Text Mining the Works of Christopher Marlowe

Ruben Thoplan

Lecturer in Statistics, Department of Economics and Statistics, Faculty of Social Studies and Humanities, University of Mauritius, Réduit, Mauritius

ABSTRACT

In this paper, the application of statistical techniques in literature through Christopher Marlowe’s works is explored through the use of text data mining algorithms. A tag cloud is used as visualization technique to identify patterns in the words used by Marlowe in his plays and poem. A cluster analysis using the complete linkage method with squared Euclidean distance is adopted to identify any agglomeration of the different texts of Marlowe. The findings of this paper shows that Marlowe uses the words “king”, “death”, “love”, “heaven”, “crown”, “soul” relatively often in his plays or poem. Besides, three main agglomerations of the texts of Marlowe can be identified.

Keywords: text data mining, tag cloud, cluster analysis, complete linkage and squared Euclidean distance

1. Introduction

The field of statistics has always proved to be supportive in many disciplines like economics, management, agriculture, law and many others. However, very often, it is not given its due credit in terms of its applications in the field of literature. For example, statistics has been used in authorship attribution which can be seen in the works of Kukushkina et al. (2001) and Sallis & Shanmuganatham (2008). Traditional statistical tools like chi square tests have often been used in authorship attribution like in the example of Grieve (2007). Apart from authorship attribution in literature, statistics can be used to underpin similarities between different texts using techniques like cluster analysis (Berry & Castellanos, 2007).

In this paper, the works of Christopher Marlowe are statistically analysed to identify any patterns which may arise in his plays or poem. Christopher Marlowe is an eminent English Renaissance poet and playwright who lived in the 16th century (Metcalf, 2002). The works of Marlowe which will be analyzed in this paper are the following:

Text 1: The Tragedy of Dido, Queen of Carthage
Text 2: The Tragical History of Dr. Faustus
Text 3: The Jew of Malta
Text 4: Tamburlaine the Great, Part I
Text 5: Tamburlaine the Great, Part II.
Text 6: Massacre at Paris
Text 7: Hero and Leander
Text 8: Edward II.

*The Tragedy of Dido, Queen of Carthage* is a tragic play which revolves around Dido, the Queen of Carthage, now modern Tunisia, and the thematic issues are love, betrayal and death (Marlowe, The Tragedy of Dido Queene of Carthage by Christopher Marlowe, 1594). *The Tragical History of Dr. Faustus*, which relates to religion and black magic, involves Dr. Faustus, a Renaissance scholar, who sells his soul to the devil for the acquisition of power and knowledge (Marlowe, The Tragical History of Doctor Faustus, 1616). *The Jew of Malta* is also about religion, power and death (Marlowe, The Jew of Malta, 1633). *Tamburlaine the Great*, Part I (Marlowe, Tamburlaine the Great, Part I, 1590) and
Part II (Marlowe, Tamburlaine the Great, Part II, 1590) are both about power and politics. *Massacre at Paris* is another play based on true historical facts which involve murder due to a result of religious war (Marlowe, Massacre at Paris, 1593). In fact in, *The Tragical History of Dr. Faustus*, *The Jew of Malta* and *Massacre at Paris*, the common denominator is corruption of power, and the subversion of “natural” order. *Hero and Leander* is a tragic love poem that results in death (Marlowe, Hero and Leander, 1598) and *Edward II* is about power and death (Marlowe, Edward the Second, 1594) where the death scenes are very intense like in *The Tragical History of Dr. Faustus*.

In this paper, it is attempted to extract information from the different works of Marlowe, using text mining algorithm, in view of identifying patterns which may arise in his plays or poem. Although a program cannot read and understand a text as a person would, text mining algorithm can be used to derive associations which are not necessarily explicitly evident to a human, among words and different texts. Indeed, text mining methods such as clustering algorithms and visualization techniques have been applied for literature analysis (Tsatsoulis, 2013).

This paper is organized such that the second section introduces the text mining methods adopted in this paper such as clustering and other techniques. The third section provides the results obtained from the applications of the methods stated in section two. The four section attempts some discussions and recommendations.

