	4.0	4.0	0.6
nominalizations	19.9	0.0	71.0	71.0	14.4
gerunds	7.0	0.0	23.0	23.0	3.8
nouns	180.5	84.0	298.0	214.0	35.6
agentless passives	9.6	0.0	38.0	38.0	6.6
BY passives	0.8	0.0	8.0	8.0	1.3
BE as main verb	28.3	7.0	72.0	65.0	9.5
existential THERE	2.2	0.0	11.0	11.0	1.8
THAT verb complements	3.3	0.0	20.0	20.0	2.9
THAT adj. complements	0.3	0.0	3.0	3.0	0.6
WH clauses	0.6	0.0	7.0	7.0	1.0
infinitives	14.9	1.0	36.0	35.0	5.6
present participial clauses	1.0	0.0	11.0	11.0	1.7
past participial clauses	0.1	0.0	3.0	3.0	0.4
past prt. WHIZ deletions	2.5	0.0	21.0	21.0	3.1
present prt. WHIZ deletions	1.6	0.0	11.0	11.0	1.8
THAT relatives: subj position	0.4	0.0	7.0	7.0	0.8
THAT relatices: obj. position	0.8	0.0	7.0	7.0	1.1
WH relatives: subj. position	2.1	0.0	15.0	15.0	2.0
WH relatives: obj. position	1.4	0.0	9.0	9.0	1.7
WH relatives: pied pipes	0.7	0.0	7.0	7.0	1.1
sentence relatives	0.1	0.0	3.0	3.0	0.4
adv. subordinator - cause	1.1	0.0	11.0	11.0	1.7
adv. sub. - concession	0.5	0.0	5.0	5.0	0.8
adv. sub. - condition	2.5	0.0	13.0	13.0	2.2
adv. sub. - other	1.0	0.0	6.0	6.0	1.1
prepositions	110.5	50.0	109.0	159.0	25.4
attributive adjectives	60.7	16.0	115.0	99.0	18.8
predicative adjectives	4.7	0.0	19.0	19.0	2.6
adverbs	65.6	22.0	125.0	103.0	17.6
type/token ratio	51.1	35.0	64.0	29.0	5.2
word length	4.5	3.7	5.3	1.6	0.4
conjuncts	1.2	0.0	12.0	12.0	1.6
downtoners	2.0	0.0	10.0	10.0	1.6
hedges	0.6	0.0	10.0	10.0	1.3
amplifiers	2.7	0.0	14.0	14.0	2.6
emphatics	6.3	0.0	22.0	22.0	4.2
discourse particles	1.2	0.0	15.0	15.0	2.3
demonstratives	9.9	0.0	22.0	22.0	4.2
possibility modals	5.8	0.0	21.0	21.0	3.5
necessity modals	2.1	0.0	13.0	13.0	2.1
predicitve modals	5.6	0.0	30.0	30.0	4.2
public verbs	7.7	0.0	40.0	40.0	5.4
private verbs	18.0	1.0	54.0	53.0	10.4
suasive verbs	2.9	0.0	36.0	36.0	3.1
SEEM/APPEAR	0.8	0.0	6.0	6.0	1.0
contractions	13.5	0.0	89.0	89.0	18.6
THAT deletion	3.1	0.0	24.0	24.0	4.1
stranded prepositions	2.0	0.0	23.0	23.0	2.7
split infinitives	0.0	0.0	1.0	1.0	0.0
split auxiliaries	5.5	0.0	15.0	15.0	2.5
phrasal coordination	3.4	0.0	12.0	12.0	2.7
non-phrasal coordination	4.5	0.0	44.0	44.0	4.8
synthetic negation	1.7	0.0	8.0	8.0	1.6
analytic negation	8.5	0.0	32.0	32.0	6.1

