Part V - Find Similar Themes with Reinert Textual Data Clustering

Rees W. Morrison

2022-02-04

Concluding this fifth post in a series of notes on Theme centrality and complexity, I turn to the Reinert textual clustering method. To invoke it, I removed portions of blog posts that are similar to all posts, as well as blogdown header material, and then combined all the posts for a Theme into a single "text." Accordingly, 24 texts. Then I used R's rainette package to perform Reinert textual clustering of those 24 texts.

Here are some characteristics of the method:

- it uses a **singular value decomposition** (SVD) of the **document-term matrix** (DTM) created from the corpus (collection of all 24 texts) [more on those two methods in later posts]
- it assigns each text (Theme) to only one cluster
- it is better suited for small "homogeneous" documents

The Reinert algorithm carries out divisive hierarchical clustering (as distinct from Part IV's agglomerative clustering) with the goal to maximise the inter-cluster **Chi-squared distance** between clustered texts. The algorithm works its magic on the DTM, which only stores 1's or 0's (to show the presence or absence of terms in a text), not the frequencies of terms.

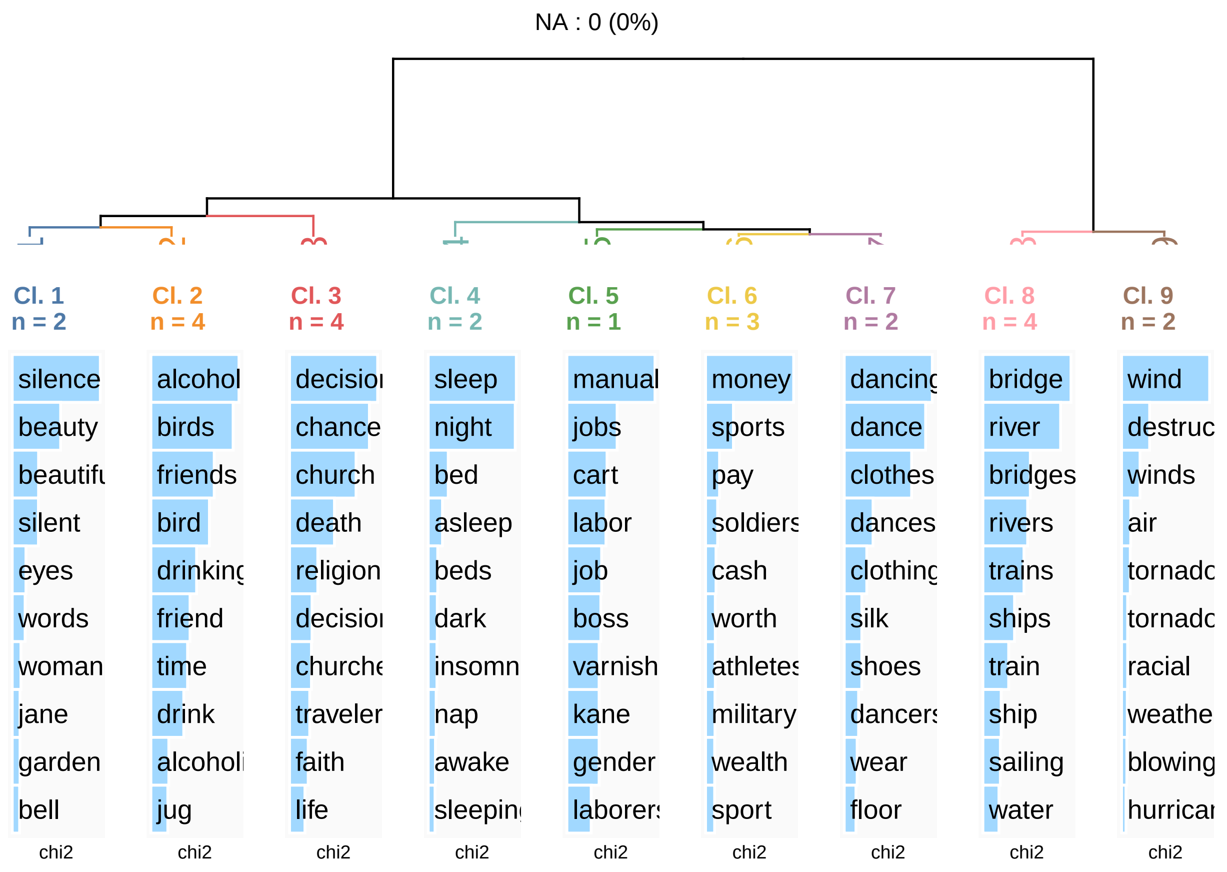
The algorithm splits the DTM into two clusters by maximizing the Chi-squared distance between them. To do so, it carries out the following steps:

- first, Themes are ordered according to their coordinates on the first axis of the **correspondence analysis** (CA) of the binary TDM (see DisplayR for a clear explanation of CA;
- next, Themes are grouped in two clusters based on this order, and the grouping with the maximum inter-cluster **Chi-squared distance** is kept;
- based on this grouping, each Theme is assigned in turn to the other cluster. If the new assignment gives a higher inter-cluster Chi-squared value, the Theme is kept in the other cluster. The operation is repeated until no new assignment gives a higher Chi-squared value;
- on the resulting clusters' binary matrices, features are selected based on their frequency and on a contingency coefficient minimum value[1]; and
- the largest of the two resulting clusters is then split with the same algorithm.

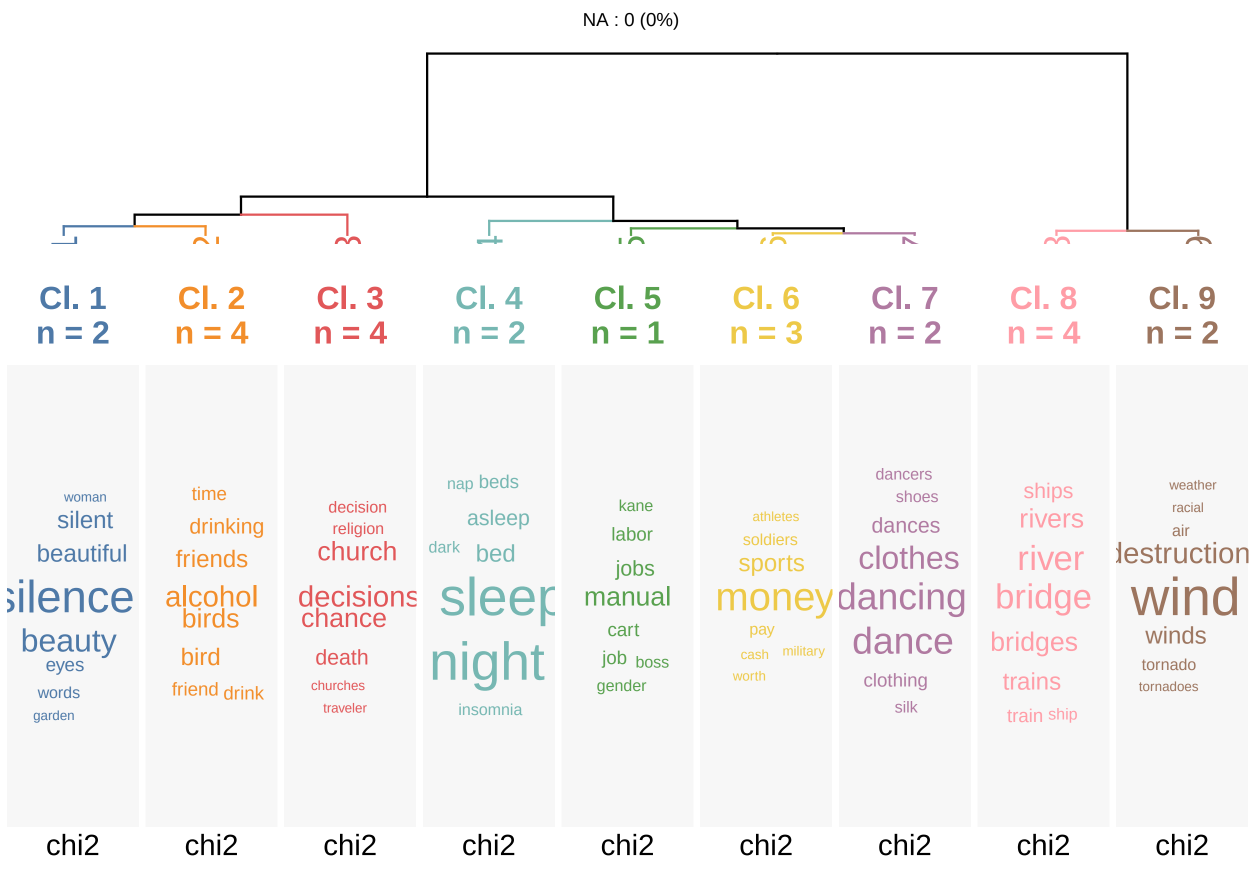
According to the calculations of the rainette package, the strongest associations (the highest Chi-squared values) are between Night and Sleep (almost 1100, in Cluster 4). The next time we use the Reinert text clustering method we will standardize (lemmatize) the Theme variants, such as converting "dancing" to "dance" and adjust plurals, such as "soldiers" to "soldier."

