American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

© Global Society of Scientific Research and Researchers

http://asrjetsjournal.org/

Visualization and Clustering of Text Retrieval

Abdel Naser Pouamoun*

International Computer Institute, Ege University, 35100, Izmir, Turkey

Email: anpouamoun@gmail.com

Abstract

In a fast transforming world where all objects will be generating data, dealing with large data collections has been a major concern for data scientists. Major challenges faced by those scientists are among others the difficulty to represent these data in a better way and therefore to communicate hidden information from these data to the users. Accordingly, many data analysis and data visualization techniques have been proposed. Moreover, depending on the nature of data to visualize and the type of information to communicate, a certain number of data processing techniques should be considered. In this work, we analyze and visualize a sample data of TREC-6 from the TREC (Text Retrieval Conference) collections. TREC document collections comprise full text from newspapers articles and US government records. They are primarily dedicated to researchers in Information Retrieval (IR) systems and Natural Language Processing for the development of their works. First, documents are parsed and words extracted to build a corpus in a form of a $n \times p$ matrix. Then, Principal Component Analysis is applied to the corpus matrix to reduce the dimension from p to 2. Eventually, the unsupervised K-means algorithm is used to discriminate data into clusters that are interactively visualized thanks to the popular visualization tools such as Pie Chart, Stacked Bar Chart and Scatter Chart. The diversity of the nature of information contained in TREC-6 can be observed thanks to the most frequent words of each cluster that appear on the Bar Chart upon clicking on the Pie Chart of the corresponding cluster.

Keywords: data visualization; TREC collections; Principal Component Analysis; K-means algorithm; clustering.

1. Introduction

Data Visualization aims at communicating information from data to users by means of graphical visual objects such as bars, points, lines etc. [1].

----

* Corresponding author.

Throughout this work, the dataset used is mainly from the TREC (Text Retrieval Conference) Collections. TREC Document Collections comprise full text from newspapers articles and US government records. They are primarily dedicated to researchers in Information Retrieval (IR) systems and Natural Language Processing for the development of their works [2]. In this work, we analyze, cluster and visualize documents from TREC Collections in order to figure out what they are mainly about and how diversified are the main topics addressed in the original articles. Thus, we first prepare our data by building a $n \times p$ matrix (where n is the number of documents in the dataset and p the number of words in the overall corpus); and, we secondly perform Principal Component Analysis (PCA) in order to reduce the dimensions of our dataset from p ($p > 2000$) to 2. We then use the PCA-processed dataset for Document Clustering using K-means algorithm. At the end, clusters are visualized with Pie Chart, Stacked Bar Chart and projected to a Cartesian space using Scatter Chart. Moreover, for a better understanding of what will follow in this paper we would like to highlight one of the important expectations behind this work. This work is mainly about Data Visualization of a dataset of documents clustered into a given K groups (clusters) following similarities between documents. The clustered dataset would be then used to fulfil any other scientific work's needs such as Distributed Information Retrieval, for instance.

1.1. System overview

Upon launching the program, we have a window with three empty titled panes corresponding to the three charts that will show our data upon processing (Figure 1a). In addition, at the far left, we have a text box to contain the initial number k of clusters, a Process Data button to launch the data processing and a Clear Data button to clear possible previously processed data. As Figure 1b suggests, at the end of the data processing, the window shows three charts (from right to left):

- A Pie Chart representing clusters proportion in the whole dataset and their legend.
- A Stacked Bar Chart showing the 10 most frequent words in a cluster upon clicking on its corresponding slice in the Pie Chart.
- A Scatter Chart where each scattered point represents a document within the dataset and takes the color of the cluster to which the document belongs.

