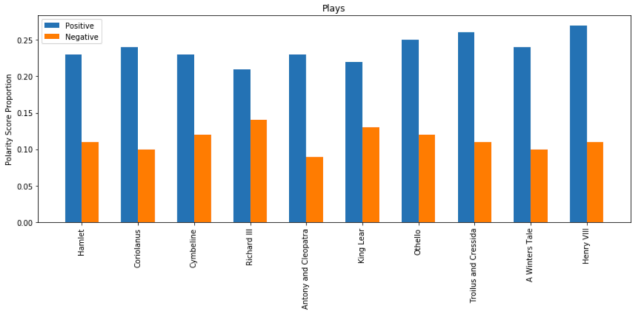
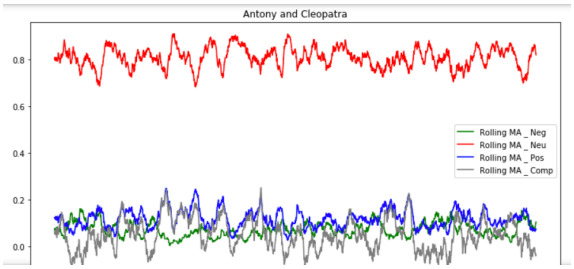
Student: Negri, Maria

In order to determine whether Shakespeare was a single genius or team of playwrights, we implemented several tools learned in class to inspect the master pieces.

To begin with, we decided to check whether there was any particular sentiment pattern both across each text and across all of the pieces. We applied Textblob to each of the lines and calculated the polarity score proportion of each sentiment (Positive, Negative, Neutral) to each piece. We can see the results pasted in a bar plot; the "Neutral" sentiment dominates the results and none of the texts exhibit a clear "Positive" or "Negative" tendency.

Afterwards, we performed a similar analysis using VADER sentiment analysis tool across each text. Once again, we see no clear pattern or tendency (please see the Jupyter code for the rest of the texts).

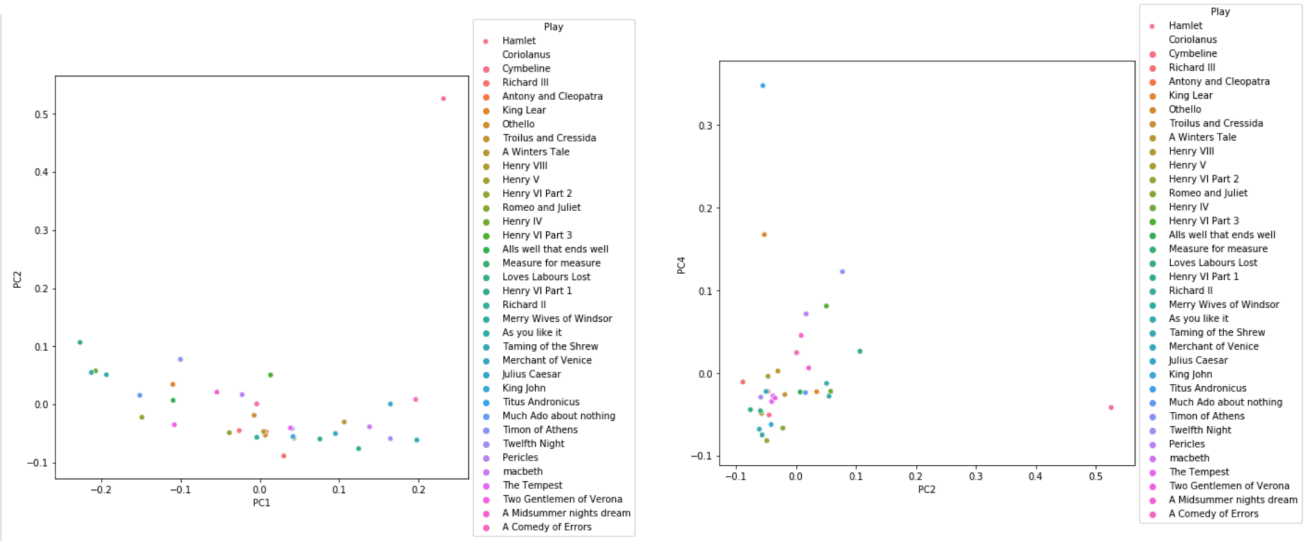




Since the first step of the analysis did not allow us to reach any conclusion, we decided to apply TFIDF (Term Frequency Inverse Document Frequency). This statistical technique will allow us to translate Shakespeare pieces into matrices by and labeling the most relevant words into numbers and creating a big matrix. This should allow us to detect the most distinctive words in each text and then run the cosine similarity analysis. The cosine analysis between two pieces will output a number closer to one when they are similar. As can see from the cosine_similarity matrix, the results show no clear no distinctive patters among texts.

