Text Mining: Clustering Novels

Fraser Walker (001219429)

STATS 780 Final Project

McMaster University

2018

March

Abstract

The goal of this report is to investigate if famous authors of fiction write in a distinctive, recognizable way. Twenty books from four authors (five each) are collected and used as data, then unsupervised clustering techniques are applied. Partitional techniques such as k-means and k-medoids outperformed hierarchical techniques like agglomerative or divisive hierarchical clustering. Principal Component Analysis (PCA) was used to reduce the unwieldy dimensions of the Document Term Matrix (DTM), ultimately having mixed results when used for clustering. In addition to the standard Manhattan, and Euclidean distance metrics, cosine similarity is introduced and performs well on the text data. K-medoids performed on the full DTM with cosine (dis)similarity achieved perfect classification. Conventional metrics to evaluate clustering results, however, are inconclusive. The results indicate that the clustering methods can identify books written by the same author, but more investigation is required.

1 Introduction

The great authors of fiction throughout history are revered for their command of language. Critics and academics will refer to an authors writing style as being uniquely their own and pervasive through his/her works. One could easilly identify the characters of Skimpole, Smallweed, or Tulkinghorn in *Bleak House* as belonging to Dickens, known for his peculiar names, even if they've never read the book. However, is it true that these authors have a distinctive vocabulary and consistent writing style? The complete unabridged texts of 20 books - five each from Charles Dickens, William Shakespeare, Jane Austen, and Fyodor Dostoyevsky - are treated as unlabelled data and the goal is to cluster together books written by the same author. Since there are only 20 books total, the question is treated as an unsupervised clustering problem as oppose to a classification problem because there aren't enough observations (books) to train an algorithm. So an ancilliary question becomes, can an algorithm recognize that four is the most appropriate number of clusters? In the case of Dostoyevsky,

who wrote in his native Russian, all of his works are translated into English. Thus, the data is not representative of his original vocabulary, rather the chosen vocabulary of the translator to represent his work in English. The translator from work to work isn't consistent either, including Constance Garnett, C.J. Hogarth, and Eva Martin. For Dostoyevsky, the question extends to whether or not a translation can do justice to his style (dissimilarity), and if the style is consistent from translator to translator (similiarity). K-means, k-medoids, and agglomerative and divisive hierarchical clustering were the primary clustering algorithms used in this report.

2 The Data

Five books each from four authors: Charles Dickens, Jane Austen, William Shakespeare, and Fyodor Dostoyevsky, were collected from Project Gutenberg [7], a resource whose goal is to make literature in the public domain freely available. Every book is available in English and in the case of Dostoyevsky, this means his works have been translated from Russian. The data is given as text, and needs to be manipulated into numerical data so that it may be used in a clustering algorithm. A conventional way to achieve this is with a **Document Term Matrix (DTM)** (see Table 1). A DTM is a matrix representation of the corpus (the entire collection of books) where each document (book) is one row/observation, and each unique term (word) present in the corpus is one variable/column. The $(i, j)^{th}$ entry represents the presence of the j^{th} term in the i^{th} document, and the value depends on the weighting used. The most intuitive weighting is Term Frequency (Tf): $f_{i,j} = k$ means term j appears k times in document i. For example, in Table 1 the word *abode* appears six times in *Mansfield Park* and only once in *A Christmas Carol*.

The most popular weighting scheme is Term Frequency Inverse Document Frequency (TfIdf) [5]. A TfIdf scheme for a DTM A is given by:

$$(a_{i,j}) = f_{i,j} \cdot \log \frac{N}{n_j} \tag{1}$$

where $f_{i,j}$ is the term frequency, N is the total number of documents in the corpus,

Table 1: Example head of a 20×35486 Document Term Matrix with Term Frequency weighting (Tf).

	abed	**abels**	**abject**	**abode**	**abrahams**
A Christmas Carol	1	1	1	1	1
A Tale of Two Cities	2	0	1	1	0
Northanger Abbey	0	0	0	1	0
Mansfield Park	0	0	0	6	0
Emma	0	0	0	1	0

and n_j is the number of documents the given term appears in. The Inverse Document Frequency alteration is to filter out terms that are common to the whole corpus and emphasize rarer terms that are more specific to just a few documents. A term that appears in many documents will have its weight reduced more than a rare term that only appears in a few. TfIdf will be the only weighting scheme used in this report.