B Factor Graphs

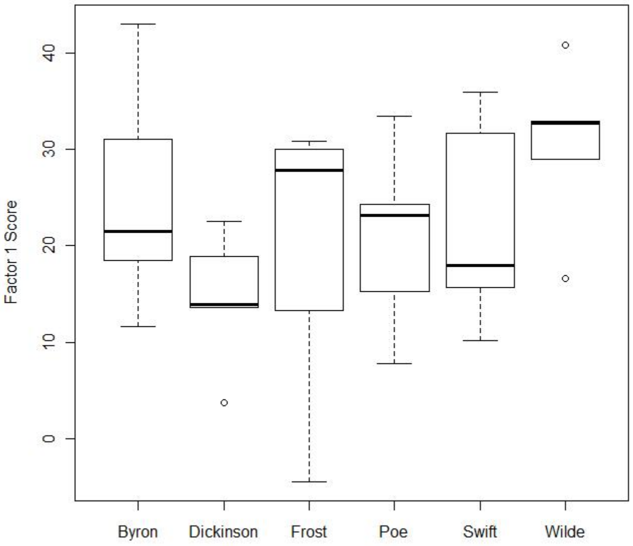

Figure 3: Factor 1 Scores by Author

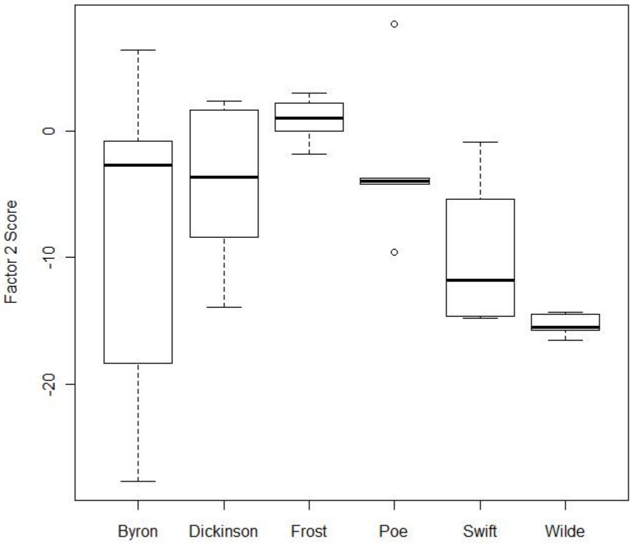

Figure 4: Factor 2 Scores by Author

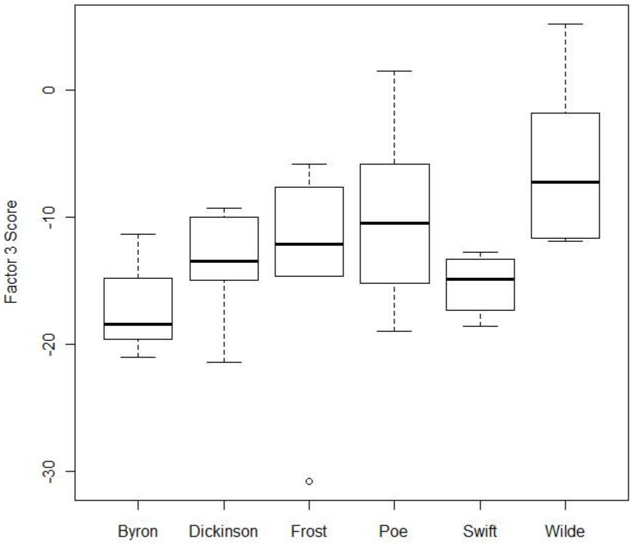

Figure 5: Factor 3 Scores by Author

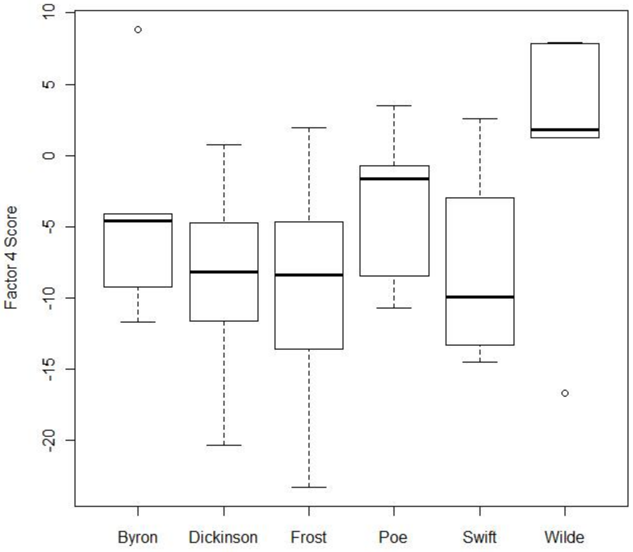

Figure 6: Factor 4 Scores by Author

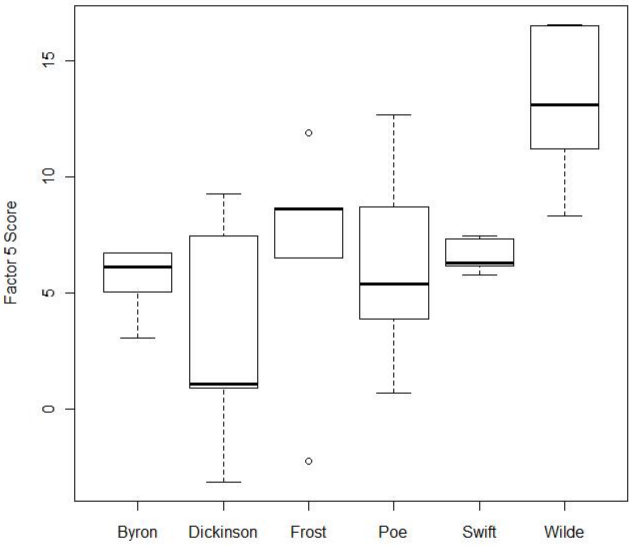

Figure 7: Factor 5 Scores by Author

C Full *Mathematica* Code

This creates the string to break apart

```
(*This is where user inputs text.  It is important to remove all line breaks.*)
instring = "Input text here.";

(*This changes all letters to lowercase and removes all symbols and numbers*)
stringlist =
  ToLowerCase[
   StringSplit[instring, Characters["():,;.!?’-\[LongDash]1234567890 "] ..]];
(*Calculates a final word count*)
Length[stringlist]
```