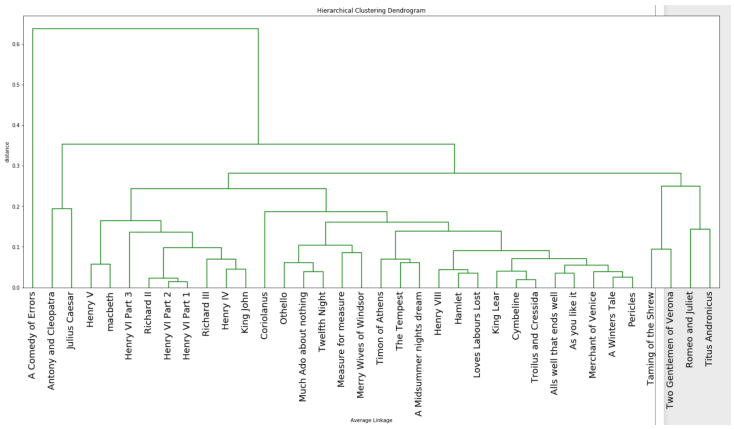
We continued our analysis by running PCA on the matrix that resulted from running TFIDF. PCA is a linear dimensionality reduction method that will allow us to build orthogonal projections that most of the variability in the matrix under study. By plotting the projections on the scatter plots, we were able to detect potential outliers. PCA analysis shows suggests that the following plays are outliers:

- A Comedy of Errors
- Julius Caesar
- Antony and Cleopatra
- Titus Andronicus



Therefore, we run K-Means to confront and confirm this evidence. K-means is an unsupervised learning method. It is used to look for patterns in data when there is no particular target feature, or dependent variable. K-means clustering is a simple and elegant approach for partitioning a data set into K distinct, non-overlapping clusters. K-means problem is solved using Lloyd's algorithm, which partitions the observations into K clusters such that the total within-cluster variation, summed over all K clusters, is as small as possible. Silhouette scores can be used to help evaluate the appropriate number of clusters that are truly in the data.

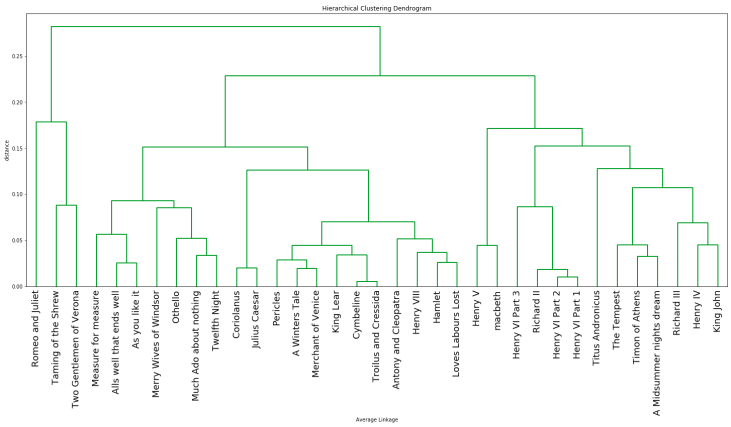
After running K-means on our dataset we found that number of clusters that reach the highest silhouette score (0.53) is three.



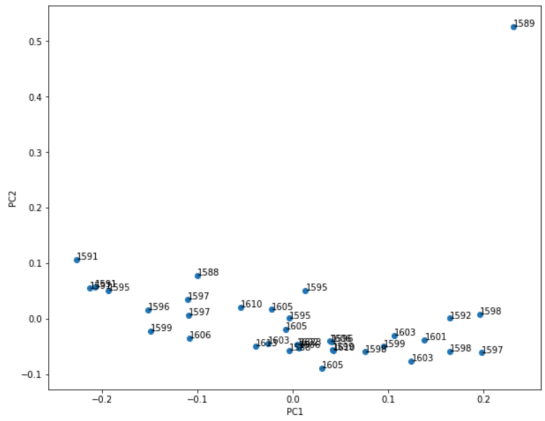
We then applied the Hierarchical analysis. The result of hierarchical clustering is a tree-based representation of the objects, which is also known as dendrogram. Each node represents a group.