[1]A measure of association based on chi-square. The value ranges between 0 and 1, with 0 indicating no association between the text (row) and term (column) variables and values close to 1 indicating a high degree of association between the variables.

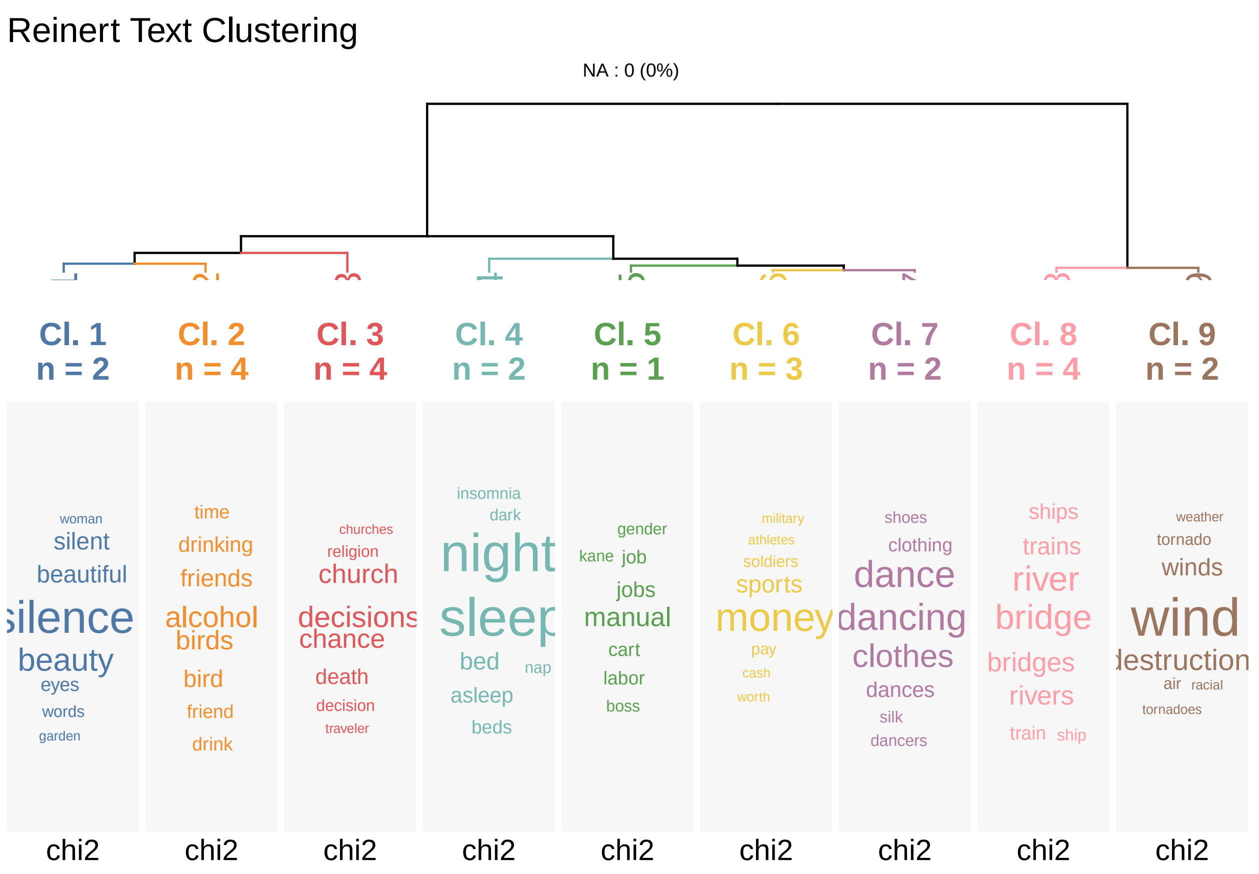
The plot below displays the maximum of nine clusters that the algorithm can handle. The length of the bars visible behind the 10 terms that most represent a cluster correspond to the Chi-squared value. The “n =” row tells how many Themes comprise that cluster. The small dendrogram atop the cluster table visualizes the divisive, top-down methodology.