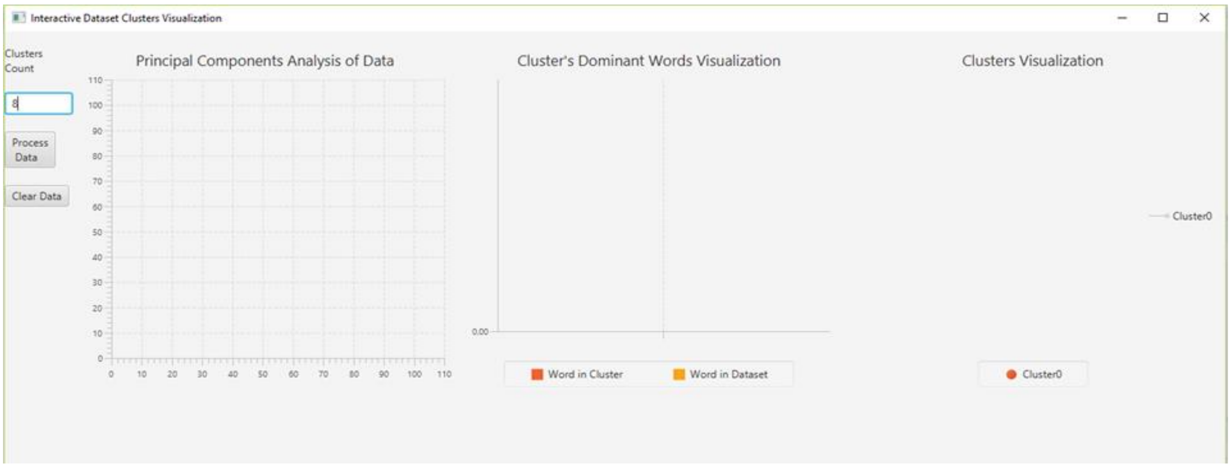


Figure 1a: Starting window

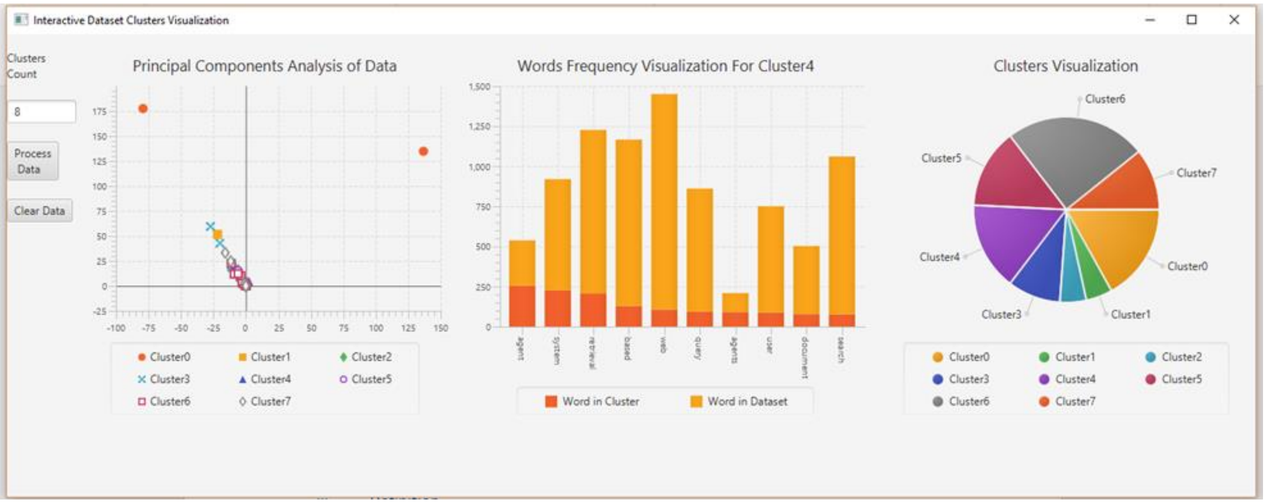


Figure 1b: System overview

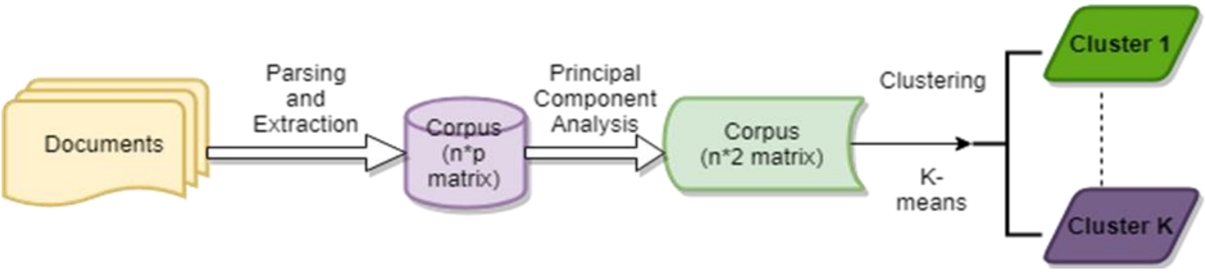


Figure 2: Data processing steps

2. Principal component analysis

2.1. Definition

Principal Component Analysis (PCA) is a mathematical process that uses a series of linear algebra operations to transform a set of features of possibly interrelated variables into a set of new features of independent variables called principal components. Principal components are orthogonal and statistically uncorrelated to one another. In other words, PCA is a procedure of detecting patterns in large datasets by highlighting their similarities and differences. However, it is very difficult to extract identified patterns from large data and even impossible to visualize data with a high number of dimensions. PCA is then used to analyze and compress data by reducing the number of dimensions with sometimes a few loss of information [3]. Karl Pearson first mentioned PCA in 1901. In addition, later in the 1930s, PCA was developed and named by [4]. Depending on the field of application, PCA takes different denominations. In our case, we will be applying some linear algebra operations to our dataset by significantly reducing its dimension for an easier clustering and visualization into a $2D$ space. This process is called eigenvalue decomposition.

2.2. Method

2.2.1. Preparation of the dataset

As mentioned in the introduction, our dataset here is a set of documents from TREC (Text Retrieval Conference) Collections namely TREC 4 & 5. Since this dataset has more than thousands of documents and could take hours to be loaded and parsed, we have made a smaller sample of 67 files that we have used throughout our work. Concretely, the data preparation consists of extracting all the words from our files by removing stop words (if, then, but, before, etc.) such as to obtain an $n \times p$ matrix where n the number of rows represents the number of documents in the dataset and p the number of features corresponding to the words in the corpus (bag of words). For a given document (row i) and a given word (column j), $Mij = x$ if the word j appears exactly x times in the document i. In order words, x represents the occurrence of the word j in the document i.

2.2.2. Calculation and subtraction of the mean