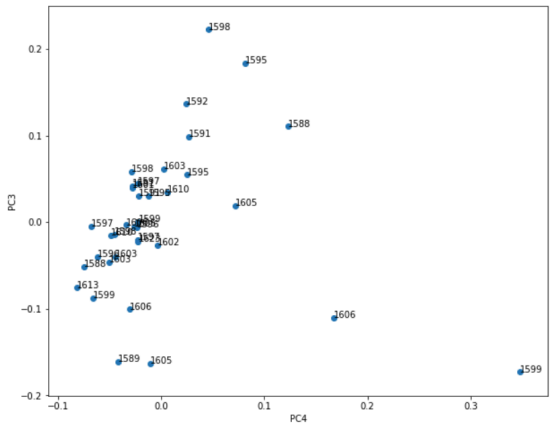
Both clustering techniques show similar results. We decided to explore further into these three clusters and see if there is any writing pattern relevant in each of them.

By looking at the dendrogram, we decided to re-run K-means Hierarchical analysis again but this time excluding the play called "A Comedy of Errors". Since K-Means reported the highest silhouette score with three clusters, we cut the H-tree by three. This second process allowed us to extract four clusters, which we will discuss further below.



Before doing that, we wanted to check whether there was any hint that these clusters were associated in time. To do this, we plotted the PCA projections and labeled each play using the year in which they were written according to Wikipedia. As we can see from the plots below, there does not seem to be any relation between the year in which the plays were published and the clusters we obtained above. However, it is clear that if it is true that Shakespeare was a single person, it is clear that he had a great talent since the majority of the plays were written in a short span of time.

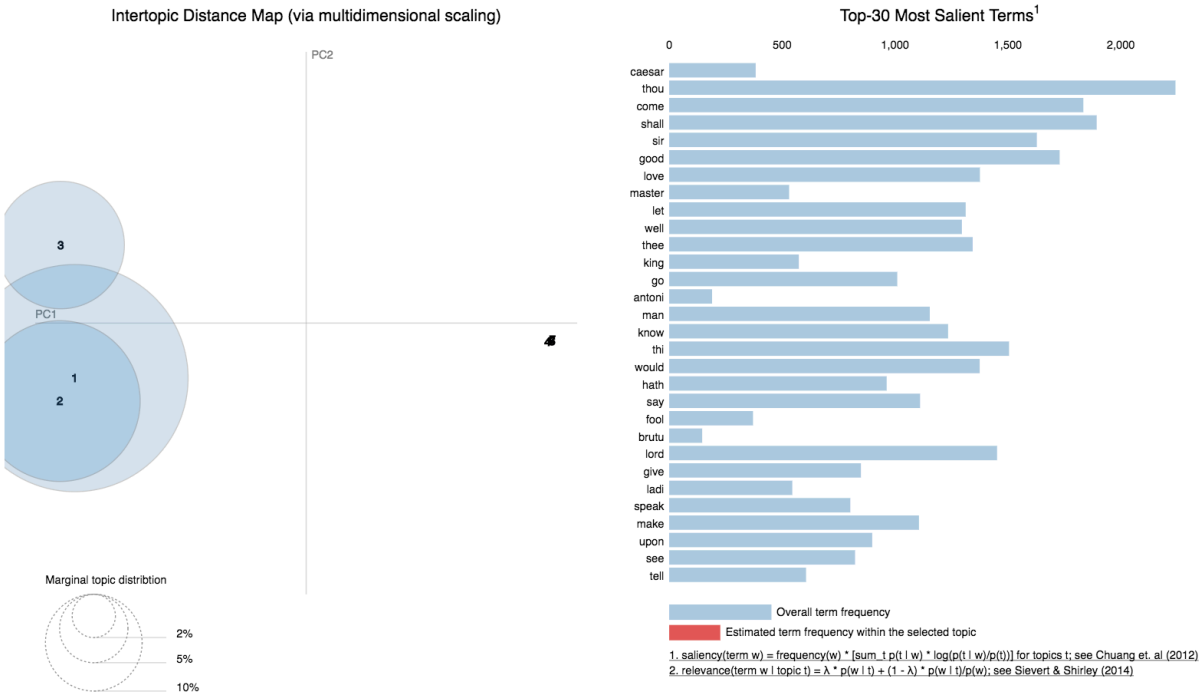




Finally, we realized that the resulting clusters shared common traits. Each cluster has a broad theme associated. They are either tragedies, stories related to kings, love stories or comedies. There is evidence to suggest that maybe Shakespeare was a group of people who focused on different writing styles. In order to get a deeper insight on this fact, I decided to explore run LDA in each of the cluster and check the dominant themes and words. The results can be seen below.