2. Method

Text mining, also known as text data mining, is an interdisciplinary field related to data mining, linguistics, computational statistics and computer science (Feinerer, Hornik, & Meyer, 2008). According to Feinerer, et al. (2008), text is transformed into a structured format using frequencies of words where data mining techniques like clustering and classification can be applied.

Following the framework described by Williams (2014), the first step in the text data mining process would be to load a corpus. A corpus is simply a collection of documents that are used for analysis and the corpus used in this study is the collection of text 1 to text 8 described in section 1. However, before setting up the corpus, preliminary processing of the texts in “.txt” format downloaded from Project Gutenberg website (Hart, 1971), has been carried out to remove any part of the texts which are unnecessary for analysis. The processing also involves collapsing a whole text into a single long string of words. This process is repeated for each text discussed in this paper.

After, the corpus has been loaded in the R statistical software; it is cleaned in line with any data mining procedure. The cleaning involves removing numbers from the text, removing all punctuation marks, removing any words that are not necessary for the analysis including stop words like “my”, “you”, and so on, stemming documents, and removing superfluous white spaces between words. This cleaning process is a very important task in the text mining process because, without a proper cleaning of the corpus, the analysis will be flawed.

Stemming involves the removal of suffixes in words. For example, by applying stemming, the suffix “ed” in the word “worked” would be removed to become “work”. In this paper, the Porter’s algorithm for word stemming is implemented by means of the “SnowballC” package available in R. The algorithm involves a series of steps to remove suffixes but avoids removing suffixes for which the stem is too short (Porter, 1980). The reason why the Porter’s algorithm is adopted in this paper is because it has become the standard for stemming English words (Willet, 2006).

After having prepared the corpus, a document term matrix can be obtained. In simple terms, a document term matrix is a matrix where the rows represent the documents in the corpus and the columns represent the words. Then, each cell in the matrix contains the frequency of a precise word in a specific document. In general, it is expected that a document term matrix would contain high
sparsity. High sparsity implies that the proportion of zero elements in the matrix is very high. From the document term matrix, the sum of the frequencies in each column would give the total number of occurrences of a specific word from all the documents in the corpus. Hence, the most frequent and least frequent terms used by the author can be identified.

Since the idea is about the identification of most common words used by Marlowe across documents, a sparse matrix would not be very helpful. In this regard, it is important to reduce the sparsity to focus on the most common terms used to be able to start identifying patterns. After reducing the sparsity, it is possible to identify the associations among terms. This is obtained by specifying a correlation limit showing how often two words appear together in a corpus. The most often two words appear together, the higher will be the correlation (value tending to 1). The less often two words appear together, the lower will be the correlation (value tending to 0).

In text mining, visualizing techniques can be applied to identify the patterns arising in a corpus in terms of the most frequent terms used. This can be done through a traditional bar chat and also through a word cloud also known as the tag cloud. A tag cloud generally displays words by varying the font size according to the importance of the words (Bateman, Gutwin, & Nacenta, 2008). A word with high importance appears in a font of large size and one with low importance appears in small size.

Apart visualizing methods, text data mining involves supervised and unsupervised learning methods. Supervised learning methods include predictive modeling whereas unsupervised learning methods include clustering techniques. There are hierarchical and non hierarchical clustering methods. In this paper, a hierarchical clustering method is adopted. According to Willett (1988), the complete linkage, a hierarchical clustering method is the best in terms of retrieval performance. For this reason, the complete linkage method for hierarchical clustering is adopted.

According to Aggarwal & Zhai (2012), “the similarity between two clusters is the worst-case similarity between any pair of documents in the two clusters.” The idea of the complete linkage method is that the distance between clusters is the maximum between the document in a specific cluster and that of another document from another cluster. Using the formulation of Lance & Williams (1967), the complete-link method can be represented as follows:

\[ d(i + j, k) = \max(d(i, k), d(j, k)) \]

where \( d(i, j) \) represents the distance between clusters \( i \) and \( j \).