This creates the lists of Parts of Speech including all needed intersections and unions

```
(*Creates master lists of each part of speech to cross-reference in the \
future*)
verb = StringSplit[ToString[Import[
     "\Users\owner\Documents\MATLAB\MASTERS\Parts of Speech\verb.txt"]], \
{Whitespace}];
noun = StringSplit[ToString[Import[
     "\Users\owner\Documents\MATLAB\MASTERS\Parts of \
Speech\common_noun.txt"]], {Whitespace}];
pronoun = StringSplit[ToString[Import[
     "\Users\owner\Documents\MATLAB\MASTERS\Parts of Speech\pronoun.txt"]], \
{Whitespace}];
adjective = StringSplit[ToString[Import[
     "\Users\owner\Documents\MATLAB\MASTERS\Parts of Speech\adjective.txt"]], \
{Whitespace}];
adverb = StringSplit[ToString[Import[
     "\Users\owner\Documents\MATLAB\MASTERS\Parts of Speech\adverb.txt"]], \
{Whitespace}];
preposition = StringSplit[ToString[Import[
     "\Users\owner\Documents\MATLAB\MASTERS\Parts of \
Speech\preposition.txt"]], {Whitespace}];
conjunction = StringSplit[ToString[Import[
     "\Users\owner\Documents\MATLAB\MASTERS\Parts of \
Speech\conjunction.txt"]], {Whitespace}];
interjection = StringSplit[ToString[Import[
     "\Users\owner\Documents\MATLAB\MASTERS\Parts of \
Speech\interjection.txt"]], {Whitespace}];
articles = StringSplit[ToString[Import[
     "\Users\owner\Documents\MATLAB\MASTERS\Parts of Speech\articles.txt"]], \
{Whitespace}];


(*Creates lists for words with multiple parts of speech \begin*)
verbnoun = Intersection[verb, noun]; verbpronoun =
 Intersection[verb, pronoun]; verbadj = Intersection[verb, adjective];
verbadverb = Intersection[verb, adverb]; verbprep =
 Intersection[verb, preposition]; verbconj = Intersection[verb, conjunction];
```

```
verbinter = Intersection[verb, interjection]; verbart =
 Intersection[verb, articles];

nounpronoun = Intersection[noun, pronoun]; nounadj =
 Intersection[noun, adjective]; nounadverb = Intersection[noun, adverb];
nounprep = Intersection[noun, preposition]; nounconj =
 Intersection[noun, conjunction]; nouninter =
 Intersection[noun, interjection]; nounart = Intersection[noun, articles];

pronounadj = Intersection[pronoun, adjective]; pronounadverb =
 Intersection[pronoun, adverb]; pronounprep =
 Intersection[pronoun, preposition];
pronounconj = Intersection[pronoun, conjunction]; pronouninter =
 Intersection[pronoun, interjection]; pronounart =
 Intersection[pronoun, articles];

adjadverb = Intersection[adjective, adverb]; adjprep =
 Intersection[adjective, preposition]; adjconj =
 Intersection[adjective, conjunction];
adjinter = Intersection[adjective, interjection]; adjart =
 Intersection[adjective, articles];

adverbprep = Intersection[adverb, preposition]; adverbconj =
 Intersection[adverb, conjunction]; adverbinter =
 Intersection[adverb, interjection]; adverbart =
 Intersection[adverb, articles];
```

```
prepconj = Intersection[preposition, conjunction]; prepinter =
 Intersection[preposition, interjection]; prepart =
 Intersection[preposition, articles];

interconj = Intersection[interjection, conjunction]; interart =
 Intersection[interjection, articles];

conjart = Intersection[conjunction, articles];
(*Creates lists for words with multiple parts of speech \end*)

(*Creates a list of all words with multiple parts of speech to crosscheck \
with {stringlist}*)
twoface = Union[verbnoun, verbpronoun, verbadj, verbadverb, verbprep,
   verbconj, verbinter, nounpronoun, nounadj, nounadverb, nounprep, nounconj,
   nouninter, pronounadj, pronounadverb, pronounprep, pronounconj,
   pronouninter, adjadverb, adjprep, adjconj, adjinter, adverbprep,
   adverbconj, adverbinter, prepconj, prepinter, verbart, nounart, pronounart,
    adjart, adverbart, prepart, interconj, interart, conjart];
```