Cluster 1: Tragedy

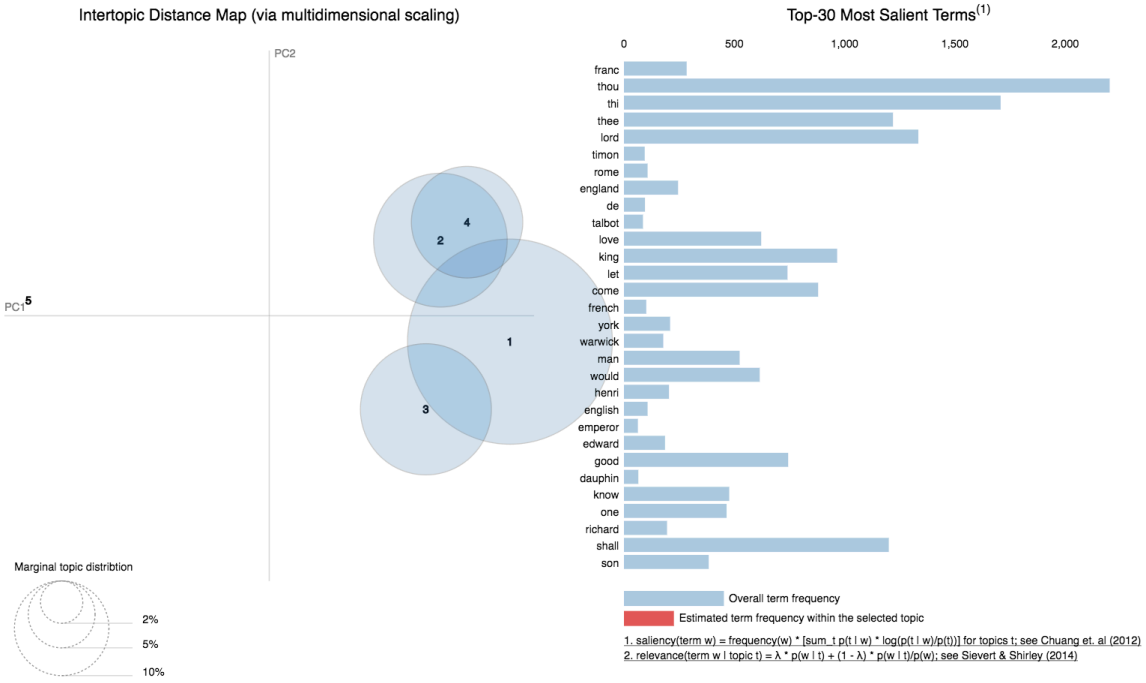
```
['Hamlet', 'Coriolanus', 'Cymbeline', 'Antony and Cleopatra', 'King Lear', 'Othello', 'Troilus and Cres
sida', 'A Winters Tale', 'Henry VIII', 'Alls well that ends well', 'Measure for measure', 'Loves Labour
s Lost', 'Merry Wives of Windsor', 'As you like it', 'Merchant of Venice', 'Julius Caesar', 'Much Ado a
bout nothing', 'Twelfth Night', 'Pericles']
```




```
[   (   0,
        '0.011*"thou" + 0.010*"shall" + 0.009*"come" + 0.007*"good" + '
        '0.007*"caesar" + 0.007*"lord" + 0.007*"let" + 0.006*"thi" + '
        '0.006*"know" + 0.006*"would" + 0.006*"man" + 0.006*"thee" + '
        '0.005*"well" + 0.005*"go" + 0.005*"sir"'),
    (   1,
        '0.011*"love" + 0.010*"man" + 0.009*"come" + 0.008*"good" + '
        '0.008*"would" + 0.008*"thou" + 0.007*"shall" + 0.007*"well" + '
        '0.007*"hath" + 0.007*"thee" + 0.007*"lord" + 0.006*"know" + '
        '0.006*"ladi" + 0.006*"god" + 0.006*"say"'),
    (   2,
        '0.011*"thou" + 0.009*"lord" + 0.009*"shall" + 0.008*"good" + '
        '0.008*"come" + 0.007*"sir" + 0.007*"thi" + 0.006*"thee" + '
        '0.006*"know" + 0.006*"let" + 0.006*"would" + 0.006*"love" + '
        '0.006*"make" + 0.006*"well" + 0.005*"one"'),
    (   3,
        '0.012*"posthumu" + 0.010*"imogen" + 0.010*"pisanio" + 0.010*"britain" '
        '+ 0.009*"roman" + 0.007*"cloten" + 0.007*"leonatu" + 0.007*"briton" + '
        '0.006*"milford" + 0.005*"luciu" + 0.005*"cymbelin" + 0.004*"jupit" + '
        '0.004*"tribut" + 0.004*"garment" + 0.004*"fidel"'),
    (   4,
        '0.015*"master" + 0.014*"sir" + 0.013*"come" + 0.011*"good" + '
        '0.011*"shall" + 0.009*"page" + 0.008*"go" + 0.008*"well" + '
        '0.008*"mistress" + 0.006*"let" + 0.006*"would" + 0.006*"ford" + '
        '0.005*"say" + 0.005*"thou" + 0.005*"hath"'),
    (   5,
        '0.000*"good" + 0.000*"thou" + 0.000*"shall" + 0.000*"sir" + '
        '0.000*"love" + 0.000*"come" + 0.000*"let" + 0.000*"thi" + 0.000*"say" '
        '+ 0.000*"know" + 0.000*"lord" + 0.000*"would" + 0.000*"well" + '
        '0.000*"thee" + 0.000*"like"'),
    (   6,
        '0.011*"thou" + 0.009*"come" + 0.008*"sir" + 0.008*"shall" + '
        '0.008*"good" + 0.008*"thi" + 0.007*"love" + 0.007*"thee" + '
        '0.007*"would" + 0.007*"well" + 0.006*"man" + 0.006*"let" + '
        '0.006*"say" + 0.006*"one" + 0.005*"make"')]
```

