GRINREY

Advanced Research in Computer Engineering

Sandip A. Kale
Editor

Research Transcripts in Computer, Electrical and Electronics Engineering

1 Enhanced Text Clustering Approach using Hierarchical Agglomerative Clustering with Principal Components Analysis to Design Document Recommendation System

Gauri Chaudhary[a,*] and Manali Kshirsagar[b]

[a]Department of Computer Technology, Yeshwantrao Chavan College of Engineering, Nagpur, India
[b]Rajiv Gandhi College of Engineering and Research, Wanadongri, Nagpur, India

*Corresponding author: chaudhary_gauri@yahoo.com

ABSTRACT

Considering the increased usage and our increasing dependency in today's world on electronic data, substantial part of which is in textual form, it becomes necessary to devise scientific methods to infer and extract knowledge from such abundant electronic documents for strategic decision making in any target domain under consideration. The purpose of this study is to develop a common platform where all the similar text from multiple source documents from internet can be fetched and grouped using text mining and document clustering techniques. This chapter elaborates the method of hierarchical agglomerative text clustering approach to identify similar groups within documents. The method of Principal Components Analysis on text data is also further elaborated. Further combination of the two methods is proposed to find suitable clusters in text data and the results obtained show better quality clusters. For the purpose of experiments, plot summaries of movies from Wikipedia are used as the source document corpus. Various document pre-processing techniques are also explained and applied to the documents. The proposed method to get suitable clusters of similar movies can be used for recommendation to users. R programming is used for implementation of algorithms and visualization of the results.

Keywords: Data Mining, Hierarchical Agglomerative Clustering, Principal Components Analysis, Text Clustering

Advanced Research in Computer Engineering | ISBN: 978-81-948951-2-1
Book Series: Research Transcripts in Computer, Electrical and Electronics Engineering | Vol: 02

1. INTRODUCTION

With large abundance of electronic data available in today's era of internet, data mining becomes a very sought for field to extract information from such huge data for enhancing business and making business decisions [1]. A specialized field within data mining is text mining which focuses on mining information from textual data. Typical applications of text mining would be in the areas of document classification, sentiment analysis, document clustering, entity relation modeling and others [2].

Textual data is available in huge size in the form of web pages, digital libraries, articles, blogs, emails etc. In order to analyze this data, the data must be first converted into numeric form. Various pre-processing and data transformation techniques need to be applied to convert the textual data into equivalent numeric form. Data in the form of documents from various sources under the target study are collected. Document collection, also called Document corpus, is then subject to various pre-processing techniques. The focus of this study is to study and apply text mining techniques for clustering similar texts.

Clustering is one of the popular data mining functionalities that aims at identifying natural groupings within a data population based on similarity. Every data point is compared to every other data point in the population and based on some similarity measure chosen, they are assigned to groups. Document clustering is the application of clustering algorithms to collection of documents to derive meaningful clusters of documents [3]. This is a form of unsupervised learning since actual cluster labels in the data are not known in advance [4, 5].

The aim of this case study is provide a platform for choosing movie according to one's taste from plethora of options available and without having to search multiple sources. The platform should fetch similar movies based on any criterion like genres, artists, date etc. so that one can choose a movie of his taste almost immediately without having required searching multiple sources.

2. FOUNDATIONS

2.1 Text Mining

Text mining is an area which deals with processing of textual data which is unstructured like free flowing text on web pages, digital libraries, articles etc. to derive some underlying hidden knowledge [6]. These huge document collections which are not structured need to be cleaned and converted into a form which can be used for machine learning. Various pre-processing techniques mentioned in [1, 2, 6] that need to be applied to the documents are listed below:

2.2.1 Tokenization

Each document is subject first tokenization i.e. breaking of the document into collection of words or tokens. This is done typically using some delimiters such as whitespaces.

2.2.2 Remove Stop Words

Majority words in documents are actually noise i.e. words that may not really add any value to the analysis. For e.g. articles, prepositions, conjunctions etc. These are typically called stop words and need to be removed from the documents.

2.2.3 Stemming

Another major activity is stemming i.e. reducing the words to their root form. For e.g. words “running” or “ran” both have the same root word viz. “run”. So they are both replaced with the word “run”. Various stemming algorithms are available to reduce the words to their stems like Porter’.

2.2.4 TF-IDF Conversion

The documents, thus cleaned, need to be converted into numeric form such that they can be fed as an input to the mining algorithms. This can be done using TF-IDF (Term Frequency – Inverse Document Frequency) conversion. In this technique, each document is converted into a vector of words, where each word is represented by a number depicting its importance in the document collection. In the TF-IDF representation, the frequency of each word is normalized by its IDF.IDF reduces the weight of very frequent words that occur in maximal number of documents in the corpus thereby reducing the significance of commonly occurring words in the corpus. Words that occur frequently in a document but are rare across the document corpus are of interest as they would add value when comparing the documents for similarities in the clustering context. This method does so by assigning weighted scores to each word in the documents. For every document, each term is replaced by TF-IDF score explained in [7-9] which is calculated as follows (“doc” in the equations 1 and 2 indicates a document in the corpus):

$$\text{TF ("term")} = \frac{\text{frequency ("term") in a doc}}{\text{total no.of terms in the doc}} \tag{1}$$

$$\text{IDF ("term")} = \frac{\log(\text{total no.of docs})}{\text{no.of docs containing the term}} \tag{2}$$

$$TF - IDF("term") = TF("term") * IDF("term") \tag{3}$$

2.2 Document Clustering

Document clustering aims at identifying segments within a document collection based on similarities [10]. It is a form of unsupervised learning since the groups within the documents are not known beforehand [11]. The model is trained based on the existing documents and its characteristics (typically words) so that groups within the data can be discovered based on similarities. There are various clustering approaches available. In this section we provide an overview of hierarchical clustering technique and Principal Components Analysis.