Preprocessing is an essential element when handling raw text data. Each document being a novel or a play, is quite long and this could naturally lead to $100,000$ (or more) unique terms present, and ultimately a very large and very sparse DTM. Standard techniques used to reduce the dimension size include changing all capital letters to lowercase, otherwise for example 'Beyond' and 'beyond' would be considered two unique terms, just because one might have been used at the beginning of a sentence. Another common technique is to remove what are referred to as "stop words". Stop words are terms like *the*, *a*, or *he*: they are potentially present hundreds of times in virtually every document in any kind of text corpus, and do not offer much information. *R* has a standard stop words library of over 1000 terms, and removing these helps remove noise from the data. After this initial cleaning step, Figure 1 shows the most common terms in the corpus.

Even after this initial preprocessing, the DTM is still quite sparse - being mostly filled with zeros. Another problem that arises when calculating the distance between two books is length. The number of unique terms in the data after preprocessing

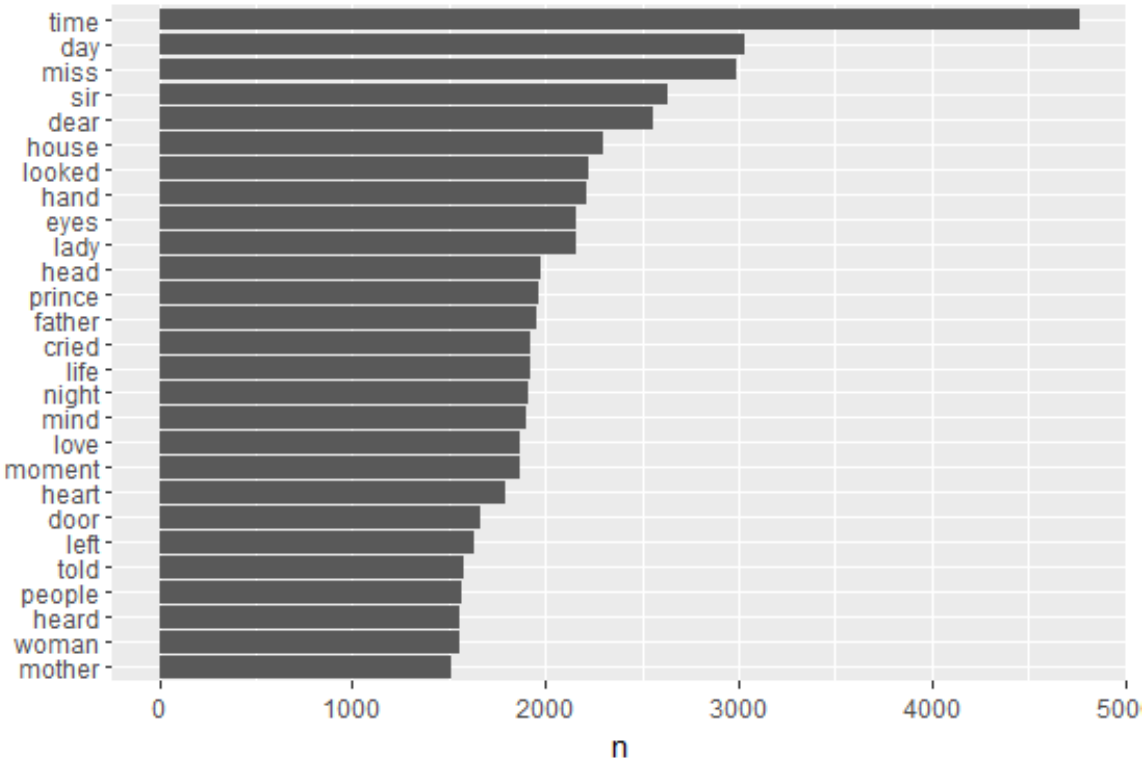


Figure 1: Most frequent terms after removing stop words, and changing to lowercase [3]

for each book ranges from 2586 (*Much Ado About Nothing*) to 14615 (*Bleak House*). *Notes from the Underground*, a novella, is considerably shorter than *The Brothers Karamazov*, and has less unique terms as a result (4290 vs. 11615, respectively). Although both were written by Dostoyevsky, a conventional metric like the Euclidian distance might ignore similarities in the vocabulary because *The Brothers Karamazov* has almost three times as many non-zero entries than does *Notes from the Underground*. To account for different lengths, the document vectors will have to be normalized [5]. A popular metric in document clustering is the *Cosine Similarity* [5]: given two documents d_1 and d_2, their cosine similarity is:

$$cosine(d_1, d_2) = \frac{d_1 \bullet d_2}{\|d_1\|\|d_2\|} \tag{2}$$

Where $\bullet$ represents the dot product. Cosine similarity calculates the angle between

the two documents in high dimensional space. The benefit cosine similarity has over a metric derived from a p-norm in this instance, is that two documents that use similar vocabulary in similar proportions will have a small angle between them, even if one document is significantly longer than another. This property makes cosine similarity - in general - a useful distance metric in text clustering/classification.

3 Principal Component Analysis

Terms that frequently appear together are going to be correlated with each other. Of the 24 times the term *tiny* appears in *A Christmas Carol*, for example, it is followed by *tim* 22 times - Tiny Tim being a prominent character in the story. Since *tim* only appears in *A Christmas Carol*, if *tiny* is already present in the DTM, then the contribution of *tim* in clustering the books is lessened because of this correlation (this information is already captured by *tiny*). Principal Component Analysis (PCA), a dimension reduction technique, is a way to address the inherent sparsity in a DTM, and also capture correlated terms at the same time. PCA uses orthogonality transformations to transform a set of correlated variables into a new set of uncorrelated variables called principal components. The first principal component captures the 'direction' of most variation in the data, the second principal component captures the direction of most variation conditional on the direction being orthogonal to the first principal component. The process continues in this way with each new principal component being orthogonal to all the previous ones.[2]

The original DTM has 35,486 variables (unique terms). When PCA is applied to the corpus, the first principal component is credited with accounting for 18.5% of the variance in the data, and the first fourteen principal components cumulatively account for 90% of the variance. This is already a significant proportion, and using PCA can not only reduce the number of variables from $35,000+$ to less than 20, but it also has the potential to improve clustering performance by capturing information stored in correlated terms.

4 Clustering methods

The first clustering technique, **k-means** aims to cluster n observations ($n = 20$ for this, and all future methods) into k groups, such that the within cluster sum of squares between the observations in a cluster S_i and the clusters *centroid* (mean $\boldsymbol{\mu_i}$) is minimized: $\forall i \in 1, .., k$

$$min \sum_{\boldsymbol{x} \in S_i} \|\boldsymbol{x} - \boldsymbol{\mu_i}\|^2 \tag{3}$$

Closely related to k-means is **k-medoids** clustering. In k-medoids, instead of calculating a centroid $\boldsymbol{\mu_i}$ (which doesn't correspond to an individual datapoint) as in k-means, one point is designated a *medoid* for each cluster and is considered representative of the cluster. K-medoids is considered more robust than k-means for at least two reasons. One, k-medoids seeks to minimize the sum of pairwise dissimilarities between points in the cluster and the medoid vs. the total mean squared error in k-means. Two, because observations with extraordinary values don't disrupt a medoid as they would a centroid in k-means [6]. Consider *Bleak House* which has 14610 non-zero values (unique terms) in the DTM, 3000 more than the next most *The Brothers Karamazov*. During the k-means algorithm, *Bleak House* could possibly shift the centroid of its assigned cluster so much as to obscure clearer relationships in the data. Both k-means and k-medoids are partitional methods, as they create one partition of the data according to the choice of k, chosen beforehand [1].

Hierarchical clustering techinques, by contrast, create a nested series of solutions for k from 1 to n. **Agglomerative** and **divisive** hierarchical clustering are two related but 'opposite' techniques. In agglomerative clustering, every obseration begins as being assigned to its own cluster. Then, the closest pairs (according to some distance/similarity metric) are combined. The process iterates in this way until every observation is part of the same single cluster. Conversely, in divisive clustering, every observation begins in the same cluster, and the largest cluster splits into two. This process repeats until every observation is in its own cluster. Both processes

will use *Ward's method* linkage to define dissimilarity from other clusters, so as to avoid chaining in the resulting dendrograms. Unlike k-means/medoids, the number of clusters isn't chosen beforehand, instead determined 'by eye' after looking at a dendrogram [1].