The distance measure between two documents \( x \) and \( y \), represented by their term vectors \( \vec{x} \) and \( \vec{y} \) respectively, is the squared Euclidean distance. Using the notations of Huang (2008), the distance measure can be formulated as follows:

\[ d(\vec{x}, \vec{y}) = \sum_{t=1}^{n} (w_{tx} - w_{ty})^2 \]

where \( t = \{t_1, t_2, ..., t_n\}, w_{tx} = tf(x, t) \times \log \left( \frac{|d|}{df(t)} \right), \) \( tf \) stands for term frequencies and \( df(t) \) is the number of documents in which term \( t \) appears.

3. Results

The results from the application of the different text mining algorithms are here presented. In the first instance, a sample of the document term matrix after having reduced the sparsity is provided in table 1 below showing the most frequent terms used in the texts of Marlowe as a whole. It is clear from table 1 that Hero and Leander is different in terms of the words used. Words like “thou”, “thi” and “thee” are rarely used in Hero and Leander compared to the other texts. The plausible reason for this is that Hero and Leander is the only poem and the other texts are plays.
Table 1: Sample Document Term Matrix After Removal of Sparsity

<table>
<thead>
<tr>
<th>Text</th>
<th>Will</th>
<th>Thou</th>
<th>Shall</th>
<th>King</th>
<th>Thi</th>
<th>Lord</th>
<th>Now</th>
<th>Thee</th>
<th>See</th>
<th>Come</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>94</td>
<td>112</td>
<td>81</td>
<td>25</td>
<td>103</td>
<td>13</td>
<td>60</td>
<td>71</td>
<td>41</td>
<td>39</td>
</tr>
<tr>
<td>2</td>
<td>109</td>
<td>154</td>
<td>89</td>
<td>11</td>
<td>74</td>
<td>38</td>
<td>96</td>
<td>85</td>
<td>97</td>
<td>96</td>
</tr>
<tr>
<td>3</td>
<td>133</td>
<td>150</td>
<td>112</td>
<td>14</td>
<td>103</td>
<td>32</td>
<td>122</td>
<td>90</td>
<td>92</td>
<td>114</td>
</tr>
<tr>
<td>4</td>
<td>107</td>
<td>77</td>
<td>125</td>
<td>131</td>
<td>112</td>
<td>97</td>
<td>74</td>
<td>66</td>
<td>74</td>
<td>44</td>
</tr>
<tr>
<td>5</td>
<td>117</td>
<td>90</td>
<td>130</td>
<td>126</td>
<td>114</td>
<td>99</td>
<td>89</td>
<td>61</td>
<td>76</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>80</td>
<td>59</td>
<td>53</td>
<td>153</td>
<td>43</td>
<td>105</td>
<td>55</td>
<td>27</td>
<td>25</td>
<td>55</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
<td>13</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>14</td>
<td>7</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>182</td>
<td>156</td>
<td>143</td>
<td>233</td>
<td>118</td>
<td>266</td>
<td>82</td>
<td>87</td>
<td>72</td>
<td>81</td>
</tr>
</tbody>
</table>

Some associations between terms have been identified and the confidence limits are reported in between brackets as can be observed in table 2. It is important to note that some of the terms as shown in table 2 are not recognizable words as the stemming algorithm has removed suffixes in some instances. For example, the term “affect” could be inferred to be “affection” when the texts are read through. The limitation here is that the contexts within which these words are been associated are not present. Nevertheless, some insights about the association can be understood like for the case of “queen” being associated to “grace” and “death” to “enemies.”

Table 2: Words Associations with Confidence Limits.

<table>
<thead>
<tr>
<th>Lord</th>
<th>Love</th>
<th>King</th>
<th>Queen</th>
<th>Death</th>
</tr>
</thead>
<tbody>
<tr>
<td>caus (0.97)</td>
<td>deni (0.85)</td>
<td>lord (0.94)</td>
<td>seek (0.96)</td>
<td>meet (0.90)</td>
</tr>
<tr>
<td>madam (0.97)</td>
<td>affect (0.84)</td>
<td>princ (0.91)</td>
<td>princ (0.86)</td>
<td>majesti (0.88)</td>
</tr>
<tr>
<td>nobl (0.95)</td>
<td>brother (0.90)</td>
<td>murder (0.84)</td>
<td>march (0.84)</td>
<td></td>
</tr>
<tr>
<td>king (0.94)</td>
<td></td>
<td>people(0.82)</td>
<td>strong (0.84)</td>
<td></td>
</tr>
<tr>
<td>traitor (0.94)</td>
<td></td>
<td>grace (0.80)</td>
<td>fit (0.82)</td>
<td></td>
</tr>
<tr>
<td>princ (0.93)</td>
<td></td>
<td></td>
<td>desert (0.82)</td>
<td></td>
</tr>
<tr>
<td>bear (0.92)</td>
<td></td>
<td></td>
<td>enemi (0.82)</td>
<td></td>
</tr>
<tr>
<td>els (0.92)</td>
<td></td>
<td></td>
<td>glori (0.81)</td>
<td></td>
</tr>
<tr>
<td>war (0.90)</td>
<td></td>
<td></td>
<td>train (0.81)</td>
<td></td>
</tr>
</tbody>
</table>

