Development of Quantitative Summary of Plays and Evaluation with *Hamlet*

Nathan Bickel

2/25/2022

Introduction

In the plays of an author like Shakespeare, it is usually the case that a character speaking more means they are more important, and that a certain word being used more than normal in a particular play means that the concept being represented by that word is important. As such, it should be possible to quantitatively survey a play, even without any qualitative knowledge of it, and gain some insight into the characters that are important, the ideas represented in the play, and how the characters relate to those ideas. This paper attempts to develop a rudimentary framework to choose important words and characters from a given play and display connections between them, and then to evaluate the results of the generated summary for one of Shakespeare's most popular plays, *Hamlet*.

Data and Methods

Three different matrices will be used as the foundation for this analysis. The first is P, a 26952×37 matrix where $P_{i,j}$ represents the frequency a given word i occurs in a given play j. The second is S, a 26952×31066 matrix where $S_{i,j}$ represents the frequency a given word i occurs in a given speech j. The last is C, a 26952×1449 matrix where $C_{i,j}$ represents the frequency a given word i is said by a given character j.

The following setup is used to define slices of P and C corresponding to the play chosen. For this report, $\mathbf{play}$ = "Ham", but a simple change to something like $\mathbf{play}$ = "Mac" or $\mathbf{play}$ = "Rom" will yield data based entirely on *Macbeth* or *Romeo and Juliet*, respectively, with the same pattern working for the other 34 plays. Then, $\mathbf{S\_Ham}$ is the matrix that contains all the columns with the speeches from $\mathbf{play}$ but not the other 36, and $\mathbf{C\_Ham}$ is the matrix that contains all the columns with the (single) characters from $\mathbf{play}$.

```
play = "MND"
play_id = which(play_metadata$ID==play)
S_Ham = S[,which(speech_metadata$PLAY==play)]
C_Ham = C[,grep(paste("^(#[\\S]+",play,")$",sep=""),colnames(C),perl=TRUE)]
```

The first way the data were explored was by creating a vector called $\mathbf{impt\_chars}$ that includes characters who speak an unusual amount. This threshold was set by calculating the average number of words for a character in the play to speak, and characters who speak more than the average are included in the list. While the distribution of words spoken will certainly be quite right-skewed, this was actually viewed as advantageous because the important characters are the ones likely to be in the right "tail" of the distribution.

Next, another vector called $\mathbf{impt\_words}$ was created that includes words that appear to be over-represented and thus important in $\mathbf{play}$. To assist with this, a matrix called $\mathbf{ppmi\_P\_values}$ was created. This matrix's values were determined from P with the formula

$$\mathbf{ppmi\_P\_values}_{x,y} = \max\left(0, \log\frac{p(x,y)}{p(x)p(y)}\right).$$

Thus, each cell is a sort of ratio of the observed probability to the expected probability, so a high value means that the word appears much more often than would be expected if the play and word probabilities were independent, and a 0 if it is equal or lower than expected. There are four tests a word needs to pass to be included in $\mathbf{impt\_words}$: (1) the PPMI for the word is higher for $play$ than any other play (to find the words that are most represented in the play), (2) the PPMI is higher than 2 standard deviations of the PPMI over $\mathbf{ppmi\_P\_values}$ from 0 (to eliminate very common words that happen to show up the most in Hamlet), (3) the word appears at least $\mathbf{word\_threshold}$ (set to 5 in the code) times in $\mathbf{play}$ (to eliminate words that are only used very rarely in the corpus), (4) the play contains less than 50% of the tokens over the corpus (to eliminate words only relevant to the play, such as character names). In theory, this should give words that are relevant and somewhat important to the play, and give a sense for the ideas and themes.

To begin to analyze these lists, a bar chart was generated to determine which words from $\mathbf{impt\_words}$ correlate over $\mathbf{S\_Ham}$ together most. The words above $\mathbf{correlation\_threshold}$, set at 0.2, are displayed. For example, in *Hamlet*, “particular” and “star” have a correlation of around 0.25, which refers to the values of “star” and “particular” taken over $\mathbf{S\_Ham}$ with each column (the individual speeches) being compared.