5 Results and Discussion

Figure 2 [1] and Figure 3 [2] plot the terms of Shakespeare and Dostoyevsky respectively, as percentages of the particular authors total term count, against other authors. If a term isn't common to both authors in a plot (ie: at least one author has not used the word in any document), that term isn't plotted. When plotted with Shakespeare, all of Dostoyevsky, Austen, and Dickens have considerably sparse plots than when plotted with each other. Shakespeare's works predate the others by over 200 years, and his English is understandably 'older'. In Figure 2, *thou*, *tis*, *doth*, *hast* etc. are all plotted above the dotted line (representing $y = x$). Terms around the line are used with the same relative frequency in both authors' works. This means that Shakespeare unsurprisingly uses old English terms like *thou* far more often than the others.

In Figure 3, *prince* and *god* are featured prominently above the dotted line corresponding to Dostoyevsky's relatively frequent terms. Prince Myshkin is the principal character in Dostoyevsky's *The Idiot*, but *prince* was also a title for Russian nobility during Dostoyevsky's life (1821-1881) and a common social class in his works. *Baron* and espeically *count* also appear more often in Dostoyevsky and Shakespeare's works (royalty/nobility) than in Dickens and Austen who use *lady*, *sir*, and *miss* more. Dickens and Austen often wrote about poor, lower class people, highlighting a cultural difference between 19^{th} century Russian and English literature. Dostoyevsky's works often explored religious and philisophical themes, and the religious terms *holy* and *god* appear more often in his works than they do in Dickens' or Austen's work.

[1] made with help of look-up table function from dfalster [4]

[2] see previous footnote

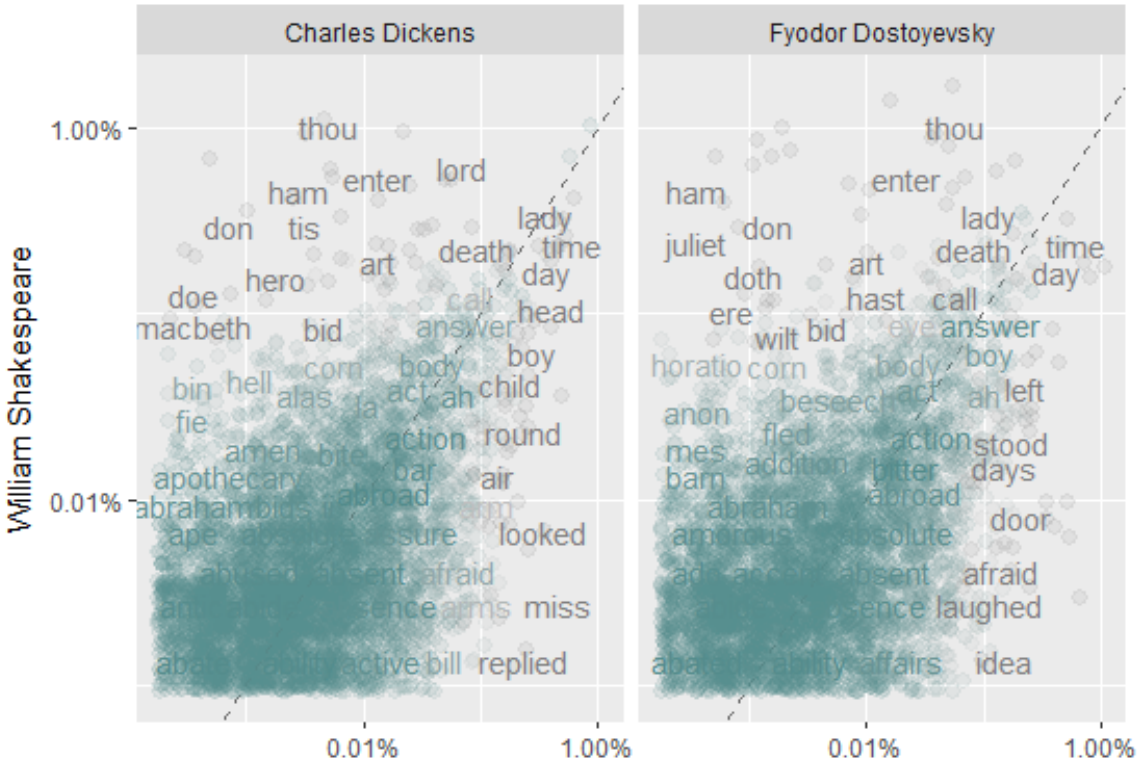


Figure 2: Shakespeare term percentages plotted against Dickens, Dostoyevsky on a logarithmic scale [3]