Cluster 2: Kings

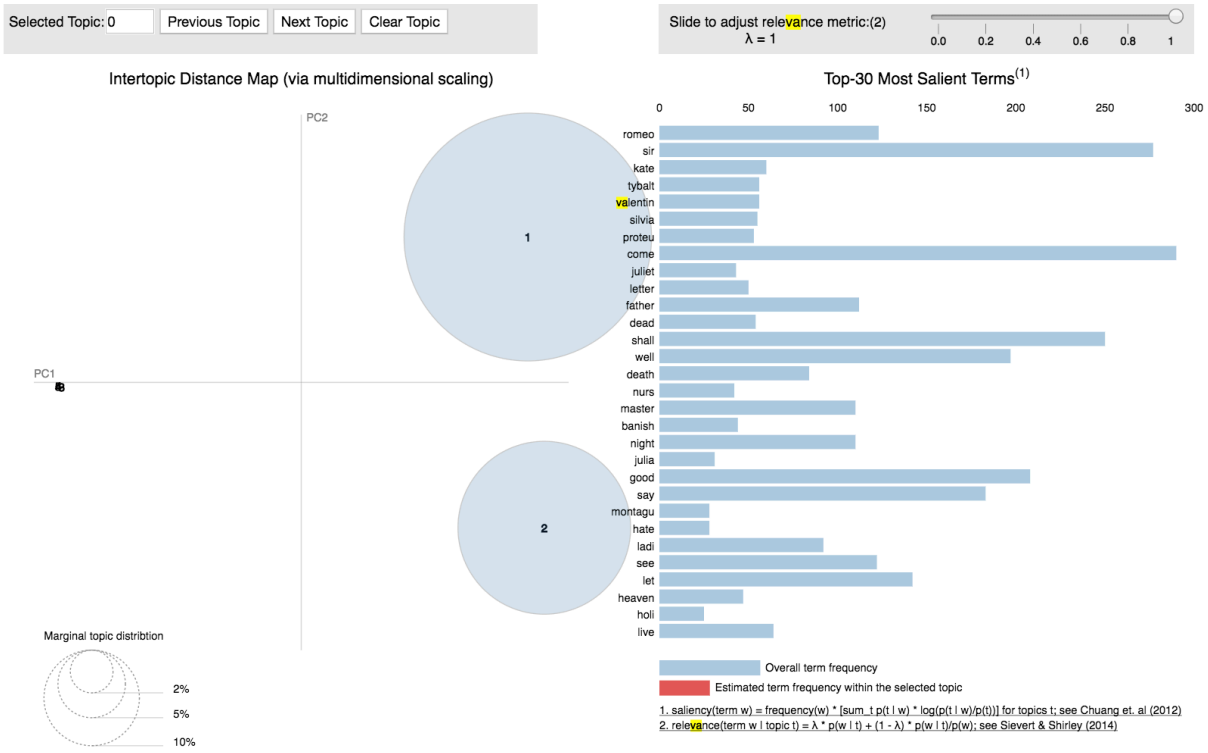
```
['Richard III', 'Henry V', 'Henry VI Part 2', 'Henry IV', 'Henry VI Part 3', 'Henry VI Part 1', '
Richard II', 'King John', 'Titus Andronicus', 'Timon of Athens', 'macbeth', 'The Tempest', 'A Mid
summer nights dream']
```




```
[   (   0,
        '0.015*"thou" + 0.015*"thi" + 0.010*"lord" + 0.010*"king" + '
        '0.009*"thee" + 0.008*"shall" + 0.006*"henri" + 0.006*"warwick" + '
        '0.005*"let" + 0.005*"father" + 0.005*"like" + 0.005*"come" + '
        '0.004*"hath" + 0.004*"make" + 0.004*"franc"'),
    (   1,
        '0.016*"thi" + 0.014*"thou" + 0.008*"thee" + 0.008*"lord" + '
        '0.007*"shall" + 0.007*"king" + 0.007*"come" + 0.006*"let" + '
        '0.005*"hand" + 0.005*"son" + 0.005*"hath" + 0.005*"make" + '
        '0.004*"say" + 0.004*"rome" + 0.004*"good"'),
    (   2,
        '0.017*"thou" + 0.014*"lord" + 0.012*"thi" + 0.010*"thee" + '
        '0.008*"shall" + 0.007*"good" + 0.006*"come" + 0.006*"king" + '
        '0.005*"god" + 0.005*"let" + 0.005*"well" + 0.005*"man" + 0.004*"make" '
        '+ 0.004*"would" + 0.004*"upon"'),
    (   3,
        '0.013*"thou" + 0.009*"shall" + 0.008*"thi" + 0.007*"thee" + '
        '0.006*"come" + 0.006*"king" + 0.006*"good" + 0.006*"love" + '
        '0.005*"upon" + 0.005*"would" + 0.005*"us" + 0.005*"make" + '
        '0.005*"let" + 0.005*"hath" + 0.005*"like"'),
    (   4,
        '0.000*"thou" + 0.000*"thi" + 0.000*"lord" + 0.000*"thee" + '
        '0.000*"let" + 0.000*"shall" + 0.000*"king" + 0.000*"say" + 0.000*"us" '
        '+ 0.000*"well" + 0.000*"like" + 0.000*"come" + 0.000*"upon" + '
        '0.000*"good" + 0.000*"know"')]
```