2.3 Hierarchical Clustering Technique

This technique starts with every document in a single cluster and goes on merging the clusters that are most similar till getting a single cluster. The clustering process and the result are displayed in the form of a tree or a "dendogram". The tree portrays the complete merging process showcasing the intermediate clusters at each level.

There are two different approaches within the hierarchical techniques: agglomerative and divisive. Agglomerative is a bottom-up approach that assumes every document in a single cluster at the bottom and goes on merging the clusters till single cluster at the top. Divisive approach is a top down approach starting with all data points in a single cluster at the top and then these are split on some similarity criterion recursively until every document is in separate cluster. In this section we describe the hierarchical agglomerative clustering (HAC) algorithm as in [12, 13, 14].

Hierarchical Agglomerative clustering algorithm:

i. Initially every document is treated as a separate cluster.
ii. Calculate a distance matrix which depicts the pair-wise similarities between clusters.
iii. Find the pair of clusters that are closest (most similar), remove the pair from matrix and merge them.
iv. Update the distance matrix to reflect the distances between the new cluster and other clusters.

Repeat steps (3) and (4) above until the distance matrix is reduced to a single element.

2.4 Principal Components Analysis

Principal Components Analysis is a method for reducing dimensions in such a way that the information loss is minimal [20, 21]. The basis of PCA is to identify the most significant of the components that capture maximum variance in the data [22]. One of the main objectives is to reduce the redundancy of the data.

Document term matrix used in text mining consists of rows as the document names and columns are the terms in the documents. This matrix is sparse due to the large number of terms. These terms are the dimensions in the case of document corpus and our objective is to reduce this number of terms and identify the most important or principal terms.

New set of terms is obtained by combining the old terms such that the new set of terms is either less than or equal to the number of old terms, indicate the maximum spread in the data and are representative of the original terms as they are derived from original terms. These new terms are called as principal components and are representative of more significant of the terms in the document corpus. Thus with PCA, the less significant terms of the corpus may be dropped thus reducing the dimensions for any text analysis.

3. RESEARCH METHODOLOGY

3.1 Data Collection And Transformation

Based on the motivation of getting movies according to one's taste, the dataset chosen for the purpose of this case study is reading Wikipedia pages of several movies of different genres. The document corpus was formed by adding the plot summaries of various movies from their Wikipedia pages. Movies from following 5 different genres were collected:

(1) Action thriller
(2) Comedy
(3) Animation
(4) Extra terrestrials
(5) Fantasy

The document corpus is shown in Fig. 1.

```
> summary(corpus)
                                  Length Class             Mode
Alice in wonderland.txt           2      PlainTextDocument list
Alice through the looking glass.txt 2    PlainTextDocument list
Avengers Age of Ultron.txt        2      PlainTextDocument list
Avengers Infinity War.txt         2      PlainTextDocument list
Despicable Me.txt                 2      PlainTextDocument list
Die Hard 2.txt                    2      PlainTextDocument list
Die Hard with a Vengeance.txt     2      PlainTextDocument list
Die Hard.txt                      2      PlainTextDocument list
Finding dory.txt                  2      PlainTextDocument list
Finding Nemo.txt                  2      PlainTextDocument list
Little big soldier.txt            2      PlainTextDocument list
Little White Lies.txt             2      PlainTextDocument list
The Avengers.txt                  2      PlainTextDocument list
Trail of the pink panther.txt     2      PlainTextDocument list
```

Fig. 1. Document Corpus

Tokenization, stop words removal and stemming are performed and document term matrix is formed. In this matrix the rows indicate the documents and the columns are all the distinct words. Each cell in the matrix indicates the frequency of the word in the corresponding document. The document term matrix is shown in Fig. 2 . The sparsity is 90% which is high indicating that there are distinct terms in the corpus which are not common, meaning not present in all of the documents. This is good when we compare the documents for similarity in the clustering algorithms.

```
> inspect(dtm)
<<DocumentTermMatrix (documents: 14, terms: 2069)>>
Non-/sparse entries: 2980/25986
Sparsity           : 90%
Maximal term length: 15
Weighting          : term frequency (tf)
Sample             :
                                  Terms
Docs                               alice dory gruber loki marlin mcclane nemo queen stark thanos
  Alice in wonderland.txt             22    0      0    0      0       0    0    17     0      0
  Alice through the looking glass.txt 22    0      0    0      0       0    0     9     0      0
  Avengers Age of Ultron.txt           0    0      0    3      0       0    0     0    10      1
  Avengers Infinity War.txt            0    0      0    1      0       0    0     0     6     23
  Die Hard 2.txt                       0    0      1    0      0      21    0     0     0      0
  Die Hard with a Vengeance.txt        0    0      2    0      0      27    0     0     0      0
  Die Hard.txt                         0    0     22    0      0      30    0     0     0      0
  Finding dory.txt                     0   23      0    0      8       0   10     0     0      0
  Finding Nemo.txt                     0   14      0    0     17       0   15     0     0      0
  The Avengers.txt                     0    0      0   19      0       0    0     0    11      0
```