For the hierarchical clustering techniques, divisive clustering performed particularly poorly. No matter whether the full DTM was used or the reduced matrix (PCA with fourteen components - 90% of the variation in the data), no matter which metric was chosen to define similarity (Manhattan, Euclidean, cosine), all dendrograms experienced severe chaining (Figure 4(b)). When $k = 4$ groups were chosen, 16-17 of the 20 observations were consistently placed in one category, while the remaining 3-4 were spread among the other three groups. Agglomerative clustering by comparison was more successful. Cosine similarity consistently performed better than Manhattan and Euclidean distance, on both the full DTM and the reduced one. PCA had inconsistent effects on performance. Agglomerative clustering with Euclidean or manhattan distance on the full DTM clustered better than on the PCA reduced DTM. However, agglomerative clustering with cosine similarity on the PCA reduced matrix

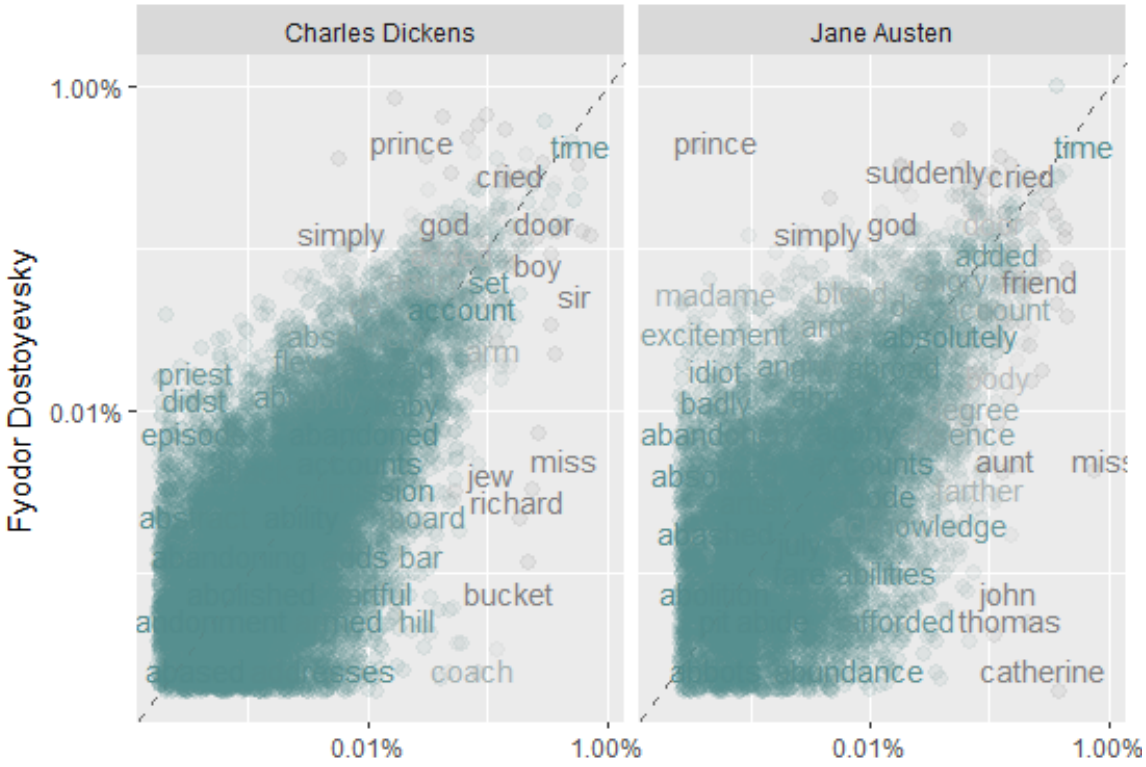
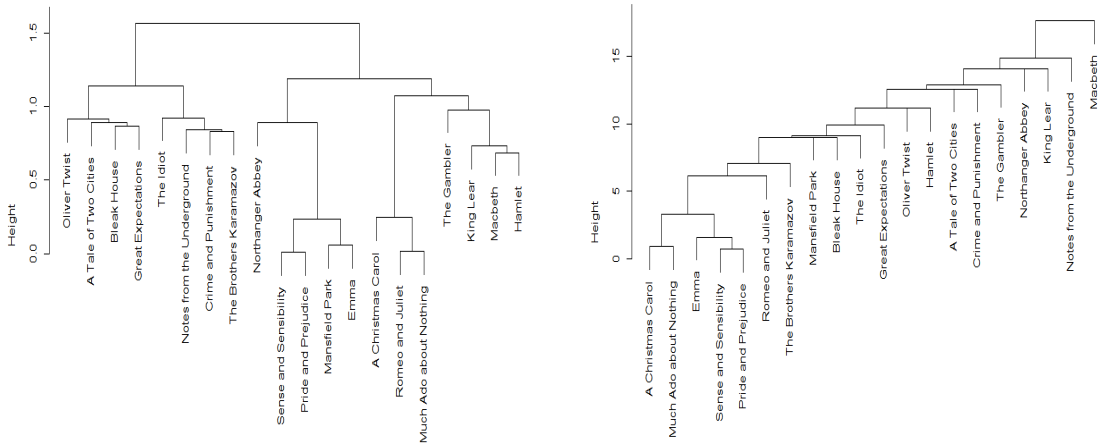