Cluster 3: Love, arranged matrimonies,

```
['Romeo and Juliet', 'Taming of the Shrew', 'Two Gentlemen of Verona']
```

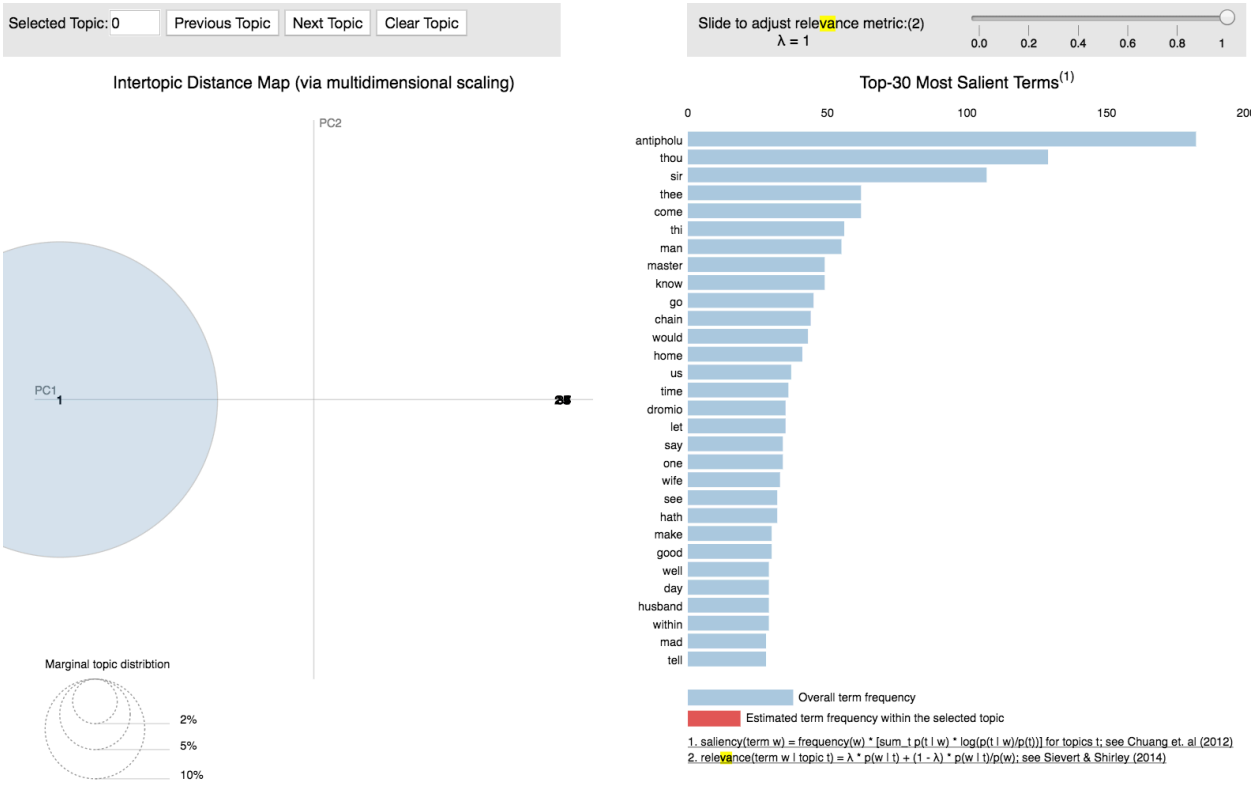



```
[   (   0,
        '0.001*"antipholu" + 0.001*"thou" + 0.001*"sir" + 0.001*"thi" + '
        '0.001*"thee" + 0.001*"man" + 0.001*"master" + 0.001*"know" + '
        '0.001*"come" + 0.001*"would" + 0.001*"home" + 0.001*"let" + '
        '0.001*"dromio" + 0.001*"chain" + 0.001*"upon" + 0.001*"us" + '
        '0.001*"well" + 0.001*"one" + 0.001*"hath" + 0.001*"till" + '
        '0.001*"say" + 0.001*"go" + 0.001*"husband" + 0.001*"make" + '
        '0.001*"hous" + 0.001*"wife" + 0.001*"tell" + 0.001*"within" + '
        '0.001*"day" + 0.001*"see"'),
    (   1,
        '0.001*"antipholu" + 0.001*"thou" + 0.001*"sir" + 0.001*"thee" + '
        '0.001*"man" + 0.001*"thi" + 0.001*"know" + 0.001*"master" + '
        '0.001*"come" + 0.001*"chain" + 0.001*"would" + 0.001*"dromio" + '
        '0.001*"us" + 0.001*"wife" + 0.001*"hath" + 0.001*"go" + 0.001*"time" '
        '+ 0.001*"home" + 0.001*"see" + 0.001*"say" + 0.001*"husband" + '
        '0.001*"one" + 0.001*"let" + 0.001*"good" + 0.001*"within" + '
        '0.001*"hous" + 0.001*"tell" + 0.001*"day" + 0.001*"well" + '
        '0.001*"shall"'),
    (   2,
        '0.025*"antipholu" + 0.018*"thou" + 0.015*"sir" + 0.009*"come" + '
        '0.009*"thee" + 0.008*"thi" + 0.008*"man" + 0.007*"master" + '
