HW1

Comparing Corpora with Corpus Statistics

Bing-Je Wu

IST 664 – Natural Language Processing

Table of Contents

[Introduction 2](#_Toc38113739)

[Analysis and Models 2](#_Toc38113740)

[About the Data 2](#_Toc38113741)

[Data Processing 2](#_Toc38113742)

[Results 4](#_Toc38113743)

[Conclusion 8](#_Toc38113744)

[Reference 8](#_Toc38113745)

# Introduction

Literature is a term used to describe written material. The word is derived from the Latin word literature meaning "writing formed with letters,". It most commonly refers to works of the creative imagination, including poetry, drama, fiction, and nonfiction. Literature represents the culture and tradition of a language or a people. Lots of valuable information can be retrieved from it. And, the corpus statistic is a useful tool that can help extract some of information from the text. But, can it find the difference between two works from the same author? The assignment is going to find out this question.

Analysis and Models  
The NLTK library and regular expression library were used to tokenize the documents, filter unwanted tokens, convert unigram to bigram and count the frequency score and pair mutual information (PMI) score.

## About the Data

Two large documents from NLTK were used for the assignment. Both documents are classic tragedy plays written by William Shakespeare. One is The Tragedy of Macbeth (Macbeth). The other is The Tragedy of Hamlet, Prince of Denmark (Hamlet). Both are the same genre of play, but one is Shakespeare’s shortest tragedy.

***Table 1****. Document length comparison*

|  |  |  |
| --- | --- | --- |
|  | Hamlet | Macbeth |
| Words | 36380 | 22188 |

Data Processing  
To get the tokens more efficiently, a string method, ‘lower()’, had been applied to make all characters become lowercase for both of documents. After converting string to lowercase, a ‘word\_tokenize()’ method from NLTK was implemented for converting text into tokens. With the tokens converted, there were some statistics can be computed, such as the frequency distribution of tokens. The Table 2 shows the counts for each of tokens after applying the string method, ‘lower()’. By observing on the Table 2, it is easy to find that some of the tokens were made of non-alphabetic character. To get rid of the tokens that were made of non-alphabetic character, a user-defined function, ‘alpha\_filter()’, was built to filter out those non-alphabetic characters.

***Table 2****. Sample of word frequency count after converting to lowercase from Hamlet and Macbeth*

|  |  |
| --- | --- |
| **Hamlet**  Token, Counts  (',', 2892),  ('.', 1877),  ('the', 993),  ('and', 862),  ('to', 683),  ('of', 610),  (':', 566),  ('i', 559),  ('you', 527),  ('my', 502),  ('a', 497),  ('?', 459) | **Macbeth**  Token, Counts  (',', 1962),  ('.', 1174),  ('the', 649),  ('and', 545),  (':', 477),  ('to', 383),  ('of', 338),  ('i', 331),  ('?', 241),  ('a', 239),  ('that', 236),  ('is', 211), |

According to the Zipf’s Law, a few occur very often while many others occur rarely, such as stopwords. The common stopwords are like ‘I’, ‘of’, ‘at’…etc. The stopwords from NLTK package was used to filter out the common stopwords. Because two documents are plays instead of an ordinal novel, some character’s name would appear many times. Therefore, some of main character’s name needed to be removed, including the characters’ full name and abbreviated name. By recognizing the high frequent characters’ name or abbreviation, it can be added to the list of stopwords as filter.

***Table 3****. Word counts for filtering the non-alphabetic characters (alpha\_filter)*

|  |  |  |
| --- | --- | --- |
|  | Hamlet | Macbeth |
| Before filtering | 36380 | 22188 |
| After filtering | 30055 | 18049 |

***Table 4****. User-defined stopwords*

|  |  |  |
| --- | --- | --- |
|  | Hamlet | Macbeth |
| User-defined stopwords | 'laer', 'laertes', 'horatio',  'rosin', 'ophe', 'hor',  'ham', 'hamlet', 'qu', 'queene',  'polon', 'pol', "'s",  'sha', 'wo', 'y', "'s", "'d",  "'ll", "'t", "'m", "'re",  "'ve", "n't" | 'macb', 'macbeth', 'macd',  'banquo', 'lenox', 'mal',  'banq','rosse', "'s", 'sha',  'wo', 'y', "'s", "'d",  "'ll", "'t", "'m", "'re",  "'ve", "n't" |