Fig. 2. Document Term Matrix

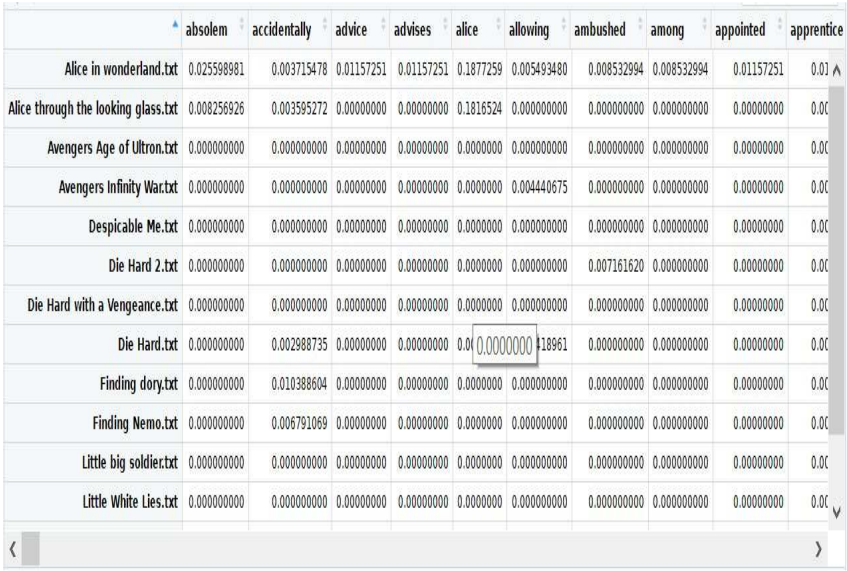
3.2 Conversion to vector space model

After cleaning the document using the pre-processing steps like tokenization, stop words removal and stemming, the corpus is converted to Vector Space Model (TF-IDF representation) as explained in [17]. TF-IDF value for every term in the document is calculated using equation (3) above. This document corpus is transformed into vector of values. Each document vector consists of TF-IDF value for each of the term in the document. Fig. 3 shows a portion of the TF-IDF representation for the documents.

3.3 Similarity Measure

Since clustering is based on grouping of similar documents, some similarity measure must be chosen to compare the documents. There are a number of possible measures for computing the similarity between documents. One of the most widely used is the cosine measure, which is explained in [8, 12, 17, 18] and defined as:

$$cosine(doc1, doc2) = \frac{doc1 * doc2}{|doc1||doc2|} \quad (4)$$



	absolem	accidentally	advice	advises	alice	allowing	ambushed	among	appointed	apprentice
Alice in wonderland.txt	0.025598981	0.003715478	0.01157251	0.01157251	0.1877259	0.005493480	0.008532994	0.008532994	0.01157251	0.01
Alice through the looking glass.txt	0.008256926	0.003595272	0.00000000	0.00000000	0.1816524	0.000000000	0.000000000	0.000000000	0.00000000	0.00
Avengers Age of Ultron.txt	0.000000000	0.000000000	0.00000000	0.00000000	0.0000000	0.000000000	0.000000000	0.000000000	0.00000000	0.00
Avengers Infinity War.txt	0.000000000	0.000000000	0.00000000	0.00000000	0.0000000	0.004440675	0.000000000	0.000000000	0.00000000	0.00
Despicable Me.txt	0.000000000	0.000000000	0.00000000	0.00000000	0.0000000	0.000000000	0.000000000	0.000000000	0.00000000	0.00
Die Hard 2.txt	0.000000000	0.000000000	0.00000000	0.00000000	0.0000000	0.000000000	0.007161620	0.000000000	0.00000000	0.00
Die Hard with a Vengeance.txt	0.000000000	0.000000000	0.00000000	0.00000000	0.0000000	0.000000000	0.000000000	0.000000000	0.00000000	0.00
Die Hard.txt	0.000000000	0.002988735	0.00000000	0.00000000	0.0(0.0000000	18961	0.000000000	0.000000000	0.00000000	0.00
Finding dory.txt	0.000000000	0.010388604	0.00000000	0.00000000	0.0000000	0.000000000	0.000000000	0.000000000	0.00000000	0.00
Finding Nemo.txt	0.000000000	0.006791069	0.00000000	0.00000000	0.0000000	0.000000000	0.000000000	0.000000000	0.00000000	0.00
Little big soldier.txt	0.000000000	0.000000000	0.00000000	0.00000000	0.0000000	0.000000000	0.000000000	0.000000000	0.00000000	0.00
Little White Lies.txt	0.000000000	0.000000000	0.00000000	0.00000000	0.0000000	0.000000000	0.000000000	0.000000000	0.00000000	0.00

Fig. 3. TF-IDF representation of the document corpus

4. EXPERIMENTAL RESULTS

4.1 Optimal number of clusters

There are various methods to find the optimal number of clusters of which few popular ones are listed below:

4.1.1 Elbow method

In this method as explained in [19], the number of clusters is plot as a function of within cluster sum of square distances. The value of K is selected as the point in graph where there is noticeable decrease in sum of square distances i.e. position of a bend in the plot. Fig. 4 shows the plot of elbow method applied to the movie dataset under consideration for this case study. The formation of elbow can be seen at k= 5 or k =6.

4.1.2 Average Silhoute Criterion

Silhoute is a value indicating similarity of a document to its own cluster as against to other clusters. In this method, the average silhouette of documents for different values of K is plot. The desired value of K is the one for which the average silhoute value is maximum as explained in [15]. Fig. 5 shows the average Silhoute plot against the number of clusters. The possible values of good K from the plot could be 5, 7 and above.