        '0.007*"know" + 0.006*"go" + 0.006*"chain" + 0.006*"would" + '
        '0.006*"home" + 0.005*"us" + 0.005*"time" + 0.005*"let" + '
        '0.005*"dromio" + 0.005*"say" + 0.005*"one" + 0.005*"wife" + '
        '0.005*"hath" + 0.005*"see" + 0.004*"make" + 0.004*"good" + '
        '0.004*"well" + 0.004*"day" + 0.004*"within" + 0.004*"husband" + '
        '0.004*"mad" + 0.004*"tell"'),
    (   3,
        '0.001*"antipholu" + 0.001*"thou" + 0.001*"sir" + 0.001*"thi" + '
        '0.001*"know" + 0.001*"thee" + 0.001*"man" + 0.001*"go" + 0.001*"see" '
        '+ 0.001*"come" + 0.001*"master" + 0.001*"would" + 0.001*"dromio" + '
        '0.001*"time" + 0.001*"let" + 0.001*"us" + 0.001*"home" + 0.001*"hath" '
        '+ 0.001*"husband" + 0.001*"chain" + 0.001*"within" + 0.001*"one" + '
        '0.001*"say" + 0.001*"think" + 0.001*"upon" + 0.001*"till" + '
        '0.001*"shall" + 0.001*"wife" + 0.001*"day" + 0.001*"gold"'),
    (   4,
        '0.001*"antipholu" + 0.001*"thou" + 0.001*"sir" + 0.001*"thee" + '
        '0.001*"come" + 0.001*"thi" + 0.001*"man" + 0.001*"chain" + 0.001*"go" '
        '+ 0.001*"would" + 0.001*"know" + 0.001*"let" + 0.001*"home" + '
        '0.001*"hath" + 0.001*"master" + 0.001*"husband" + 0.001*"dromio" + '
        '0.001*"us" + 0.001*"one" + 0.001*"well" + 0.001*"wife" + 0.001*"till" '
        '+ 0.001*"see" + 0.001*"say" + 0.001*"time" + 0.001*"day" + '
        '0.001*"mistress" + 0.001*"within" + 0.001*"good" + 0.001*"tell"')]
```