Then, a new matrix called $\mathbf{ppmi\_C\_values}$ was generated using a subset of C: the word rows selected are from $\mathbf{impt\_words}$, and the character columns selected are from $\mathbf{impt\_chars}$. Then, the same method of PPMI was applied to the matrix, so a high value in $\mathbf{ppmi\_C\_values}$ means that a certain important character said a certain important word an unusual amount, given how much the character speaks and how much the word is used compared to the other important characters and words. Then, each word is printed with the character for whom the PPMI is the highest: this attempts to give an overview of which characters are associated with particular important concepts in $\mathbf{play}$.

Finally, a vector called $\mathbf{character\_words}$ was created. The idea is to pick a $\mathbf{character}$ from $\mathbf{impt\_chars}$ and place the words associated with them from $\mathbf{ppmi\_C\_values}$ in the greater context of the corpus. The first character from $\mathbf{impt\_char}$, Horatio, was used as example. Horatio has five words from the previous section, so for each of those 5 words, the $\mathbf{threshold}$ (set to 6) top words that are most correlated over C are printed. Thus, for the second word, “star”, the top 6 words (excluding “star”) are printed for which a character saying that word and saying “star” are mostly closely correlated, and this is repeated for the other four words Horatio is associated with.

Analysis

```
mean = mean(colSums(C_Ham))
impt_chars = which(colSums(C_Ham)>mean)
print(paste("Mean:",mean))
```

```
## [1] "Mean: 571.88"
```

```
print(names(impt_chars))
```

```
##  [1] "#Theseus_MND"         "#Hermia_MND"          "#Demetrius_MND"
##  [4] "#Lysander_MND"        "#Helena_MND"          "#Quince_MND"
##  [7] "#Bottom_MND"          "#RobinGoodfellow_MND" "#Oberon_MND"
## [10] "#Titania_MND"
```