In the PCA process, the mean of each data dimension (vector) is calculated and subtracted across each dimension. In addition, it will be used later for the calculation of the covariance. In our case, we calculate mean of each feature vector and subtract it across the column. If X is a vector with n elements, the mean of X is given by the mathematical formula:

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \qquad (1)$$

Notice the symbol $\overline{\mathrm{X}}$ to indicate the mean of the set X.

2.2.3. Calculation of the Covariance matrix

In order to have a covariance matrix, we should calculate the covariance of the vectors of our matrix. Covariance is always measured between two dimensions. It allows finding out how much the dimensions vary from the mean with respect to each other. The covariance between the dimensions X and Y is given by the following formula:

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \qquad (2)$$

If we calculate the covariance between one dimension and itself, we get the variance. Therefore, if we have a 3-dimensional dataset (X, Y, Z), then we could measure the covariance between the X and Y dimensions, the X and Z dimensions, and the Y and Z dimensions. In fact, for a p-dimensional dataset, we can calculate $\frac{p!}{(p-2)!*2}$ different covariance values. In order to get all the possible covariance values between all the different dimensions, we can calculate them all and put them in a matrix. This matrix is then called the covariance matrix. Thus, the definition of the covariance matrix for a set of data with p dimensions is:

$$C^{p \times p} = (C_{i,j}\,,\; C_{i,j} = cov(Dim_i, Dim_j)) \quad (3)$$

where $C^{p\times p}$ is a matrix with p rows and p columns, and Dim_x is the x^{th} dimension. All that this formula says is that if we have a p-dimensional dataset, then the matrix has p rows and columns (so is square) and each entry in the matrix is the result of calculating the covariance between two separate dimensions. E.g. the entry on row 3, column 2, is the covariance value calculated between the third dimension and the second dimension. It is important to note that along the main diagonal, we can notice that the covariance value is between one of the dimensions and itself. These are the variances for that dimension. The other point is that since $cov(a,b) = cov(b,a)$, the matrix is symmetrical about the main diagonal.