Cluster 4: Comedy, short story

A Comedy of Errors

```
[   (   0,
        '0.001*"antipholu" + 0.001*"thou" + 0.001*"sir" + 0.001*"thi" + '
        '0.001*"thee" + 0.001*"know" + 0.001*"master" + 0.001*"man" + '
        '0.001*"come" + 0.001*"would" + 0.001*"home" + 0.001*"let" + '
        '0.001*"dromio" + 0.001*"upon" + 0.001*"us"'),
    (   1,
        '0.001*"antipholu" + 0.001*"thou" + 0.001*"sir" + 0.001*"thee" + '
        '0.001*"thi" + 0.001*"man" + 0.001*"know" + 0.001*"come" + '
        '0.001*"master" + 0.001*"would" + 0.001*"chain" + 0.001*"dromio" + '
        '0.001*"hath" + 0.001*"wife" + 0.001*"us"'),
    (   2,
        '0.025*"antipholu" + 0.018*"thou" + 0.015*"sir" + 0.009*"come" + '
        '0.009*"thee" + 0.008*"thi" + 0.008*"man" + 0.007*"master" + '
        '0.007*"know" + 0.006*"go" + 0.006*"chain" + 0.006*"would" + '
        '0.006*"home" + 0.005*"us" + 0.005*"time"'),
    (   3,
        '0.001*"antipholu" + 0.001*"thou" + 0.001*"sir" + 0.001*"thi" + '
        '0.001*"know" + 0.001*"thee" + 0.001*"go" + 0.001*"man" + 0.001*"come" '
        '+ 0.001*"see" + 0.001*"would" + 0.001*"master" + 0.001*"dromio" + '
        '0.001*"let" + 0.001*"time"'),
    (   4,
        '0.001*"antipholu" + 0.001*"thou" + 0.001*"sir" + 0.001*"thee" + '
        '0.001*"come" + 0.001*"thi" + 0.001*"man" + 0.001*"chain" + 0.001*"go" '
        '+ 0.001*"would" + 0.001*"know" + 0.001*"let" + 0.001*"hath" + '
        '0.001*"home" + 0.001*"husband"'),
    (   5,
        '0.001*"antipholu" + 0.001*"thou" + 0.001*"sir" + 0.001*"thee" + '
        '0.001*"chain" + 0.001*"know" + 0.001*"man" + 0.001*"master" + '
        '0.001*"come" + 0.001*"dromio" + 0.001*"thi" + 0.001*"go" + '
        '0.001*"would" + 0.001*"home" + 0.001*"let"'),
    (   6,
        '0.001*"thou" + 0.001*"antipholu" + 0.001*"sir" + 0.001*"man" + '
        '0.001*"come" + 0.001*"thee" + 0.001*"thi" + 0.001*"master" + '
        '0.001*"chain" + 0.001*"would" + 0.001*"home" + 0.001*"know" + '
        '0.001*"time" + 0.001*"us" + 0.001*"see"')]
```



I could have continued to explore deeper and exclude words that do not seem to add information to the LDA analysis. Due to time constraint I have stopped the analysis here. We could explore in each text if there is any pattern in the different characters of the texts that belong to the same cluster. That could be done by first identifying the lines of each of this character and reorganize the LDA analysis by means of characters and not texts. The result of this analysis would show us whether characters share common patters. This could lead us to conclude whether writing patterns change among characters and clusters.