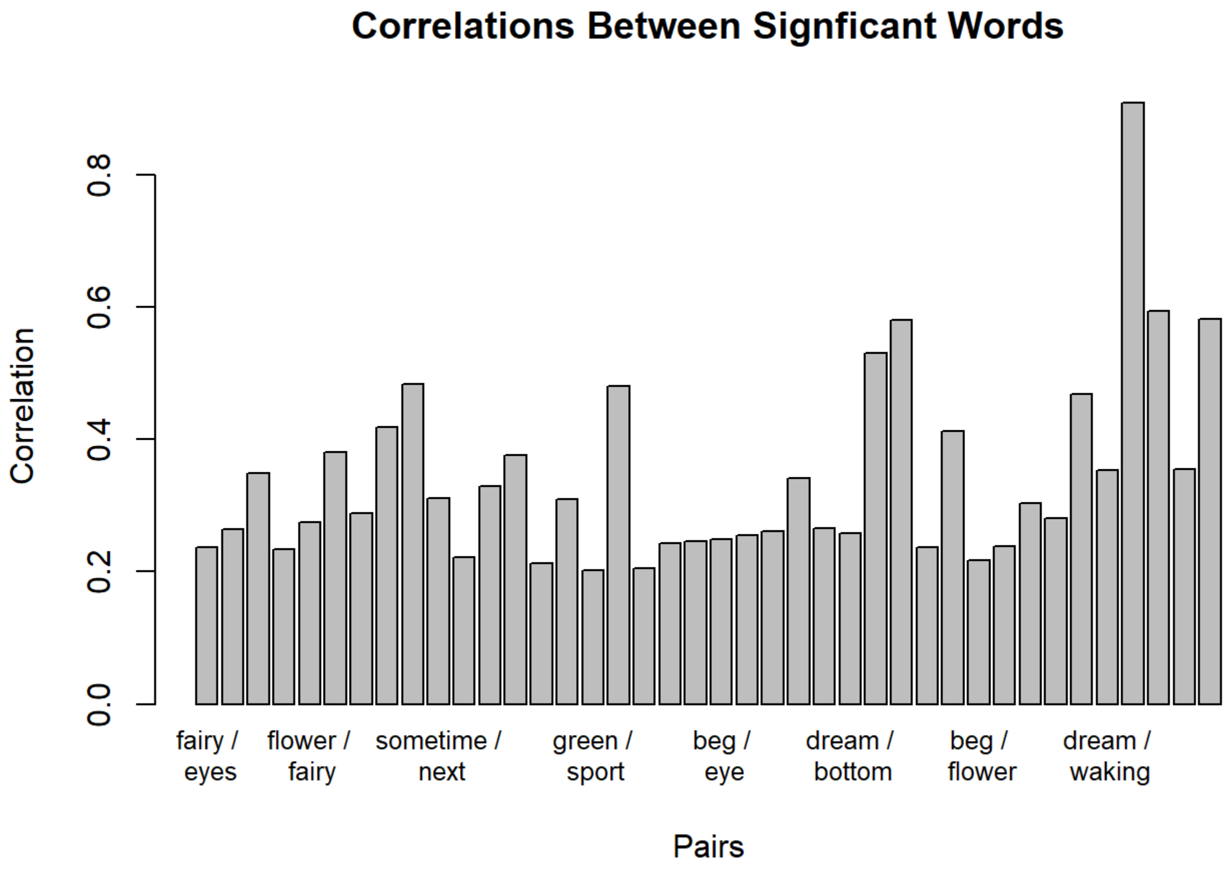
This gives a list of the important characters, and it seems to have worked fairly well for *Hamlet*. It would certainly be hard to argue that any of these characters are not important, and while there are characters that have a significant impact on the plot missing (like the ghost and the players), this seems to be most of the core group of characters that play focuses on.

```
ppmi_P_values = ppmi(P)
st_dev = sum(ppmi_P_values^2)/length(ppmi_P_values)
word_threshold = 5
impt_words = c()
for (r in 1:nrow(P)) {
    if (which.max(ppmi_P_values[r,])==play_id & ppmi_P_values[r,play_id]>2*st_dev & P[r,play_id]
> word_threshold & 2*P[r,play_id] < sum(P[r,])) {
        impt_words = append(impt_words,rownames(P)[r])
    }
}
print(impt_words)
```

```
##  [1] "eyes"       "meet"       "fairy"      "next"       "sleeping"  
##  [6] "moon"       "prologue"   "low"        "lion"       "monsieur"  
## [11] "wood"       "sport"      "through"    "eye"        "either"    
## [16] "roar"       "bottom"     "methinks"   "loves"      "robin"     
## [21] "voice"      "flower"     "play"       "green"      "kill"      
## [26] "beard"      "sometime"   "beg"        "brief"      "discretion"
## [31] "waking"     "methought"  "wall"       "choice"     "dream"     
## [36] "peter"      "lovers"
```

This generates the list of important words (in no particular order). It appears to have had mixed results—a number of these words are not particularly relevant to the play, such as “exit” and “behind” (which may have been included from stage directions) and “aught”, but others are quite relevant. “Question” appears in probably the most famous line from the play (and maybe even from Shakespeare’s plays as a whole), “To be or not to be—that is the the question” (3.1.64), and it also characterizes much of the play with both Hamlet and Laertes’ internal debates of how to deal with the murders of their respective fathers. “Memory” is a theme running throughout with many of the characters’ grief, and “madness” is a word heavily associated with Hamlet and, after Polonius’s death, Ophelia. Finally, “wholesome” is often used to contrast with the current situation in the play—for example, Lucianus in the production says “On wholesome life immediately. / *Pours the poison in his ears*” (3.2.286-287) as he kills Gonzago. Thus, while this list of words certainly isn’t perfect, it does find some words that represent quite important themes in *Hamlet*, and one could conceivably get a sense from it for some of the ideas in the play without having read it.

```
correlations = matrix(,nrow=length(impt_words),ncol=length(impt_words))
rownames(correlations) = colnames(correlations) = impt_words
for (r in 1:nrow(correlations)) {
     for (c in 1:ncol(correlations)) {
          correlations[r,c] = correlation(S_Ham[impt_words[r],],S_Ham[impt_words[c],])
          if (r==c) {
              correlations[r,c] = 0
          }
     }
}
correlation_threshold=0.2
high_corr_values = unique(correlations[which(correlations>correlation_threshold)])
for (i in 1:length(high_corr_values)) {
  index = which(abs(correlations-high_corr_values[i])<0.0001)[1]
  name = paste(rownames(correlations)[arrayInd(index, dim(correlations))[1]],
               "/\n",rownames(correlations)[arrayInd(index, dim(correlations))[2]])
  names(high_corr_values)[i] = name
}
barplot(high_corr_values, main="Correlations Between Signficant Words",xlab="Pairs",ylab="Correlation",cex.names=.8)
```