2.2.4. Eigenvalue Decomposition

This step is considered as the marrow of the Principal Component Analysis. It allows finding the eigenvectors and eigenvalues of the covariance matrix. It consists of computing the matrix E of eigenvectors that diagonalizes the covariance matrix C :

$$E^{-1}C\,E = D, \qquad (4)$$

where D is the diagonal matrix of eigenvalues of the covariance matrix C. This step involves a number of quite complex arithmetic operations including equations resolutions especially when we have more than three dimensions to deal with ($p > 2000\ in\ our\ case$). For that reason, it is recommended to use readily available algorithm provided by libraries depending on the language or system used. For our case, as we are coding in Java programming language, we have used JAMA. JAMA is a basic linear algebra package for Java. It provides user-level classes for constructing and manipulating real, dense matrices [5].

2.2.5. Determining the new dataset

Upon obtaining the eigenvectors and the eigenvalues matrices, we sort the eigenvectors about the eigenvalues in descending order as to obtain the eigenvectors with the biggest eigenvalues at the beginning of the matrix E while paying attention to keep the correct pairing between columns of matrices. Very often and fortunately, some libraries when called return sorted matrices in descending or ascending order (the case of JAMA). We therefore only have to select a subset of p' eigenvectors as basis vectors. In our case $p' = 2$ since the purpose is to project at the end our dataset into a Cartesian space. We will therefore have a $p \times 2\ E'$ matrix of 2 columns. Finally, to obtain our new dataset we perform the following operation:

$$M'' = M' \times E' \quad (5)$$

where M'' (our new dataset matrix) is an $n \times 2$ matrix (remember we have n documents in our dataset) and M' the mean-subtracted (we subtracted the mean from each column at the beginning of PCA) $n \times p$ matrix.

3. Document clustering

3.1. Definition

Clustering is a process consisting of subdividing a larger set into smaller subsets called clusters. Elements of each subset are similar and have some attributes in common. The similarity or proximity is calculated using the distance measurement methods called distance metrics [6]. When clustered elements are textual documents, we talk about document clustering or text clustering. Document clustering implicates extraction and use of features. Features are bags of words and characterize the contents of the clusters [7].Very often, especially in Machine Learning and Natural Language Processing, Document Clustering is strictly assimilated to text clustering that is referred as Topic Modeling and the result is clusters of similar words. Unlike the Topic Modeling, our work here involved the physical splitting of dataset into a given number of clusters containing similar documents. In short, we are performing Document partitioning.

3.2. Clustering using K-means algorithm

3.2.1. Definition

K-means clustering is one of the most famous algorithms for cluster study in data mining. It consists of isolating n elements into k clusters in which each element belongs to the cluster with the closest mean, serving as a model of the cluster. At every iteration, the mean of each cluster is recomputed and elements are reassigned accordingly. MacQueen designed the K-Means algorithm in 1967. This algorithm has two key parameters: (1) a dataset, (2) a positive integer k representing the proposed number of clusters to build from the dataset [8].

***3.2.2.** Principle*

K-means algorithm functions as follows:

Input: The number of k and a dataset containing n objects.

Output: A set of k-clusters that minimize the squared-error criterion.

Method:

(1) Arbitrarily choose k objects as the initial cluster centers;

(2) Repeat;

(3) (Re) assign each object to the cluster to which the object is the most similar based on the mean value (centroid) of the objects in the cluster;

(4) Update the cluster mean, i.e. calculate the mean value of the object for each cluster;

(5) Until no change.