***Table 5****. Word counts for filtering the stopwords (stopwords from NLTK + user-defined stopwords)*

|  |  |  |
| --- | --- | --- |
|  | Hamlet | Macbeth |
| Stopwords | 203 | 199 |
| Before filtering | 30055 | 18049 |
| After filtering | 14776 | 9489 |

A NLTK method, ‘bigram()’ was used to convert the unigrams to bigrams. Since the original tokens were consisting the non-alphabetic characters and stopwords, the unigram vector, after applying filters, was used as the input of the ‘bigram()’ method.

# Results

***Table 6:*** *List of top 50 words by frequency (normalized by the length of the document)*

|  |  |
| --- | --- |
| **Hamlet**  lord : 0.0143  haue : 0.0118  king : 0.0116  shall : 0.0072  come : 0.007  let : 0.007  thou : 0.007  good : 0.0066  thy : 0.0061  enter : 0.0058  oh : 0.0055  like : 0.0052  well : 0.0047  know : 0.0047  would : 0.0046  selfe : 0.0045  may : 0.0044  loue : 0.0044  sir : 0.0042  vs : 0.0041  giue : 0.004  thee : 0.0039  ile : 0.0039  must : 0.0039  hath : 0.0039  speake : 0.0037  make : 0.0037  say : 0.0035  doe : 0.0035  vpon : 0.0034  heere : 0.0034  father : 0.0034  go : 0.0032  one : 0.0031  see : 0.0031  man : 0.0031  time : 0.003  mine : 0.003  much : 0.0029  heauen : 0.0029  tell : 0.0029  thinke : 0.0028  thus : 0.0028  mother : 0.0027  play : 0.0027  night : 0.0026  yet : 0.0025  death : 0.0024  vp : 0.0024  againe : 0.0023 | **Macbeth**  haue : 0.0083  thou : 0.0059  enter : 0.0055  shall : 0.0046  thee : 0.0041  vpon : 0.004  yet : 0.0039  thy : 0.0038  vs : 0.0038  come : 0.0037  king : 0.0036  hath : 0.0035  good : 0.0032  lady : 0.0032  would : 0.0032  time : 0.0031  let : 0.0028  like : 0.0027  say : 0.0026  make : 0.0026  doe : 0.0026  lord : 0.0026  must : 0.0024  done : 0.0024  ile : 0.0024  feare : 0.0024  wife : 0.0023  man : 0.0022  well : 0.0022  know : 0.0022  selfe : 0.0022  one : 0.0022  great : 0.0021  see : 0.0021  may : 0.0021  exeunt : 0.002  speake : 0.002  night : 0.002  sir : 0.002  mine : 0.0018  vp : 0.0018  th : 0.0018  heere : 0.0018  thane : 0.0017  giue : 0.0016  looke : 0.0016  things : 0.0016  sleepe : 0.0016  hand : 0.0016  blood : 0.0016 |