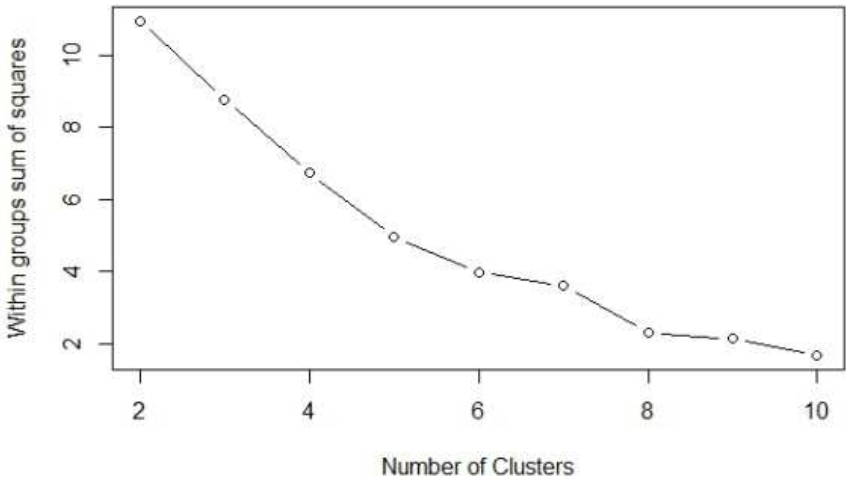


Fig. 4. Elbow method applied to the movie dataset

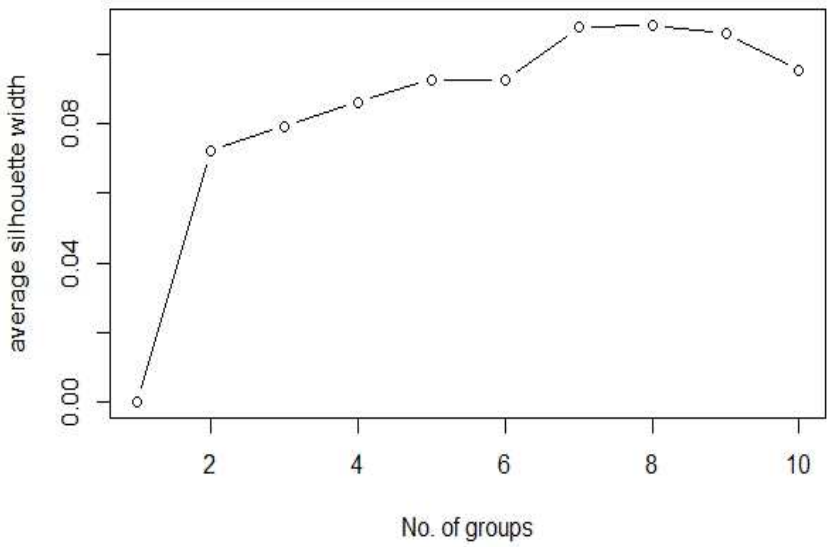


Fig. 5. Average Silhoutte Plot for the movie dataset

4.1.3 Calinski Harabasz

The Calinski Harabasz criterion is also based on low within cluster sum of squares and high between cluster sum of square distances as explained in [15]. The desirable value of K from the plot is the one that shows highest value for the Calinski Harabasz index. Fig. 6 shows the Calinski Harabasz index plot against the number of clusters for the movie dataset considered. Value of K = 5 and above look to be a good choice.

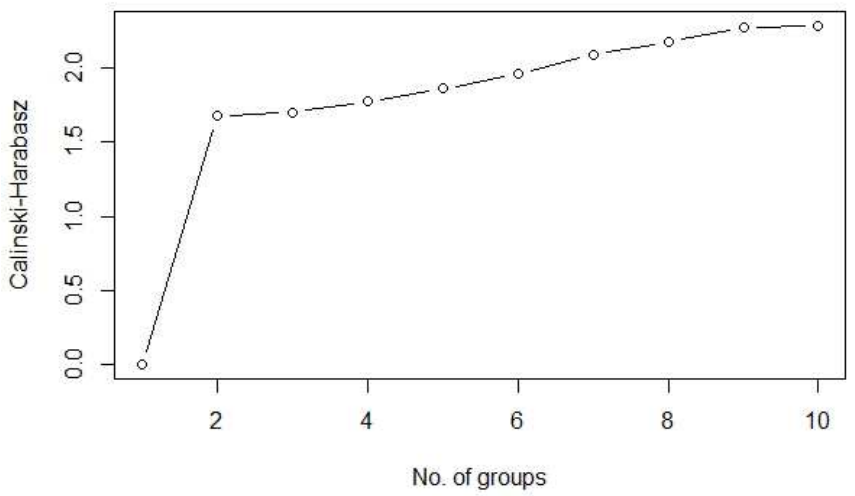


Fig. 6. Calinski Harabasz index for the movie dataset

Based on the above methods for selecting the optimal number of clusters, 5 or higher value is suitable.

4.2 Implementation of hierarchical clustering algorithm (HAC)

HAC is implemented by considering each document in one cluster to start with and then merging the clusters based on cosine similarity.

	Alice in wonderland.txt	Alice through the looking glass.txt	Avengers Age of Ultron.txt
Alice through the looking glass.txt	35.34119		
Avengers Age of Ultron.txt	48.51804	50.86256	
Avengers Infinity War.txt	53.38539	55.29014	45.16636
Despicable Me.txt	45.62894	47.50789	42.49706
Die Hard 2.txt	50.28916	52.95281	47.40253
Die Hard with a Vengeance.txt	58.96609	60.99180	56.50664
Die Hard.txt	58.08614	60.32412	55.91064
Finding dory.txt	50.47772	52.02884	47.83304
Finding Nemo.txt	51.83628	54.03702	49.44694
Little big soldier.txt	38.19686	41.20680	34.48188
Little White Lies.txt	37.38984	40.48456	33.79349
The Avengers.txt	50.87239	53.00000	37.89459
Trail of the pink panther.txt	45.16636	47.46578	42.21374

Fig. 7. Distance Matrix

Distance matrix is used to depict the similarity between two clusters whose ij^{th} element expresses the distance between the i^{th} and j^{th} cluster. Fig. 7 shows the distance matrix applied to the movie dataset.

