



# Similarity Measure Based Cluster Generation of Text Documents

Garima Kansal<sup>1\*</sup>, Mukesh Rawat<sup>2</sup>

## Abstract

Search engine is just as important as having access to the World Wide Web. Since the dawn of the internet, the development of faster, more productive, and increasingly precise search engines has been a hot topic among online researcher. The cluster of Document will be required by search engine to identify similar documents specific to a user query. An Efficient algorithm will be suggested that helps to identify the proper cluster of the document based on the internal behaviour matching between the specific document and the document residing in a specific cluster. In this paper a comparative study is done among the various similarity measures of the documents such as cosine similarity, Jaccard similarity, Pearson correlation, Dice coefficient, Euclidean distance these similarity measures are used to group the documents into a cluster according to their common features and the best similarity measure algorithm is selected.

**Key Words:** Document Clustering, Similarity Measure, Cosine Similarity, Purity, Document Fetcher, Cluster Generation.

**DOI Number:** 10.14704/nq.2022.20.8.NQ44109

**NeuroQuantology 2022; 20(8):1008-1016** 1008

## Introduction

There is various part of document clustering like topic extraction, quick information retrieval or filtering and automatic document arrangement. (also known as text clustering). [1] It has a lot in common with data clustering. In response to a wide query, browser make it difficult to find relevant information to from thousands of pages return by online search engine to users.

A descriptor is a group of words that describe the contents of a cluster [15,16]. Document clustering is often considered a centralized procedure. [2] Clustering web documents for search users is an example of clustering documents. Online and offline applications of document classification can be separated into two groups. Online apps frequently have efficiency problems as compared to offline ones. Documents can be grouped into hierarchical structures that are ideal for browsing by aggregating or splitting them. However, such an algorithm is prone to inefficiency issues. The K-means algorithm and its modifications are used to create the other algorithm. It is usually more efficient than the hierarchical method, but less accurate. [9]

We can use clustering to discover hidden relationships between data points in a dataset.

1. In marketing, buyers are segregated based on their commonalities in order to conduct focused marketing.
2. Given a collection of texts, we must categorize them into a topic hierarchy based on content similarities.
3. Identifying various types of patterns in image data (Image processing). It's useful for detecting underlying patterns in biology studies. [10]

There are numerous ways to assign documents to a cluster. In this method, we start with a cluster of many documents, combine them all into one super file, and use the cosine similarity measure technique to match a new document to this super file. This paper focussed on deriving a formula to filter super file's keywords such that it gives enhanced similarity result, that is similar and non-similar clusters of documents are polarised for differentiation.

**Corresponding author:** Garima Kansal



**Address:** <sup>1,2</sup>Department of Computer Science and Engineering, Meerut Institute of Engineering and Technology, Meerut.  
E-mail: <sup>1</sup>garima.kansal.mtcs.2020@miet.ac.in; <sup>2</sup>mukesh.rawat@miet.ac.in

Filtering keywords of super File allows a faster algorithm for matching a new document than the conventional method by reducing document loading, processing and search time by reducing data size and keeping the most effective keywords only.

### Issues and Challenges for Cluster Generation

In many cases, the web documents may suffer from ranking problems of search engine. Some of the issues and challenges based on page ranking with their possible solution are mentioned below:

#### *Time taken to identify a Proper Cluster for a given Document*

The solution is as the number of documents are increasing in the created clusters a technique will be developed so that a proper cluster will be assign to a newly fetched document by comparing with its limited number of documents of various created clusters.

#### *Choosing a Proper Data Structure to Accommodate a Large Number of Clusters Created*

The solution is as the greater number of clusters created a technique will be developed for grouping similar clusters and arrange them in a hierarchal way so that identify the relevant information according to user query in less amount of time.

#### *Improving the Cluster Purity Factor*

The solution is a technique will be developed which results in a close binding and similarity between the document within a cluster.

### Document Cluster Techniques

#### *Flat Clustering*

A group of documents is divided into subsets or clusters by clustering techniques. The algorithms' purpose is to produce clusters that are internally consistent but clearly distinct from one another. In other words, documents in a cluster should be consistent. Papers in one cluster should be as similar as possible, whereas documents in other clusters should be as distinct as feasible.

#### *Hierarchical Clustering*

A flat, unstructured, non-deterministic group of clusters with a predetermined cluster as input

[15,16]. Flat clustering creates an unstructured collection of clusters, but hierarchical clustering creates a hierarchy, a structure which is more informative than the unstructured group of clusters produced by flat clustering. When using hierarchical clustering, the number of clusters does not need to be predefined, and the majority of hierarchical algorithms used in IR are deterministic.