***Table 7:*** *List of top 50 bigrams by frequencies*

|  |  |
| --- | --- |
| **Hamlet**  ('good', 'lord') : 23  ('enter', 'king') : 16  ('wee', 'l') : 13  ('haue', 'seene') : 11  ('exeunt', 'enter') : 10  ('thou', 'hast') : 9  ('haue', 'heard') : 9  ('enter', 'polonius') : 9  ('lord', 'haue') : 9  ('fathers', 'death') : 8  ('let', 'vs') : 7  ('thou', 'art') : 7  ('king', 'haue') : 7  ('would', 'haue') : 7  ('set', 'downe') : 7  ('good', 'friends') : 7  ('well', 'lord') : 7  ('rosincrance', 'guildensterne') : 7  ('let', 'see') : 7  ('dost', 'thou') : 7  ('king', 'king') : 7  ('mine', 'owne') : 6  ('king', 'oh') : 6  ('ile', 'haue') : 6  ('let', 'come') : 6  ('sit', 'downe') : 5  ('enter', 'ghost') : 5  ('heauen', 'earth') : 5  ('lord', 'would') : 5  ('let', 'know') : 5  ('lord', 'exeunt') : 5  ('wilt', 'thou') : 5  ('tell', 'vs') : 5  ('reynol', 'lord') : 5  ('shall', 'heare') : 5  ('come', 'come') : 5  ('king', 'shall') : 5  ('king', 'polonius') : 5  ('get', 'thee') : 4  ('good', 'night') : 4  ('marcellus', 'mar') : 4  ('thy', 'selfe') : 4  ('father', 'lost') : 4  ("would'st", 'thou') : 4  ('giue', 'leaue') : 4  ('take', 'thy') : 4  ('lord', 'king') : 4  ('mar', 'lord') : 4  ('enter', 'ophelia') : 4  ('thy', 'soule') : 4 | **Macbeth**  ('exeunt', 'scena') : 15  ('thane', 'cawdor') : 13  ('knock', 'knock') : 10  ('thou', 'art') : 9  ('haue', 'done') : 8  ('enter', 'lady') : 8  ('good', 'lord') : 8  ('let', 'vs') : 7  ('wee', 'l') : 7  ('enter', 'three') : 5  ('three', 'witches') : 5  ('scena', 'secunda') : 5  ('enter', 'king') : 5  ('worthy', 'thane') : 5  ('thy', 'selfe') : 5  ('euery', 'one') : 5  ('would', 'haue') : 5  ('mine', 'eyes') : 5  ('make', 'vs') : 5  ('enter', 'malcolme') : 5  ('mine', 'owne') : 5  ('ten', 'thousand') : 4  ('shew', 'shew') : 4  ('haue', 'seene') : 4  ('come', 'come') : 4  ('malcolme', 'donalbaine') : 4  ('haile', 'king') : 4  ('would', 'make') : 4  ('hath', 'made') : 4  ('scena', 'prima') : 4  ('see', 'thee') : 4  ('tertia', 'enter') : 4  ('old', 'man') : 4  ('enter', 'macduffe') : 4  ('thy', 'face') : 4  ('woman', 'borne') : 4  ('borne', 'woman') : 4  ('king', 'scotland') : 3  ('thee', 'thy') : 3  ('enter', 'angus') : 3  ('king', 'thane') : 3  ('scena', 'tertia') : 3  ('thunder', 'enter') : 3  ('ile', 'doe') : 3  ('doe', 'ile') : 3  ('giue', 'thee') : 3  ('looke', 'haue') : 3  ('weyward', 'sisters') : 3  ('looke', 'like') : 3  ('haile', 'haile') : 3 |