Figure 3: Dostoyevsky term percentages plotted against Dickens, Austen on a logarithmic scale [3]

(henceforth simply referred to as agglomerative clustering) was the only hierarchical clustering method not to fit at least ten observations in any one cluster, the largest being seven (each author has five books, the cluster sizes should be even) (Figure 4(a)).

The agglomerative clustering technique had a classification rate of 90% only misclassifying two books: *A Christmas Carol* and *The Gambler*. The classification table is shown in Table 2. It should be noted that both the misclassified books are among the shortest of their respective authors sampled works (Dickens and Dostoyevsky), and they have been grouped with Shakespeare's plays; on average among the shortest of the works sampled. The Adjusted Rand Index (ARI), a variation of the Rand Index which looks at the ratio of pairwise agreements to total pairs of observations, accounts for chance agreements that might occur randomly so the expected ARI for



(a) Agglomerative hierarchical clustering with cosine similarity, after PCA

(b) Divisive hierarchical clustering with cosine similarity, after PCA

Figure 4: Comparing hierarchical clustering techniques and an example of the 'chaining' problem

a random classification is 0. The ARI for the agglomerative clustering technique is 0.707, fairly high.

Table 2: Classification table for agglomerative hierarchical clustering technique using cosine similarity after applying PCA

		Predicted Value			
		Group 1	Group 2	Group 3	Group 4
True Value	Austen	5	0	0	0
	Shakespeare	0	5	0	0
	Dostoyevsky	0	1	4	0
	Dickens	0	1	0	4

The k-means, and k-medoids family of clustering algorithms were demonstrably better than hierarchical clustering. K-means performed on the full DTM with $k = 4$ clusters had a 90% classification rate, missclassifying two different books than the agglomerative clustering algorithm: Shakespeare's *Much Ado About Nothing* and *Romeo and Juliet* as belonging to Charles Dickens. The ARI for k-means with $k = 4$ is 0.756,

about a 0.5 improvement over agglomerative hierarchical clustering. If k-means is run with $k = 5$, then both Shakespeare plays are put into their own separate cluster, giving five homogeneous clusters (Shakespeare is split into two distinct clusters) (Table 3). Curiously, despite fairly accurate classification and cluster homogeneity for k-means, when the 'total sum of squares within clusters' distance is plotted against k, as in an elbow plot (Figure 5), no obvious elbow emerges for k from 1 to 20 (where $k = 20$ represents every book being its own cluster). If four natural clusters existed in the data, we'd hope for a sharp 'kink' to appear in Figure 5 at $k = 4$, instead the total within cluster sum of squares distance seems to decrease (roughly) linearly as k increases.

Table 3: Classification Table for k-means clustering with $k = 5$ performed on the full DTM

		Predicted Value				
		Group 1	Group 2	Group 3	Group 4	Group 5
True Value	Austen	5	0	0	0	0
	Shakespeare	0	3	2	0	0
	Dostoyevsky	0	0	0	5	0
	Dickens	0	0	0	0	5