The bar chart shows which words from **impt_words** are most correlated throughout the speeches in Hamlet. Somewhat amusingly, the highest by far is “arras and behind”, which comes from Polonius saying “Behind the arras I’ll convey myself” (3.3.30) and then a number of stage directions indicating this—this indicates that arras and behind essentially only appear in this context, which suggests that these are not actually good choices to include in **impt_words**. It's possible that "particular and star" are connected (there does seem to be a connection in the play with stars and fortune), and the connection between "phrase" and "effect" could have something to do with Hamlet's writing to Ophelia and the effect it has on here. However, the connections seems shaky at best, which is supported by the fact that none of the correlations are very high. It may be more effective to compare correlations over whole scenes, as **impt_words** is small enough that is unlikely two of them will show up together in the same speeches particularly often.

```
ppmi_C_values = ppmi(C_Ham[impt_words,impt_chars])
chars_words = vector(mode="list", length=length(impt_chars))
names(chars_words) = names(impt_chars)
for (i in 1:length(impt_words)) {
    index = which.max(ppmi_C_values[i,])
    chars_words[[index]] = append(chars_words[[index]],impt_words[i])
}
print(chars_words)
```

```
## $`#Theseus_MND`
## [1] "moon"       "either"     "methinks"   "brief"      "discretion"
## [6] "wall"       "lovers"    
## 
## $`#Hermia_MND`
## [1] "low"
## 
## $`#Demetrius_MND`
## [1] "lion"  "wood"  "loves"
## 
## $`#Lysander_MND`
## [1] "through" "choice" 
## 
## $`#Helena_MND`
## [1] "eyes" "eye"  "kill"
## 
## $`#Quince_MND`
## [1] "meet"     "prologue" "bottom"   "play"    
## 
## $`#Bottom_MND`
## [1] "monsieur"  "roar"      "beard"     "methought" "dream"     "peter"    
## 
## $`#RobinGoodfellow_MND`
## [1] "fairy"    "sport"    "voice"    "sometime"
## 
## $`#Oberon_MND`
## [1] "next"   "robin"  "flower" "beg"   
## 
## $`#Titania_MND`
## [1] "sleeping" "green"    "waking"
```

This list places each word with the character for whom the PPMI is highest in **ppmi_C_values**. Thus, the idea is that Laertes would be most connected out of **impt_chars** with "question" and "brains", while Ophelia is most connected with "memory" and "speech". These results seem interesting—one would expect "question" to be more associated with Hamlet rather than Laertes. However, it's dubious whether question should actually be more associated with Laertes, because he only says the word once in the play and Hamlet says it 5 times. Additionally, Rosencrantz says it 3 times, so if he had made the cutoff for $\mathbf{impt\_chars}$ the word would certainly have been associated with him. This once again likely comes down to the issue of $C_{\mathbf{impt\_words},\mathbf{impt\_chars}}$ not having enough data for as many meaningful trends to emerge. PPMI also may not be the best choice here, because while Hamlet speaks a lot about many things, that should not necessarily disqualify him from being the most connected with some of these words.