***Table 8:*** *List of top 50 bigrams by their Mutual Information scores (using min frequency 5)*

|  |  |
| --- | --- |
| **Hamlet**  (('rosincrance', 'guildensterne'), 9.111428613152404)  (('wee', 'l'), 8.644517273515014)  (('sit', 'downe'), 8.472456527728708)  (('set', 'downe'), 7.498451736261652)  (('fathers', 'death'), 7.359115054652763)  (('dost', 'thou'), 6.498451736261652)  (('wilt', 'thou'), 6.472456527728708)  (('heauen', 'earth'), 6.354314068388943)  (('enter', 'polonius'), 6.289574121399687)  (('exeunt', 'enter'), 6.239943353675086)  (('mine', 'owne'), 6.022302722679424)  (('thou', 'art'), 5.957883354898948)  (('good', 'friends'), 5.7956857154812464)  (('haue', 'heard'), 5.762327118534818)  (('thou', 'hast'), 5.56556593212019)  (('haue', 'seene'), 5.537260562900045)  (('enter', 'ghost'), 5.371187886953338)  (('tell', 'vs'), 4.815894153604816)  (('reynol', 'lord'), 4.751357339021524)  (('shall', 'heare'), 4.524538663860136)  (('let', 'see'), 4.434321398841936)  (('king', 'polonius'), 4.424703396280339)  (('good', 'lord'), 4.038721074217058)  (('let', 'vs'), 4.027146017336062)  (('enter', 'king'), 4.015312460142638)  (('lord', 'exeunt'), 3.928235101105603)  (('let', 'know'), 3.36393207095054)  (('ile', 'haue'), 3.126738544743697)  (('would', 'haue'), 3.119649119957373)  (('let', 'come'), 3.0350512154214133)  (('well', 'lord'), 2.807940867387888)  (('come', 'come'), 2.772016809587617)  (('king', 'oh'), 2.6698158941168746)  (('lord', 'would'), 2.364334215912274)  (('king', 'shall'), 2.0051645047665545)  (('lord', 'haue'), 1.8485828518852365)  (('king', 'king'), 1.805793563635845)  (('king', 'haue'), 1.780847206505614) | **Macbeth**  (('wee', 'l'), 9.238035549771494)  (('three', 'witches'), 8.83352871798482)  (('scena', 'secunda'), 8.752608722601252)  (('knock', 'knock'), 8.533968436125914)  (('thane', 'cawdor'), 7.876306446826158)  (('exeunt', 'scena'), 7.752608722601252)  (('mine', 'eyes'), 7.374097099347523)  (('worthy', 'thane'), 6.890112246351187)  (('mine', 'owne'), 6.74606587673448)  (('euery', 'one'), 6.533968436125912)  (('thou', 'art'), 5.76909684538982)  (('enter', 'malcolme'), 5.493678715100195)  (('enter', 'three'), 5.493678715100195)  (('good', 'lord'), 5.379150327073807)  (('let', 'vs'), 4.819722918459789)  (('thy', 'selfe'), 4.726613514068308)  (('make', 'vs'), 4.44121129520606)  (('enter', 'lady'), 4.287227837632768)  (('haue', 'done'), 4.152019986730695)  (('enter', 'king'), 3.466197978678087)  (('would', 'haue'), 3.0486422468853878) |

Table 6, 7, 8 show the list of top 50 words by frequency (normalized by the length of the document), list of top 50 bigrams by frequencies and list of top 50 bigrams by their Mutual Information scores (using min frequency 5).

How to see the difference between two works, Hamlet and Macbeth? Apart from the length of the document, is there any other major difference? From the Table 6, it is easy to see that there are still differences between two documents even both works are from the same author and belong to same genre. For example, the positive word ‘good’ appeared in Hamlet has the frequency that almost doubles the frequency appeared in Macbeth. The words such as ‘shall’, ‘thou’, ‘thy’ are higher in Hamlet than Macbeth. Some of the verbs such as ‘doe’, ‘say’, ‘give’ are lower in Macbeth. It seems like William Shakespeare had been evolved on his writing style and started to reduce some of the words that he frequently used in the past.