K-medoids clustering with $k = 4$ performed on the full DTM had the best performance of all methods, achieving a perfect classification rate, and thus an ARI of 1.0. The perfect classification was achieved with a Euclidean metric to define dissimilarity between observations and the medoid, however the perfect classification was also achieved using cosine distance as well (the Manhattan distance did not perform well in this case). PCA had a strictly negative impact on classification for both k-means and k-medoids. Figure 6 presents a plot of silhouette widths of the k-medoids results with $k = 4$ and cosine distance used for dissimilarity. The silhouette width of an observation s_i is defined as:

$$s_i = \frac{b_i - a_i}{max\{a_i, b_i\}} \tag{4}$$

where a_i is the average dissimilarity of obs. i with the other observations in its cluster, and b_i is the average dissimilarity of obs. i with the observations in its "closest" neighboring cluster [6]. A high silhouette width, close to 1, implies an observation is well clustered: near like points and far away from unlike points. Strangely, despite the perfect classification rate, the highest silhouette width is 0.166 achieved by *Hamlet*, with the average silhouette width of all books being 0.03 - very low. Figure 6 implies every book is poorly clustered, as well as none of the clusters being particularly well defined, with Shakespeare's cluster presenting the highest average silhouette width with 0.1. This suggests that the data is very difficult to separate, that the books aren't that dissimilar from one another. However it is also highly unlikely that the k-medoids algorithm chanced upon a perfect classification. Silhouette widths were also generated based on the results of agglomerative hierarchical clustering done on the PCA reduced DTM (keeping the first 14 principal components), and the average silhouette width was 0.17, still low, but markedly higher than the 0.03 average silhouette width from k-medoids. The Jane Austen cluster was particularly well defined, with the average silhouette width within the cluster being 0.55 and *Sense and Sensibility* and *Pride and Prejudice* having silhouette widths around 0.7. The PCA reduced DTM has only 14 variables, while the full DTM has over $35,000$. The number of variables present may be limiting the scope of silhouette width as an explanatory measure.

6 Conclusion

The partitional family of clustering methods (k-means, k-medoids) outperformed the hierarchical family (agglomerative, divisive) in clustering the books based on author. K-medoids performed on the full DTM with Euclidean distance (or cosine distance) to define dissimilarity achieved a perfect classification rate - the best of all techniques.

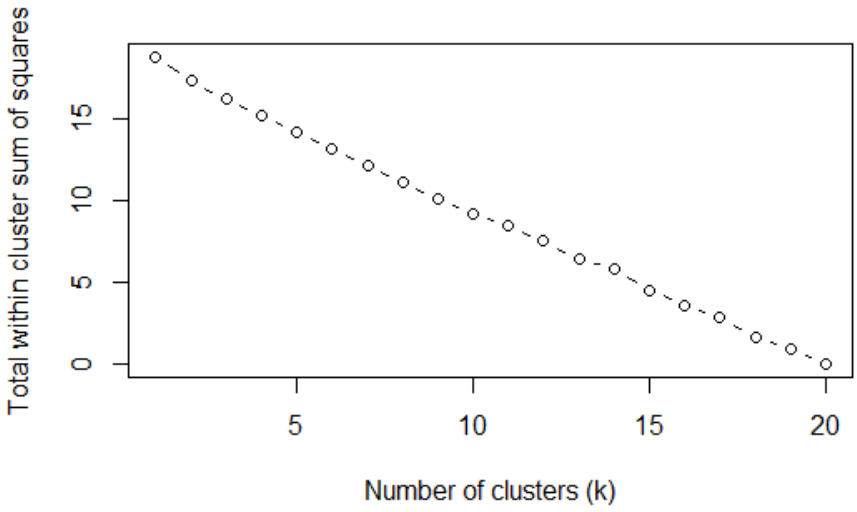


Figure 5: Elbow plot for k-means performed on the full DTM. No obvious elbow emerges, especially not near $k = 4$ or $k = 5$. Plot is approximately linear

Solutions resulting from the hierarchical family typically exhibited the 'chaining' effect in their dendrograms. Principal Component Analysis (PCA) had an inconsistent effect when used with the hierarhical clustering methods, and a strictly negative one when used with the partitional methods. Since PCA reduces the DTM from over $35,000$ variables to less than 20, too much information might be lost in the reduction for the data to remain viable. Cosine (dis)similarity was generally the best distance function whether it was used to define distance (hierarchical clustering) or dissimilarity (partitional clustering). Despite the perfect classification from k-medoids, Figure 5 and Figure 6 remain troubling as both imply that the data is difficult to separate. Figure 5 suggests that the data doesn't naturally contain any clusters, let alone $k = 4$ or $k = 5$, and Figure 6 shows that, despite perfect classification, virtually every book is poorly clustered, and each cluster is not well-defined. This may be a consequence of a shortcoming in the formulation of the problem. Although a classification problem, which generally entails training a model with labelled data (supervised learning), the lack of observations (books) made training a model impossible. Instead, unsuper-