To visually identify the most common terms used across texts by Marlowe, a bar chart is drawn as seen in figure 1. From figure 1, the word which appears most often in Marlowe’s text is “will” followed by “though”, “shall” and “king” in descending order of frequencies.

Figure 1: Bar Chart of Frequent Terms
Apart from the bar chart, a word cloud is also produced to identify the most common words used by Marlowe in his texts. Based on the texts analyzed in this paper, it is clear that Marlowe’s texts discuss a lot about power as the word “king” appears a lot in his texts. Other words that can be identified that are not common are “death”, “love”, “heaven”, “crown”, “soul” and so on. From the tag cloud, it is evident that Marlowe lived in an era where the king, the queen, war, religious conflicts and quest for power were present due to the high importance given to words like “king” and “death”. It is avoided to extrapolate more on the analysis of word cloud as the primary focus stays on the patterns that can be derived from the works of Marlowe using text mining.

![Word Cloud of Marlowe’s text](image)

**Figure 2 Word Cloud of Marlowe’s text**

As a means to identify other non-explicit patterns in the text of Marlowe, a dendrogram which results from the hierarchical clustering of texts mentioned in section 3 is presented in figure 3. Three main agglomerations of text can be identified in the dendrogram. *The Tragical History of Dr. Faustus and The Jew of Malta* form a cluster based on the analysis with strong links as they are located at a high position in the dendrogram. A possible reason for this cluster may be the fact that these two texts are relatively similar as they both are connected to religious conflict and power. Barabas, the main character of *The Jew of Malta*, dominates the action of the play as Dr. Faustus does. Both of them undergo deep personal turmoil. The other agglomeration is about *Tamburlaine the Great* Part I and Part II, which is expected, as both plays are about the same subject. On the other hand, *Massacre at Paris* and *Hero and Leander* cluster together. Marlowe uses the Greek mythology and the personal war about love for the *Hero and Leander*. In *Massacre at Paris* the playwright reproduces the events of the Saint Bartholomew’s Day Massacre of 1572 with religious ideologies at play. However, it is important to note that the strength of the link between these two texts is still low even if they are clustered together as it is located at the bottom of the dendrogram.

![Dendrogram of Marlowe’s texts](image)

**Figure 3: Dendrogram of Marlowe’s texts**
4. Conclusions
The works of Marlowe has been analyzed using text mining algorithms and the main findings related to the frequency of words having a connection with power and war. Besides, the texts of Marlowe can be clustered into three main groups where two groups are evident in terms of their similarities. One cluster is about religion and power, the other cluster relates to a play in two parts with same context. However, the last group is weaker in terms of linkage and may not provide useful insight in terms of findings.

It is shown through this paper that statistics can be a strong supportive discipline in the literature. However, it is important to recognize that text mining algorithms cannot analyze context and cannot understand a text as a human would. Nevertheless, the findings observed through this paper can be used to trigger a person who analyses literature texts to view things in a different setting like understanding the associations of words that are important in a specific text. Statistics can be juxtaposed to literature, therefore bringing literary analysis through an innovative critical lens.