***Table 9:*** *List of top 50 trigrams by frequencies after applying alpha\_filter()*

|  |  |
| --- | --- |
| **Hamlet**  ('my', 'lord', 'ham') : 62  ('my', 'lord', 'i') : 18  ('good', 'my', 'lord') : 14  ('i', 'my', 'lord') : 13  ('i', 'pray', 'you') : 12  ('lord', 'ham', 'i') : 11  ('that', 'i', 'haue') : 11  ('i', 'can', 'not') : 9  ("'t", 'is', 'a') : 8  ('my', 'good', 'lord') : 8  ('it', 'is', 'a') : 8  ('my', 'lord', 'polon') : 7  ('lord', 'i', 'haue') : 7  ('let', 'me', 'see') : 7  ('i', 'haue', 'seene') : 6  ('i', 'know', 'not') : 6  ('if', 'it', 'be') : 6  ('well', 'my', 'lord') : 6 ('rosincrance', 'and', 'guildensterne') : 6  ('you', 'can', 'not') : 5  ('it', 'is', 'not') : 5  ('ham', 'i', 'am') : 5  ('i', 'do', 'not') : 5  ('ophe', 'my', 'lord') : 5  ('and', 'with', 'a') : 5  ('my', 'lord', 'exeunt') : 5  ('exeunt', 'enter', 'hamlet') : 5 ('lord', 'ham', 'why') : 5  ('to', 'the', 'king') : 5  ('enter', 'king', 'queene') : 5  ('what', 'is', 'the') : 5  ('a', 'kinde', 'of') : 5  ('my', 'lord', 'you') : 5  ('get', 'thee', 'to') : 4  ('speake', 'to', 'it') : 4  ('what', "'s", 'the') : 4  ('you', 'for', 'your') : 4  ('my', 'lord', 'and') : 4  ('hor', 'my', 'lord') : 4  ('my', 'lord', 'it') : 4  ('i', 'saw', 'him') : 4  ('my', 'lord', 'the') : 4  ("'t", 'is', 'true') : 4  ('by', 'no', 'meanes') : 4  ('what', 'i', 'haue') : 4  ('you', 'your', 'selfe') : 4  ('as', 'it', 'is') : 4  ('i', 'doe', 'not') : 4  ('let', "'s", 'follow') : 4  ("'t", 'is', 'not') : 4 | **Macbeth**  ('thane', 'of', 'cawdor') : 13  ('the', 'thane', 'of') : 8  ('my', 'good', 'lord') : 8  ('i', 'pray', 'you') : 7  ('can', 'not', 'be') : 6  ('who', "'s", 'there') : 6  ('knock', 'knock', 'knock') : 6  ('i', 'can', 'not') : 5  ('enter', 'macbeth', 'macb') : 5 ('what', "'s", 'the') : 5  ('i', 'my', 'good') : 5  ('exeunt', 'scena', 'secunda') : 4 ('this', 'is', 'the') : 4  ('all', 'haile', 'macbeth') : 4 ('there', "'s", 'no') : 4  ('it', 'is', 'a') : 4  ('he', 'ha', "'s") : 4  ('i', 'would', 'not') : 4  ('i', 'see', 'thee') : 4  ('i', 'haue', 'done') : 4  ('good', 'lord', 'macb') : 4  ('i', 'will', 'not') : 4  ('my', 'lord', 'macb') : 4  ('she', 'ha', "'s") : 4  ('borne', 'of', 'woman') : 4  ('to', 'the', 'king') : 3  ('king', 'of', 'scotland') : 3  ('rosse', 'and', 'angus') : 3  ('exeunt', 'scena', 'tertia') : 3 ('thunder', 'enter', 'the') : 3 ('enter', 'the', 'three') : 3  ('the', 'three', 'witches') : 3  ('i', 'haue', 'a') : 3  ('thee', 'thane', 'of') : 3  ('why', 'doe', 'you') : 3  ('and', 'to', 'be') : 3  ('giue', 'me', 'your') : 3  ('exeunt', 'scena', 'quarta') : 3 ('but', 'i', 'haue') : 3  ('take', 'my', 'leaue') : 3  ('me', 'to', 'the') : 3  ('to', 'night', 'lady') : 3  ('i', 'haue', 'no') : 3  ('which', 'would', 'be') : 3  ('i', 'dare', 'not') : 3  ('now', 'do', "'s") : 3  ('you', 'haue', 'done') : 3  ('scena', 'prima', 'enter') : 3  ('with', 'a', 'torch') : 3  ('vpon', 'me', 'and') : 3 |

Table 8 is not able to show all 50 bigrams for Hamlet and Macbeth. Both documents do not have 50 bigrams by PMI scores because of the condition, having the bigram appear at least five times in the text. The Table 7 also show that not too much bigrams have appeared more than five times in Macbeth.

The bigram frequency shows the number of counts that were put together in the text. The pair mutual information (PMI) is a quantified measure for how much more or less likely we are to see the two events co-occur, given their individual probabilities, and relative to the case where the two are completely independent. The equation can be written as . Thus, when the PMI is higher, it means that the two words are likely to express a unique concept.

# Conclusion

Author’s writing style can be changed because of the history, his/her experiences and education…etc. Especially, in the early modern period in between the Middle Ages and the Industrial Revolution, people’s life is vastly changing. Utilizing the corpus statistics can see the difference between two works from William Shakespeare in different period.

# Reference

1 Understanding Pointwise Mutual Information in NLP. [Web Link](https://medium.com/dataseries/understanding-pointwise-mutual-information-in-nlp-e4ef75ecb57a) ;  
2 Hamlet, from Wikipedia. [Web Link](https://en.wikipedia.org/wiki/Hamlet) ;   
3 Macbeth, from Wikipedia. [Web Link](https://en.wikipedia.org/wiki/Macbeth)