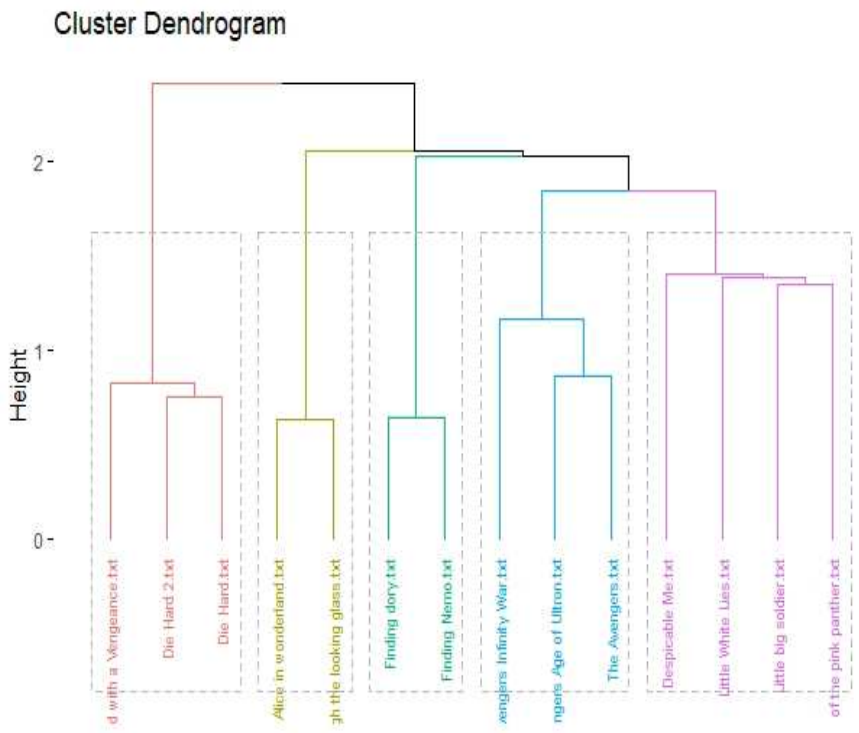


Fig. 8. Cluster dendogram using hierarchical agglomerative clustering

At each step the nodes are merged and the matrix is updated until process is complete. The output of this method is displayed in the form of a hierarchical structure called dendogram. Based on choice of optimal number of clusters, the dendogram can be cut at the desired level. Fig. 8 shows the application of hierarchical agglomerative clustering on the movie dataset resulting in the cluster dendogram.

4.3 Application of Principal Components Analysis

PCA is used to reduce the large number of variables, which in our case are terms, from the sparse document term matrix formed. The objective is to keep only the most significant terms that indicate the maximum spread in the data and use it for clustering implementation and analysis.

In order to obtain meaningful results from PCA, it is important to normalize the data first since it depends on the count of terms in a document but the length of documents is varying in the corpus. The number of principal components is chosen as (number of documents – 1) or (number of terms - 1) whichever is lesser. In our case, the number of terms is too large, so we choose the number of principal components as 13.

Fig. 9 shows the cumulative proportion of variance explained plotted against the Principal Components which increases sharply first and then gradually indicating that the maximum variance is depicted by the first few principal components. Fig. 10 indicates the proportion of variance explained against the principal components. It shows that after 5th component the proportion of variance explained is more or less steady. Hence we fixed the number of Principal Components as 5 for further analysis.

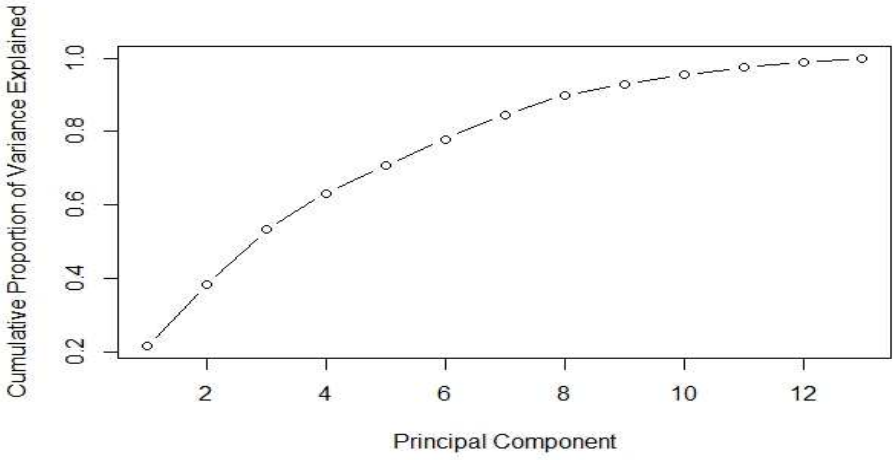


Fig. 9. Cumulative proportion of variance explained against Principal Components

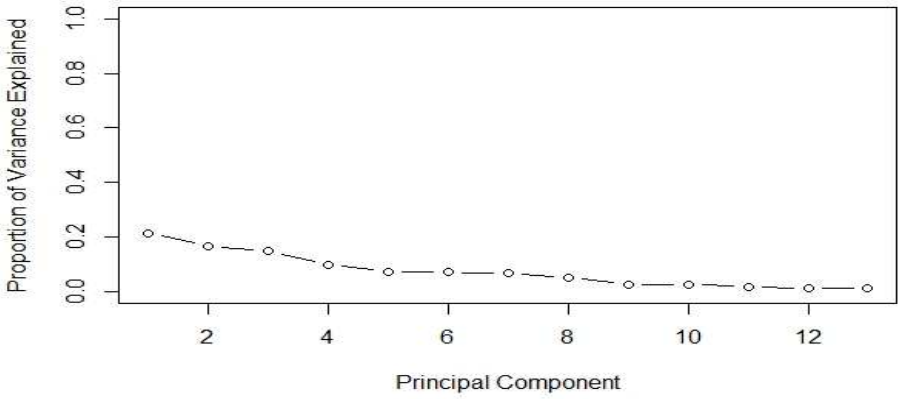


Fig. 10. Proportion of variance explained against Principal Components

Next we find which terms are the most significant in each component. Fig. 11 shows the contribution of each term in each Principal Component.

```
> tidied_pca
# A tibble: 10,680 x 3
   Tag          PC    Contribution
   <chr>        <chr>        <dbl>
 1 absolem      PC1       -0.00102
 2 accidentally PC1       -0.0243
 3 advice       PC1       -0.00114
 4 advises      PC1       -0.00114
 5 alice        PC1       -0.000656
 6 alices       PC1       -0.000656
 7 allowing     PC1        0.0251
 8 ambushed     PC1       -0.0170
 9 among        PC1       -0.0381
10 appointed    PC1       -0.00114
# ... with 10,670 more rows
```

Fig. 11. Contribution of terms in each Principal Component

Fig. 12 shows the graphical representation of contribution of terms in each of the 5 Principal Components.