However, one result that likely does hold is “star” being connected with Horatio. In the first scene, he describes the fall of Rome, saying “As stars with trains of fire and dews of blood, / Disasters in the sun; and the moist star, / Upon whose influence Neptune’s empire stands, / Was sick almost to doomsday with eclipse” (1.1.129-133). He is using this as an analogy for the misfortune signified by the arrival of the ghost, which turns out to be completely correct, seeing as the play ends with nearly everyone murdering each other. This discussion of fortune is described using the stars, so it seems fitting that the word “star” would be placed with Horatio. To further investigate this, Horatio is used as the character for the final section.

```r
character = 1;
depth = 6;
character_words = vector(mode="list",length=length(chars_words[[character]]))
names(character_words) = chars_words[[character]]
for (i in 1:length(character_words)) {
  correlation_char_vals = vector(mode="list", length=nrow(C))
  names(correlation_char_vals) = rownames(C)
  for (j in 1:nrow(C)) {
    correlation_char_vals[j] = correlation(C[names(character_words)[i],],C[j,])
  }
  character_words[[i]] = vector(mode="list",length=depth)
  character_words[[i]] = sort(unlist(correlation_char_vals),decreasing=T)[2:as.integer(1+depth)]
}
print(character_words)
```

```
## $moon
##      come        it       not        if        me       let 
## 0.4979325 0.4897281 0.4855539 0.4766068 0.4757698 0.4717631 
## 
## $either
##        or      that       but        in       not        to 
## 0.6054015 0.5803982 0.5791634 0.5698119 0.5697428 0.5685071 
## 
## $methinks
##        so      that        my        to       and       the 
## 0.5711507 0.5686899 0.5683372 0.5671018 0.5629920 0.5621305 
## 
## $brief
##      that       but        to       may    myself        by 
## 0.4726397 0.4669892 0.4637056 0.4564364 0.4517802 0.4511312 
## 
## $discretion
##       egeus philostrate comprehends   withering    chanting crook-kneed 
##   0.5850346   0.5850346   0.5850346   0.5850346   0.5850346   0.5850346 
## 
## $wall
##    cranny roughcast     snout  crannied     chink bergomask 
## 0.6435315 0.6435315 0.6296296 0.5465109 0.5139128 0.4491034 
## 
## $lovers
##    forester   hippolyta philostrate comprehends   withering    chanting 
##   0.5414084   0.4910815   0.4393405   0.4393405   0.4393405   0.4393405
```

This test attempts to see which words are most associated with the words from a character across the corpus of C —in this case, the character is Horatio, but $\mathbf{character}$ can be changed to another number and yield the words for the character chosen. Once again, the test seems to have mixed results: "'twere" appears almost completely meaningless, as it is only associated with connective words, and is thus probably not at all a good choice for $\mathbf{impt\_words}$. However, the results for "star", "sealed", and "custom" are more interesting. "Amities" makes sense for "star" because of the connection between fortune and relationships seen in Shakespeare (for example, "star-crossed lovers" in *Romeo and Juliet*), and "ponderous" because of the weight fortune is given. It's also interesting that "comply" and "compulsive" come up with "custom", as they imply a sense of dedication to the past rather than doing something for its own sake. Specific to *Hamlet*, in Act 5, Hamlet asks "Has this fellow no feeling of his business? He / sings in grave-making" (5.1.67-68), and Horatio responds "Custom hath made it in him a property of / easiness" (5.1.69-70). It's telling, then, that "yawn" is most correlated with "custom", as "yawn" is associated with boredom and reflects the desensitization to things done out of tradition that Horatio describes. Finally, with "sealed", it's rather unexpected that two of the most correlated words are characters names from *Hamlet* when this is taken across all 37 plays, and it confirms that "sealed" is a good candidate for $\mathbf{important\_words}$. Thus, while there is still data that isn't very useful, this was probably the most revealing of the three tests.

Conclusion

The methods used in this report are certainly rudimentary and produce a good deal of data that is tenuously or barely connected with *Hamlet*. However, it does also produce and point to interesting patterns. The connection with stars and fortune isn't an entirely obvious one, but the patterns shown throughout this analysis gave motivation to follow this thread through the play qualitatively. Importantly, it does so without any specifics in the code related to the play. Thus, the $\mathbf{play}$ variable in the first code block can be changed[1] and show an entirely different summary for a different play that should, in theory, reveal other patterns specific to that play. While some of the test leave much room for improvement, they offer a sort of summary of the themes of the play for someone who hasn’t read it, and a reasonable-jumping off point for deeper qualitative analysis.

---

[1] The reader is free to change $\mathbf{play}$ to something else and see the data it produces!↩