This crosschecks each Parts of Speech list to the string

```
VerbList =
 Flatten[Table[Intersection[{stringlist[[i]]}, verb], {i, Length[stringlist]}]]
Length[VerbList]
```

```
NounList =
 Flatten[Table[Intersection[{stringlist[[i]]}, noun], {i, Length[stringlist]}]]
Length[NounList]

PronounList =
 Flatten[Table[Intersection[{stringlist[[i]]}, pronoun], {i, Length[stringlist]}]]
Length[PronounList]

AdjectiveList =
 Flatten[Table[Intersection[{stringlist[[i]]}, adjective], {i, Length[stringlist]}]]
Length[AdjectiveList]

AdverbList =
 Flatten[Table[Intersection[{stringlist[[i]]}, adverb], {i, Length[stringlist]}]]
Length[AdverbList]

PrepositionList =
 Flatten[Table[Intersection[{stringlist[[i]]}, preposition], {i, Length[stringlist]}]]
Length[PrepositionList]

ConjunctionList =
 Flatten[Table[Intersection[{stringlist[[i]]}, conjunction], {i, Length[stringlist]}]]
Length[ConjunctionList]

InterjectionList =
 Flatten[Table[Intersection[{stringlist[[i]]}, interjection], {i, Length[stringlist]}]]
```

```
Length[InterjectionList]

ArticleList =
 Flatten[Table[Intersection[{stringlist[[i]]}, articles], {i, Length[stringlist]}]]
Length[ArticleList]

dent = Flatten[Table[Intersection[{stringlist[[i]]}, twoface], {i, Length[stringlist]}]]
Length[dent]

(*Creates a full list of all words from {stringlist} identified with a part \
of speech*)
masterlist =
  Union[VerbList, NounList, PronounList, AdjectiveList, AdverbList,
   PrepositionList, ConjunctionList, InterjectionList, ArticleList];

(*Creates a list of all words from {stringlist} that have not been identified \
with a part of speech*)
OtherList = Complement[stringlist, masterlist]
```

D Full *R* Code

```
rawd=read.csv(file=’C:/Users/owner/Documents/MATLAB/MASTERS/RData.csv’)
d=rawd[,-11]
cor(d)
eigen(cor(d))
plot(eigen(cor(d))$values)
factanal(d, factors=2)
factanal(d, factors=3)
factanal(d, factors=4)
factanal(d, factors=5)
factanal(d, factors=6)
m=factanal(d, factors=5)

tmp1=numeric(30)
for(i in 1:30){
tmp1[i]=sum(d[i,]*matrix(loadings(m)[,1],ncol=1))}

tmp2=numeric(30)
for(i in 1:30){
tmp2[i]=sum(d[i,]*matrix(loadings(m)[,2],ncol=1))}

tmp3=numeric(30)
for(i in 1:30){
tmp3[i]=sum(d[i,]*matrix(loadings(m)[,3],ncol=1))}
```

```
tmp4=numeric(30)

for(i in 1:30){

tmp4[i]=sum(d[i,]*matrix(loadings(m)[,4],ncol=1))}


tmp5=numeric(30)

for(i in 1:30){

tmp5[i]=sum(d[i,]*matrix(loadings(m)[,5],ncol=1))}


plot(tmp1,tmp2)

plot(tmp1,tmp3)

plot(tmp1,tmp4)

plot(tmp1,tmp5)


plot(tmp2,tmp3)

plot(tmp2,tmp4)

plot(tmp2,tmp5)


plot(tmp3,tmp4)

plot(tmp3,tmp5)


plot(tmp4,tmp5)
```

E Bibliography

Biber, D. (1988). *Variation Across Speech and Writing.* Cambridge, Great Britain: Cambridge University Press.

Davies, Mark. (2008-) The Corpus of Contemporary American English: 425 million words, 1990-present. Available online at http://corpus.byu.edu/coca/.

Hoffstein, J., Pipher, J., & Silverman, J. H. (2000). *An Introduction to Mathematical Cryptography.* New York, NY: Springer Science+Business Media, LLC.

Mardia, K. V., Kent, J. T., & Bibby, B. B. (2003). *Multivariate Analysis.* San Diego, CA: Academic Press.

Miles, J. (1960). *Renaissance, Eighteenth-Century, and Modern Language in English Poetry.* Los Angeles, CA: Cambridge University Press.

Rencher, A. C. (2002). *Methods of Multivariate Analysis.* New York, NY: John Wiley & Sons, Inc.

Singh, S. (1999). *The Code Book.* New York, NY: Anchor Books.