In a different vizualization of the same clustering outcome, the next plot presents with a “wordcloud” style, where the size of the word matches its Chi-squared value relative to the cluster. It shows only 8 key terms, for legibility.



```
## TableGrob (3 x 9) "arrange": 10 grobs
##     z     cells    name           grob
## 1   1 (1-1,1-9) arrange gtable[layout]
## 2   2 (2-3,1-1) arrange gtable[layout]
## 3   3 (2-3,2-2) arrange gtable[layout]
## 4   4 (2-3,3-3) arrange gtable[layout]
## 5   5 (2-3,4-4) arrange gtable[layout]
## 6   6 (2-3,5-5) arrange gtable[layout]
## 7   7 (2-3,6-6) arrange gtable[layout]
## 8   8 (2-3,7-7) arrange gtable[layout]
## 9   9 (2-3,8-8) arrange gtable[layout]
## 10 10 (2-3,9-9) arrange gtable[layout]
```



Combine the pairings from the three clustering methods

Once the closest pairs were obtained by k-means clustering, agglomerative hierarchical clustering, and Reinert text clustering, I combined them into a dataframe. The first inquiry was whether any Theme is closest to any other Theme more than once (or even by all three algorithms). Each letter pair in column "Two" of the table below tells us that two algorithms found that pair of Themes to be closest, e.g., Birds and Chance (both A), Bridges and Rivers (both B). The "One" column tells how many other Themes were also identified for that Theme as closest. So, for example, Dancing paired twice with Wind (both marked with D in column "Two") and with four other Themes – the "4" in column "One." The Themes in column "Single Closest Themes" had no pair, only single instances of closest Themes. Thus, Money was "closest" to seven Themes, according to column "Single."

Bear in mind that my method of picking closest Themes from the agglomerative hiearchical clustering (Part IV)[IV] yielded multiple "closest" Themes as I tied each agglomerated Theme to each of the Themes already in a unit (if that were the case). In the series of posts to come, I will refine my method and thereby reduce the number of pairings.

Table 1: Close Themes

Two Closest Themes	Two	One	Single Closest Themes	Single
birds	A	5	money	7
chance	A	4	alcohol	5
bridges	B	4	beauty	5
rivers	B, F	2	decisions	5
death	C	5	friends	5
churches	C	4	soldiers	5
dancing	D	4	clothes	4
wind	D	3	nights	4
destruction	E	2	silence	4
sailing	E	2	sports	4
sleep	F	2	trains	4
work	G	2		
time	G	1		

Upcoming Post Series

Once the blog Themes from Art reaches 30 Themes (which it did in early February 2022), incorporates new metrics (i.e., newspaper references and Bing searches), drops the most dubious metrics (college majors and readability), reconsiders the rank metrics (OED Frequency Bands and Top 1000 Words), and **lemmatizes** Theme terms (standardizes word forms), the second series of five posts is likely to address:

- Part VI: Further Observations on Themes and Metrics
- Part VII: Find Similar Themes with Cosines
- Part VIII: Find Similar Themes with Latent Dirichelet Allocation (LDA)
- Part IX: Find Similar Themes with Embeddings and Transformations
- Part X: Including Results of Above Algorithms into Closest Theme-Pairs Analysis

By the end of May 2022, a third series of five analytic posts may encompass 36 Themes, rely on a modified set of metrics (e.g., movie synopses, figurative expressions, improved consistency with singular and plural forms of the Themes), and address the following topics:

- Part XI: Critique of Quantifying Centrality and Complexity in Themes
- Part XII: Find Similar Themes with Principal Components Analysis (PCA)
- Part XIII: Find Similar Themes with Latent Semantic Analysis (LSA)
- Part XIV: Find Similar Themes with Correspondence Analysis (CA)
- Part XV: Further Refinement of Closest Theme Pairs Analysis

A fourth series of posts in the Summer of 2022 may incorporate as many as 40 Themes. Among its potential topics, we envision:

- Part XVI: More Clustering Techniques
- Part XVII: Metaclustering, Validation Tests, Assumptions of the Models
- Part XVIII: Structural Topic Models (STM)
- Part XIX: Other Tools for Analyzing Relationships between Themes
- Part XX: Assessment Closest Theme-Pair Findings to Centrality and Complexity