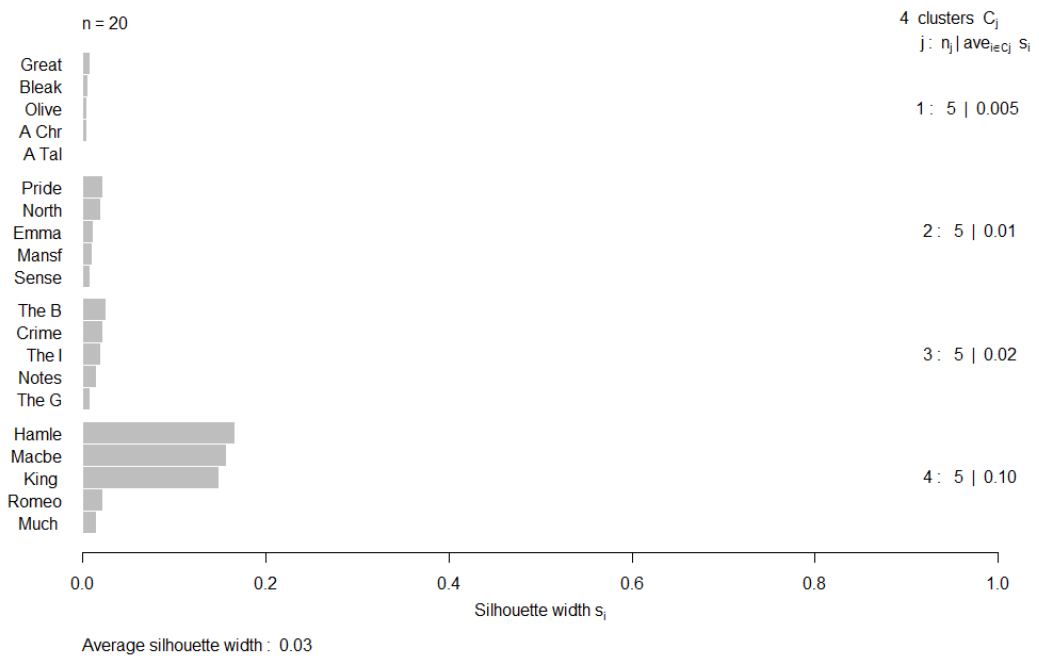


Figure 6: Plot of silhouette widths for k-medoids clustering performed on the full DTM with $k = 4$ and cosine/angular distance for dissimilarity. None of the books are particularly well clustered.

vised learning techniques (clustering) are applied to the data where the algorithms are not given criteria as to what qualifies correct classification. The hope is, if authors do write distinctively and consistently, that the data will naturally cluster itself by author. The perfect classification rate achieved by K-medoids is encouraging, and suggests that each author does indeed have their own distinctive tendencies that can be learned from vocabulary and word frequency. In Dostoyevsky's case, this also means his works are translated distinctively, from Russian. However, further investigation is required to explain why standard explanatory measures such as silhouette width seemingly contradict this conclusion.

References

[1] Dr. Sharon McNicholas. *STATS 780: Data Science, Lecture 12 notes*. Retrieved from `https://ms.mcmaster.ca/ sharonmc/STATS780/lecture12_notes.pdf`

[2] Dr. Sharon McNicholas. *STATS 780: Data Science, Lecture 18 notes*. Retrieved from `https://ms.mcmaster.ca/ sharonmc/STATS780/lecture18_notes.pdf`

[3] Julia Silge, David Robinson. *Text Mining with R: A Tidy Approach*. O'Reilly Media, 2017. Retrieved from `https://www.tidytextmining.com/`

[4] Daniel Falster, `https://gist.github.com/dfalster/5589956`

[5] Michael Steinbach, George Karypis, Vipin Kumar. *A Comparison of Document Clustering Techniques*. University of Minnesota, 2000.

[6] Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K. (2017). *cluster: Cluster Analysis Basics and Extensions*. R package version 2.0.6.

[7] David Robinson (2018). *gutenbergr: Download and Process Public Domain Works from Project Gutenberg*. R package version 0.1.4. `https://CRAN.R-project.org/package=gutenbergr`