REFERENCES


APPENDIX

catwd("C:/Users/Thoplan/Documents/University/Research/Ruben/textmining")
library(tm)
library(SnowballC)
library(ggplot2)
library(wordcloud)

The Following Codes Read the Books of Marlow in R

# The Tragedy of Dido Queene of Carthage
a <- readLines(file("C:/Users/Thoplan/Documents/University/Research/Ruben/textmining/16169-0.txt"))

# The Tragical History of Dr. Faustus
b <- readLines(file("C:/Users/Thoplan/Documents/University/Research/Ruben/textmining/pg811.txt"))
# The Jew of Malta
c <- readLines(file("C:/Users/Thoplan/Documents/University/Research/Ruben/textmining/pg901.txt"))
# Tamburlaine the Great, Part I
d <- eadLines(file("C:/Users/Thoplan/Documents/University/Research/Ruben/textmining/pg1094.txt"))
# Tamburlaine the Great, Part II.
e <- eadLines(file("C:/Users/Thoplan/Documents/University/Research/Ruben/textmining/pg1589.txt"))
# Massacre at Paris
f <- eadLines(file("C:/Users/Thoplan/Documents/University/Research/Ruben/textmining/pg1496.txt"))
# Hero and Leander
g <- readLines(file("C:/Users/Thoplan/Documents/University/Research/Ruben/textmining/pg18781.txt"))
# Edward II. Marlowe's Plays
h <- readLines(file("C:/Users/Thoplan/Documents/University/Research/Ruben/textmining/pg20288.txt"))

The Following Codes Prepare the Data Read Before the Corpus is Loaded
text1 <- a[-c(1:182,2378:2749)]
text2 <- b[-c(1:136,3883:4249)]
text3 <- c[-c(1:171,4673:5059)]
text4 <- d[-c(1:237,4289:4656)]
text5 <- e[-c(1:125,4309:4676)]
text6 <- f[-c(1:151,2291:2657)]
text7 <- g[-c(1:47,894:1257)]
text8 <- h[-c(1:79,3075:3446)]
combinetext <- c(text1,text2,text3,text4,text5,text6,text7,text8)
marlowe <- paste(combinetext,collapse=" ")
marlowe <- strsplit(marlowe,"DOCUMENTSEPARATOR")[[1]]

Loading the Corpus
doc <- Corpus(VectorSource(marlowe))

The Following Codes Clean the Corpus
doc <- tm_map(doc,tolower)
doc <- tm_map(doc,removeNumbers)
doc <- tm_map(doc,removePunctuation)
doc <- tm_map(doc,removeWords,stopwords("english"))
doc <- tm_map(doc,stemDocument)
doc <- tm_map(doc,removeWords,c("reads","enter","exeunt","footnot","chorus","act","scene","exit","sestiad" ))
doc <- tm_map(doc,stripWhitespace)

Document Term Matrix (DTM)
matrix <- DocumentTermMatrix(doc)

Deriving Most Frequent Terms from DTM
frequency <- colSums(as.matrix(matrix))
order <- order(frequency)
frequency[tail(order)]
Reducing the sparcity of the DTM
sparse<-removeSparseTerms(matrix,0.2)
m<-as.matrix(sparse)
write.csv(m, file="Matrix.csv")

Finding Frequent Terms After Reducing the sparcity of the DTM
findFreqTerms(sparse,lowfreq=200)

Finding Associations Among Terms
#Association with words with correlation limit of 0.9
findAssocs(sparse,"lord",corlimit=0.9)
findAssocs(sparse,"love",corlimit=0.8)
findAssocs(sparse,"king",corlimit=0.9)
findAssocs(sparse,"queen",corlimit=0.8)
findAssocs(sparse,"death",corlimit=0.8)

Obtaining Bar Chart Showing Frequent Terms
frequency<-sort(colSums(m),decreasing=TRUE)
wordfreq<-data.frame(word=names(frequency),freq=frequency)
p<-ggplot(subset(wordfreq, freq>300),aes(word, freq))
p<-p+geom_bar(stat="identity")
p<-p+theme(axis.text.x=element_text(angle=45,hjust=1))
p<-p+theme(panel.background = element_blank())

Obtaining Word Cloud Showing Frequent Terms
set.seed(2)
wordcloud(names(frequency),frequency,scale=c(5,0.1))

Unsupervised Learning of Words to Determing Clusters
dMat<-dist(scale(sparse))
fit<-hclust(dMat,method="complete")
plot(fit)