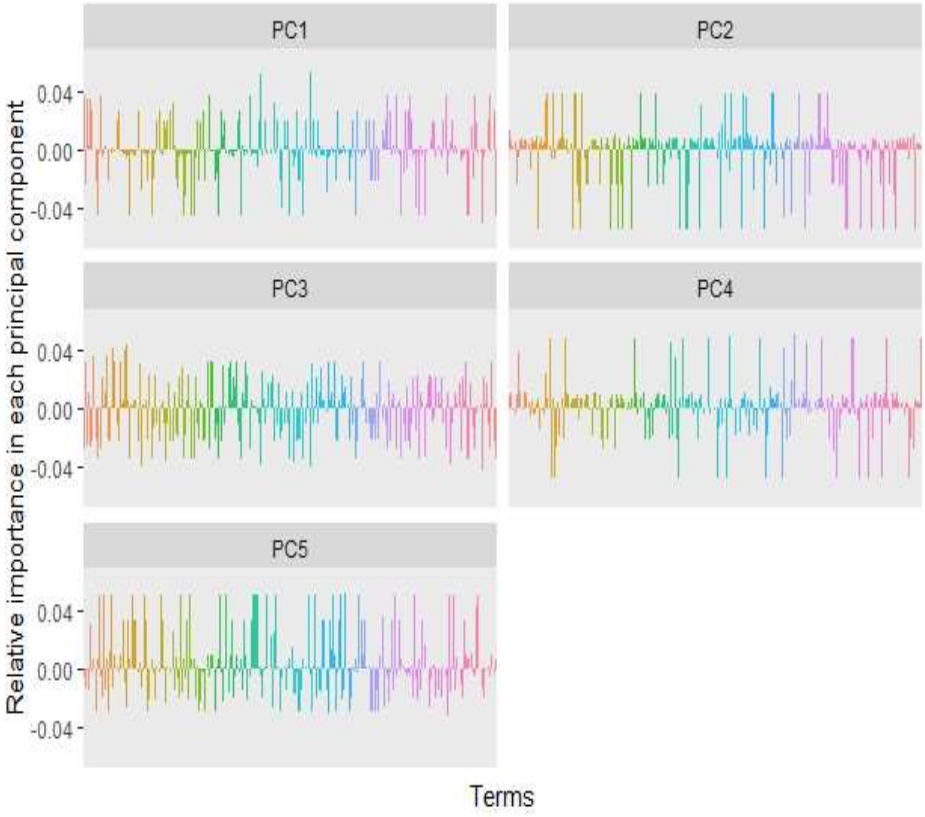


Fig. 12. Contribution of terms in all of the 5 Principal Components

We can zoom each component contribution to study in detail the highest contribution or highest absolute loadings of terms in each of the principal component. Fig. 13 depicts the contribution of terms in PC1 indicating highest positive as well as negative loadings on PC1. Fig. 14 shows the percentage variation as of the complete document corpus indicated by each principal component. As can be seen from Fig. 14, PC1 indicates the highest value at 21.58%, PC2 at 20.55% and so on.

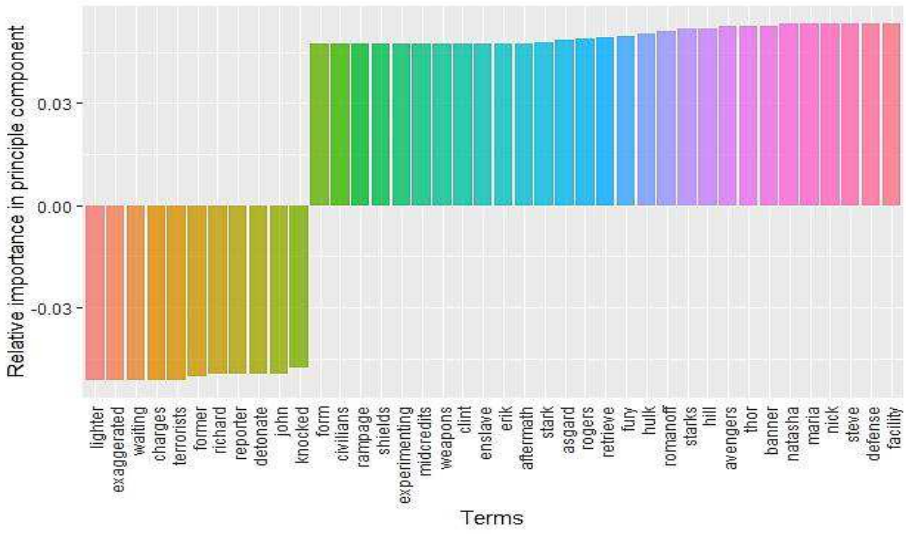


Fig. 13. Contribution of terms in PC1

```
> percent_variation
[1] 0.2168365 0.2055533 0.1982060 0.1913610 0.1880433
```

Fig. 14. Percentage Variation of the complete document corpus explained by each PC

4.4 Application of Hierarchical Agglomerative Clustering on first 2 Principal Components

PC1 and PC2 indicate the maximum variance in the entire document corpus as indicated in Fig. 14. Hence next we applied hierarchical agglomerative clustering on the first 2 principal components. Fig. 15 shows the dendogram obtained on application of HAC on PC1 and PC2.