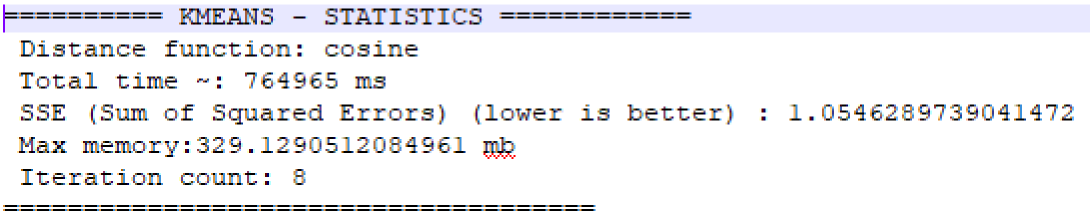
Moreover, it is important to highlight that the dataset documents should be parsed to obtain a bag of words in the form of matrix as explained in the section above dedicated to the PCA (preparation of the dataset). Fortunately, for our K-means clustering, we used the reduced matrix M'' obtained from PCA application since it well represents the initial dataset. In addition, that allows having less arithmetic operations. Another point to note is that K-means is a random operation and gives different results for the same inputs. Therefore, it is important to run it with different distance metrics and retain the result that well meets our expectations.

At the end of the K-means clustering, we copy each document from the dataset to its corresponding cluster folder in order to get our different data partitions.

3.2.3. Distance metrics

In the K-means Clustering algorithm, different distance metrics can be used depending on the dataset. However, the most common distance metrics are Manhattan distance, Euclidian distance, Cosine distance, Jaccard distance, correlation, etc.

Cosine distance has given the lowest Sum of Squared Errors (SSE) and a better observable clustering representation for our dataset (Figure 3). However, since distance metrics benchmarking is not the purpose of this work, we will not linger on.



```
========== KMEANS - STATISTICS ============
 Distance function: cosine
 Total time ~: 764965 ms
 SSE (Sum of Squared Errors) (lower is better) : 1.0546289739041472
 Max memory:329.1290512084961 mb
 Iteration count: 8
=====================================
```

Figure 3: K-means performance statistics for Cosine distance

4. Visualization

4.1. Pie Chart

Pie Chart is one of the most popular data visualization tools. Very often, it is used to visualize the distributions of a set of objects according to certain segregation criteria. In our case, it helps us to figure out how big each of our cluster is. The number of slices corresponds to the number of clusters that contain at least one document (empty clusters are not considered). The area of each slice is proportional to the number of files (documents) contained in the corresponding cluster (Figure 4).