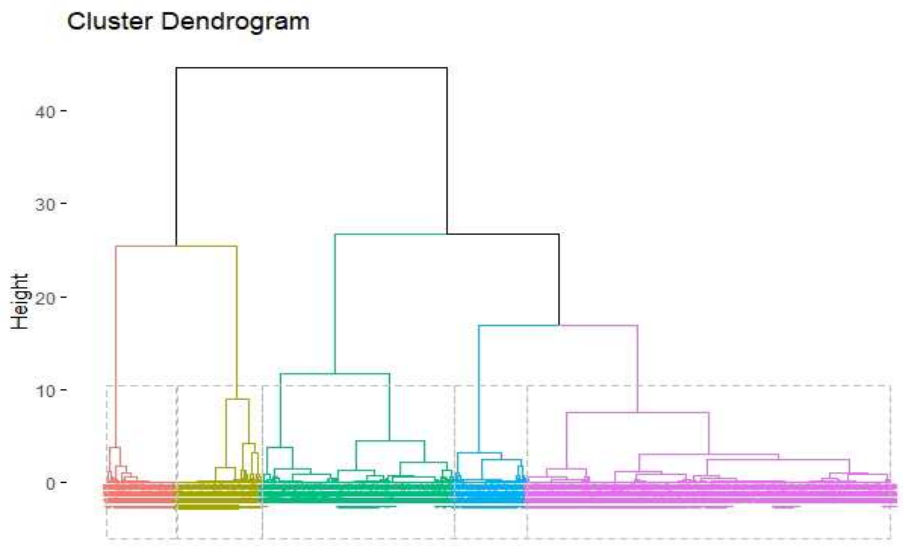


Fig. 15. Dendogram from application of HAC on principal components PC1 and PC2

5. EVALUATION

Silhouette coefficient measures how well an observation is clustered and it estimates the average distance between clusters (i.e. the average silhouette width). Observations with negative silhouette are probably placed in the wrong cluster. Fig. 16 shows the average silhouette width of 0.34 when HAC is applied to the complete document corpus.

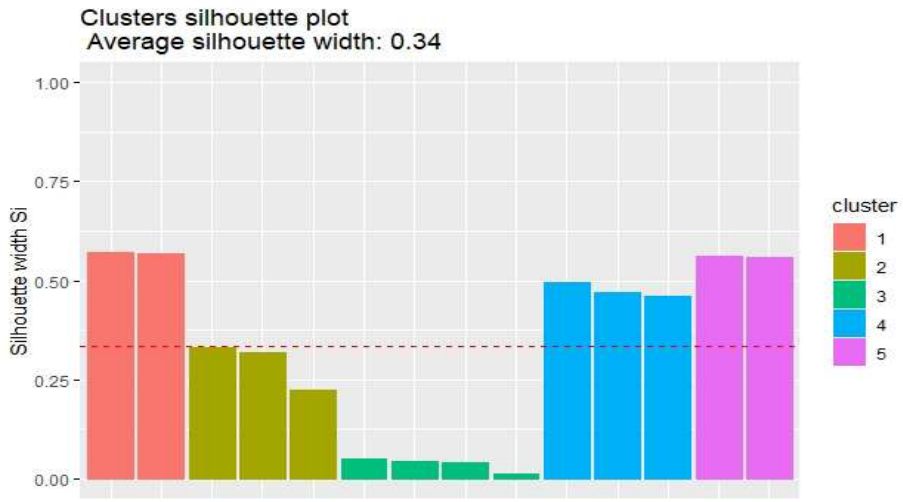


Fig. 16. Average silhouette width on application of HAC

Fig. 17 shows the average silhouette width of 0.68 when HAC is applied to the first 2 Principal Components obtained upon applying PCA to the document corpus first. As we can see the cluster quality is improved as an average silhouette width of 0.68 is obtained which is higher than that obtained on application of HAC without performing Principal Components Analysis.

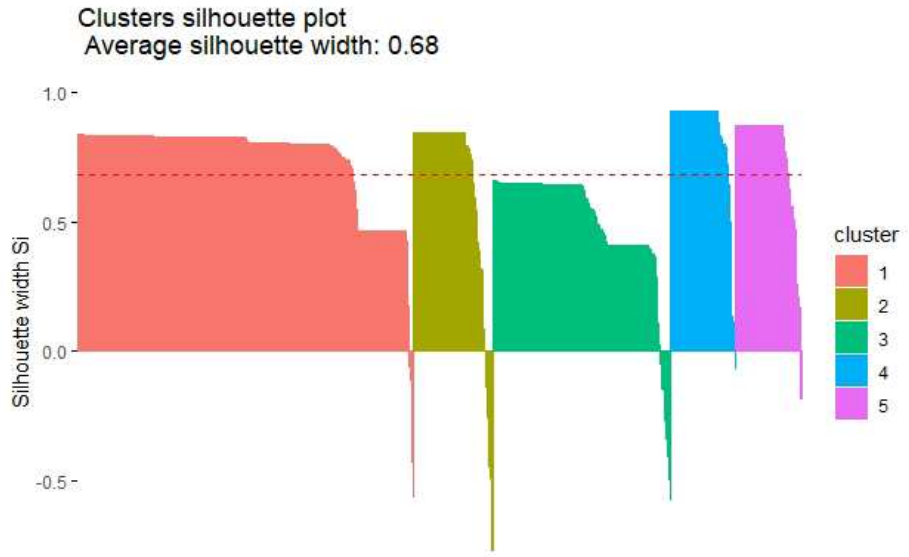


Fig. 17. Average silhouette width on application of HAC on Principal Components

6. CONCLUSION

Clustering approach for building a recommender system for movies to users based on document similarity has been discussed in this chapter. The data used was downloaded by crawling Wikipedia for movie plot summaries by different genres. Text mining methods have been applied to clean and pre-process the movie texts. Application of hierarchical agglomerative clustering and combination of hierarchical agglomerative clustering with Principal Components Analysis was studied on these move texts. When applying HAC, the optimal number of clusters has been determined using various available methods. In future, the same study can be carried out on larger dataset. Combination of other text mining and clustering approaches such as partitional approach, different similarity measures can be implemented to get still better quality clusters on larger dataset also. This can be then applied real time for building movie recommender system based on all online plot summaries available.