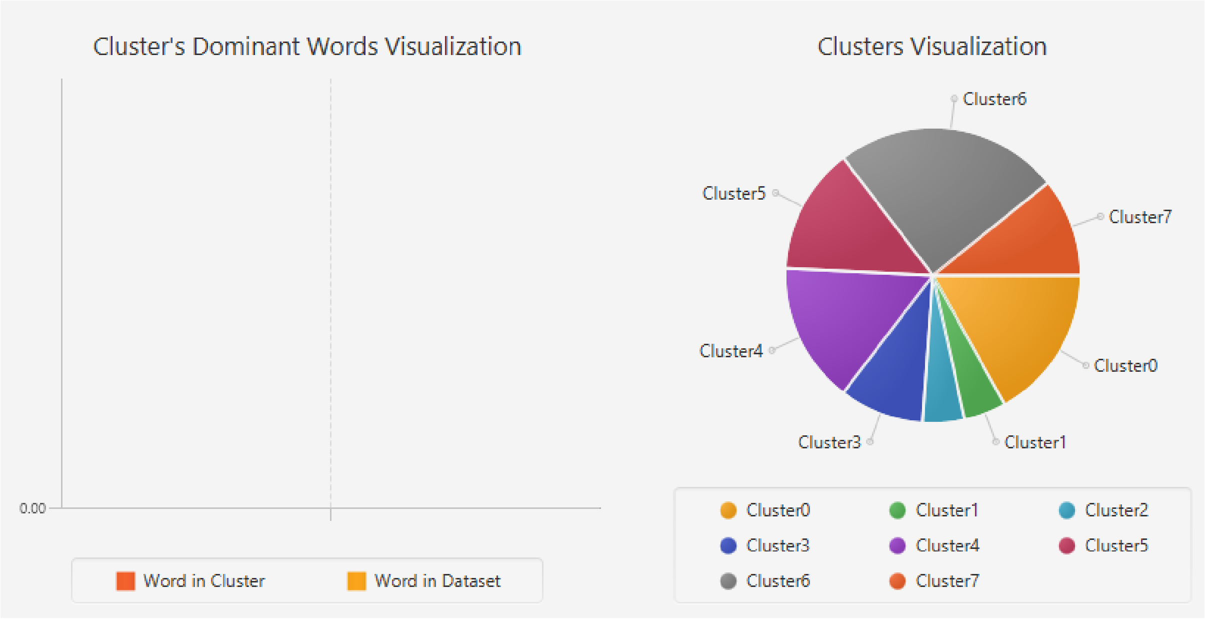


Figure 4: Pie Chart visualization of clusters distributions

4.2. Stacked Bar Chart

Stacked Bar Chart is a variant of Bar Chart that allows visualizing different proportions of an object within more than one set. In our case, it shows how frequent a word is in the cluster to which it belongs (red bars) and in the whole dataset (yellow bars).This is done interactively upon clicking on a slice of the Pie Chart (right) corresponding to a given cluster. For each cluster, we have chosen to visualize the ten most frequent words. This helps having an overview of different topics or fields of documents contained in a given cluster (Figure 5).

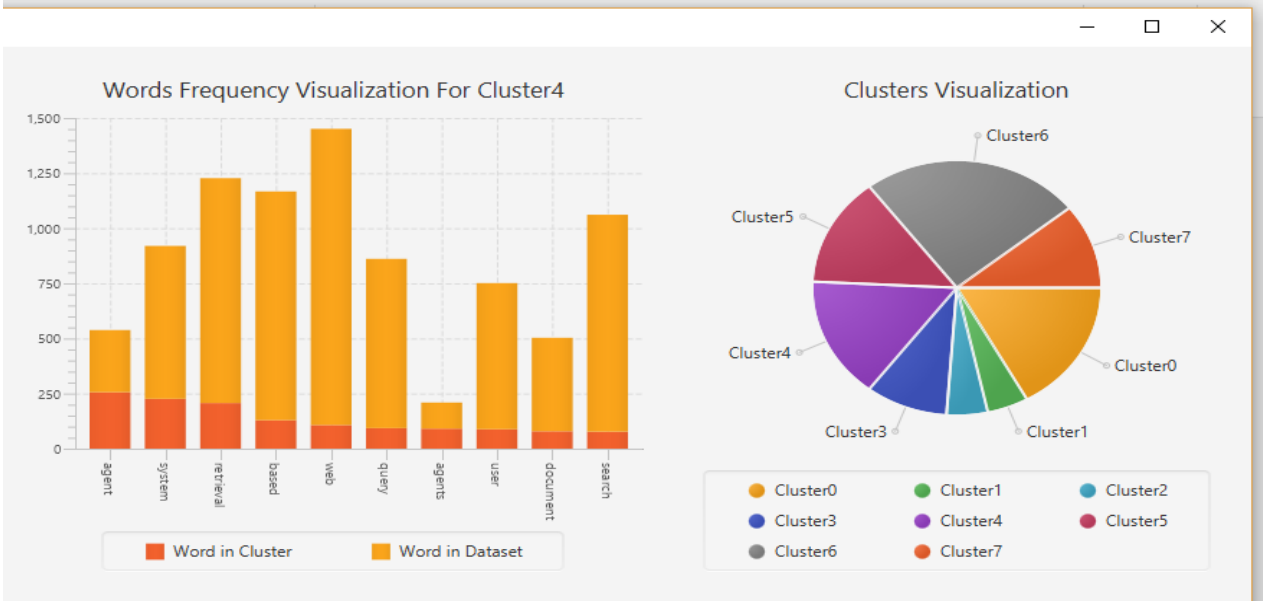


Figure 5: Stacked Bar Chart (left) showing the 10 most frequent words of Cluster4 upon clicking on its corresponding Pie Chart slice (right).