REFERENCES

1. G. Chaudhary, M. Kshirsagar, "Overview and application of text data pre-processing techniques for text mining on health news tweets,"*Helix*, vol. 8, issue 5,pp. 3764-3768, 2018
2. S. Vijayarani et al, "Preprocessing techniques for text mining – an overview," *International Journal of Computer Science & Communication Networks*, vol. 5, issue 1, pp. 7-16, 2015
3. Y. Wang, W. Xu, H. Jiang, "Using text mining and clustering to group research proposals for research project selection," *Proceedings of the 201548th Hawaii International Conference on System Sciences*, 2015, pp. 1256-1263.
4. X. Jiang, Y. Shi, S. Li, "Research of correction method in the feature space on text clustering," *Proceedings of the 2012 International Conference on Computer Science and Service System*, 2012, pp. 2030-2033.
5. K. Zhou, S. Yang, "Effect of cluster size distribution on clustering: a comparative study ofk-means and fuzzy c-means clustering," *Pattern Analysis and Applications*, Springer-Verlag London Ltd., 2019,pp. 1-12.
6. K. Rajput, N. Kandoi, "An ontology-based text-mining method to develop intelligent information system using cluster based approach," *Proceedings of the IEEE International Conference on Inventive Systems and Control*, Coimbatore, India, 2017, pp. 1-6.
7. G. Kou, Y. Peng, "A new hierarchical document clustering method," Proceedings of the 2009 Fifth IEEE International Joint Conference on INC, IMS and IDC, 2009, pp. 1789-1792.
8. X. Pan, J. Cheng, Y. Zia, X. Zhang, H. Wang, "Which feature is better? TF*IDF feature or topic feature in text clustering," *Proceedings of the 2012 Fourth IEEE International Conference on Multimedia Information Networking and Security*, 2012, pp. 425-428.
9. T. Pitchayaviwat, "A study on clustering customer suggestion on online social media about insurance services by using text mining techniques," *Proceedings of the 2016 IEEE Management and Innovation Technology International Conference (MITicon)*, Bang-San, Thailand, pp. 148-151.
10. Y. Zhu, B.C.M. Fung, D. Mu, Y. Li, "An efficient hybrid hierarchical document clustering method," Fifth International Conference on FuzzySystems and Knowledge Discovery, Shandong, China, 2008, pp. 395-399.
11. M. Carullo, E. Binaghi, I. Gallo, N. Lamberti, "Clustering of short commercial documents for the Web", *Proceedings of the 200819th International Conference on Pattern Recognition*, 2008, pp. 1-4.

12. M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," *Proceedings of the Text Mining Workshop*, KDD, 2000, pp. 1-20.
13. O. Gascuel, A. McKenzie "Performance analysis of hierarchical clustering algorithm", *Journal of Classification*, vol. 21, issue 1, pp. 3-18, 2004.
14. T. Jo, "String vector based AHC for text clustering," *Proceedings of the 2017 19th IEEE International Conference on Advanced Communication Technology (ICACT)*, Bongpyeong, South, 2017, pp. 673–678.
15. M.M.-T. Chiang, B. Mirkin, "Intelligent choice of the number of clusters in k-means clustering: An experimental study with different cluster spread," *Journal of Classification*, vol. 27, issue 1, pp. 3-40, 2009.
16. P. Hansen, E. Ngai, B. K. Cheung, N. Mladenovic, "Analysis of global k-means, an incremental heuristic for minimum sum-of-squares clustering," *Journal of Classification*, vol. 22, issue 2, pp. 287-310, 2005.
17. S. Yuan, G.Wenbin, "A text clustering algorithm based on simplified cluster hypothesis," *Proceedings of the 2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*, Toronto, ON, Canada, 23-24 Dec. 2013, pp. 412-415.
18. Y.Li, C.Luo, S.M. Chung, "Text clustering with feature selection by using statistical data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, issue 5, pp. 641-652, May 2008.
19. D. Marutho, S.H.Handaka,E. Wijaya, Muljono, "The determination of cluster number at k-mean using elbow method and purity evaluation on headline news," *Proceedings of the 2018 IEEE International Seminar on Application for Technology of Information and Communication*, Semarang, Indonesia, 29 November 2018, pp. 533-538.
20. Xiaolin Xiao and Yicong Zhou, "Two-Dimensional Quaternion PCA and Sparse PCA", *IEEE Transactions on Neural Networks And Learning Systems*, Vol. 30, No. 7, pp. 2028-2041 July 2019
21. Xiaoxu Han, "Nonnegative Principal Component Analysis for Cancer Molecular Pattern Discovery," *IEEE/ACM Transactions on Computational Biology And Bioinformatics*, Vol. 7, No. 3, pp. 537-549, JULY-September 2010.
22. Jun Yan, Ning Liu, Shuicheng Yan, Qiang Yang, Weiguo (Patrick) Fan, Wei Wei, and Zheng Chen,"Trace-Oriented Feature Analysis for Large-Scale Text Data Dimension Reduction", *IEEE Transactions on Knowledge And Data Engineering*, Vol. 23, No. 7, pp. 1103-1116, July 2011

Cite this article

Gauri Chaudhary and Manali Kshirsagar, Enhanced text clustering approach using Hierarchical Agglomerative Clustering with Principal Components Analysis to design Document Recommendation System, In: Sandip A. Kale editor, Advanced Research in Computer Engineering, Pune: Grinrey Publications, 2021, pp. 1-18