4.3. Scatter Chart

Scatter Chart or Scatter Plot is a kind of diagram often used in mathematics to project data into a Cartesian space [9]. Every set of data comprising two variables is displayed to the two-dimensional space in the form of point. The value of the first variable defines the location of the point on the horizontal direction (axis) and the value of the second variable defines the location of the point on the vertical axis. At this level, the great job done by the Principal Component Analysis (PCA) is clearly observable. Thanks to the PCA, we extracted and built up from p features vectors 2 new features that well represent our dataset. In this chart, every point represents a document that takes the color of the cluster to which it belongs. The axis of our chart are directed by our two orthogonal eigenvectors obtained from the PCA. When we look at the scatter chart, at first glance, one can wonder that documents are not very spread across the two axis of our chart (Figure 6). This is the result of the reduction from a p-dimensional space ($p > 2000$) that cannot be visualized to a 2-dimensional space. Therefore, the confinement of scattered points reveals the high similarities between documents of our dataset collection although different clusters are observable. Moreover, when we zoom in on our Scatter Chart we observe a clear aggregation of documents in clusters though some noises (Figure 7).

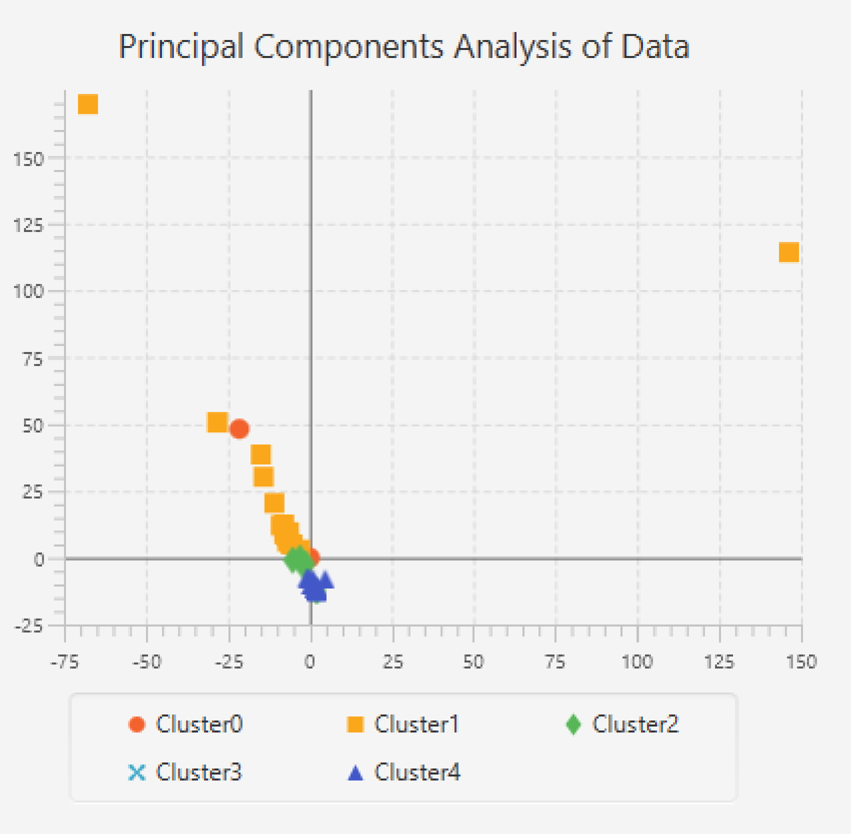


Figure 6: Scatter Chart clusters visualization without zoom in

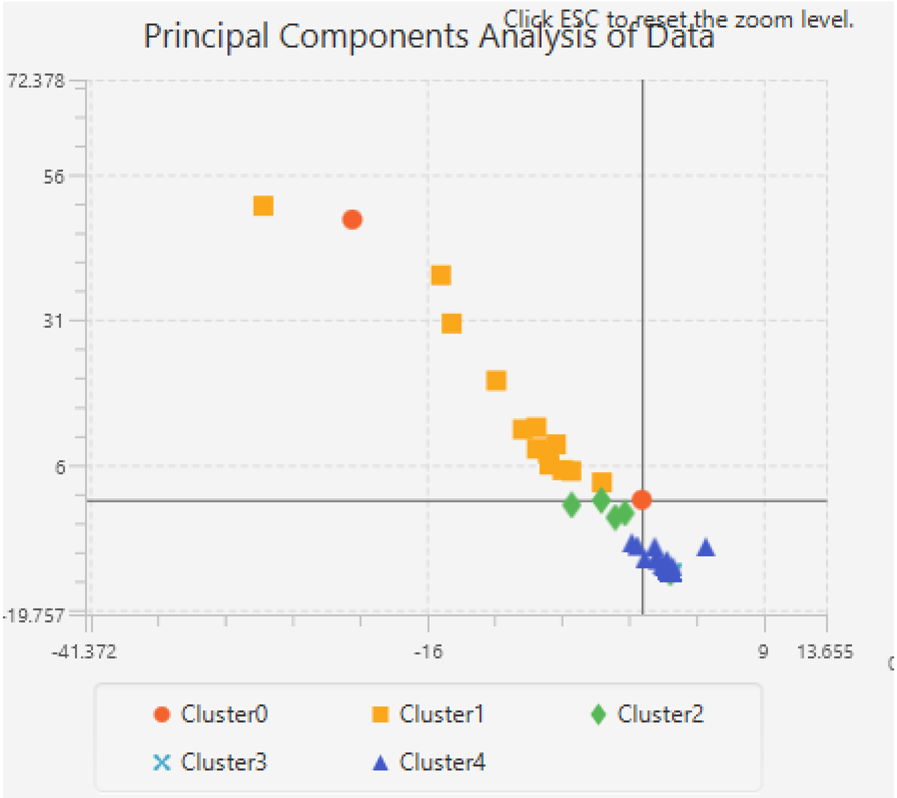


Figure 7: Scatter Chart clusters visualization with zoom in

5. Conclusion

Overall, In this work, we used Principal Component Analysis to extract two features that represent the best our initial dataset consisted of p ($p > 2000$) features less easier to cluster and even impossible to visualize. Then, Document Clustering was performed using K-means clustering algorithm. The results of clustering were visualized using Pie Chart, Stacked Bar Chart and Scatter Chart. The dataset used throughout this work mainly consists of documents from the TREC (Text Retrieval Conference) Collections. TREC Document Databases are distributed for the development and testing of Information Retrieval (IR) systems and related Natural Language Processing research [2]. It is noticeable that the ceiling number of clusters of our dataset is 7. Even if we set a bigger value as default number of clusters at the launching of the program (see Figure 1b), we will end up with exactly 7 clusters. This is evidence that K-means is an unsupervised algorithm that learns from the data.

References

[1]. C. Kelleher and T. Wagener, "Ten guidelines for effective data visualization in scientific publications," Environmental Modelling & Software, vol. 26, no. 6, pp. 822-827, June 2011.

[2]. N. I. o. S. a. Technology. [Online]. Available: http://trec.nist.gov.

[3]. L. I. Smith, "A tutorial on Principal Components Analysis," 2002.

[4]. A. N. Gorban and Z. Y. Andrei, "PCA and K-Means Decipher Genome," in Principal Manifolds for Data Visualization and Dimension Reduction, vol. 58, Heidelberg, Springer, 2008, pp. 309-323.

[5]. M. a. NIST. [Online]. Available: https://math.nist.gov/javanumerics/jama/.

[6]. Y. Gyu Jung, M. Soo Kang and J. Heo, "Clustering performance comparison using K-means and expectation maximization algorithms," Taylor & Francis, 19 June 2014.

[7]. M. . Steinbach, G. Karypis and . V. Kumar, "A Comparison of Document Clustering Techniques," in KDD workshop on text mining, Minneapolis, Minnesota, 2000.

[8]. P. Fournier-Viger. [Online]. Available: http://data-mining.philippe-fournier-viger.com/.

[9]. D. . J. Sloane, "Visualizing Qualitative Information," The Qualitative Report, vol. 14, no. 3, pp. 488-497, 2009.