Flat clustering is preferred when efficiency is a top goal, whereas hierarchical clustering is preferred when one of flat clustering's potential downsides is a concern. In addition, a lot of specialists think that hierarchical clustering generates superior clusters to flat clustering.

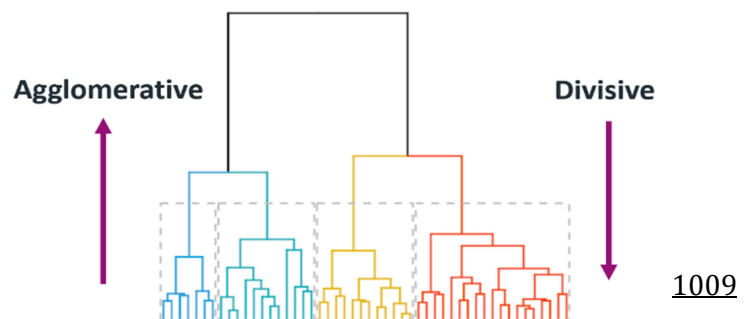


Figure 1. Hierarchical Clustering

### Text Similarity Matching Techniques

Various type of text similarity matching techniques is given below

- Cosine similarity
- Jaccard similarity
- Dice similarity
- Pearson Correlation
- Euclidean Distance

1. Cosine Similarity: A measure of the similarity between two vectors in an inner product space is called cosine similarity. In order to determine if two vectors are pointing in the same general direction, the cosine of the angle between them is measured. It is frequently applied to determine document similarity in text analysis. The frequency of a particular word or phrases in the text is captured by each of the hundreds of attributes that may be applied to a document. [12]

Cosine similarity is a similarity measure that may be used to rank texts in respect to a vector of query terms or to compare them. Consider two vectors, X and Y, to compare. As a similarity function, we have



used the cosine measure. It is calculated as cosine of 2 angles between two n-dimensional vectors in an n-dimensional space. [3] The detailed formula is given below -

$$\begin{aligned} \text{Similarity (X,Y)} \cos\theta &= \frac{X.Y}{|X| \times |Y|} \\ &= \frac{\sum_{i=1}^n XiYi}{\sqrt{\sum_{i=1}^n Xi^2} \sqrt{\sum_{i=1}^n Yi^2}} \\ &= T_1 * T_{11} + T_2 * T_{21} + T_3 * T_{31} + \dots \\ &= \sum_{i,j=1}^n Ti * Tj \\ &= f \\ &= \cos \theta \end{aligned}$$

2. Jaccard similarity: The intersection of two papers is divided by their union, which refers to the number of common terms out of a total number of words, and this is known as the Jaccard similarity.

The Jaccard similarity score ranges from 0 to 1, with 0 being the most similar and 1 being the least similar. The Jaccard similarity score will be one if the two documents being compared are identical. The Jaccard similarity is 0 if the two documents have no terms in common. [11]

We have two sample sets of words X and Y and similarity between them [4],

$$\begin{aligned} J(X,Y) &= \frac{|X \cap Y|}{|X \cup Y|} \\ &= \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|} \end{aligned}$$

3. Dice coefficient: The Srensen–Dice index, or simply the Dice coefficient, is a statistical method for comparing the similarity of two collections of data. Although this index is arguably the most widely used tool for validating Intelligence photo segmentation algorithms, it is actually a much bigger idea that can be used to sets of data for a number of applications, including natural language

processing (NLP). It is referred to as Sorensen Dice coefficient. Sorensen's authentic formulation became supposed to be implemented to discrete data.

Given sets, P and Q, it's far described as [5]

$$DSC = \frac{|P \cap Q|}{|P| + |Q|}$$

4. Pearson Correlation: The Pearson R statistical test, sometimes referred to as the Pearson's correlation coefficient, is a statistical test that evaluates the strength of correlations between the variables. It is typically to determine the strength of the association between the two variables, it is a good idea for the researcher to compute The correlation coefficient's value when conducting a statistical test between the two variables. The Pearson coefficient is a value between -1 and 1. [6]

$$\begin{aligned} &= \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}} \end{aligned}$$

5. Euclidean Distance: The magnitude of the vectors x and y determines the distance between them. An increase in the magnitude of vector x or y would result in a bigger Euclidean distance in this scenario. In terms of NLP, this indicates that, even if they are on the same topic, a larger body of text with more words in terms of diversity and frequency will have a significantly greater magnitude than a smaller body text. Euclidean distance among factors in Euclidean area is the duration of a line section among the two factors. Two-point u and v Euclidean distance defined as [7],

$$Ed = \sqrt{\sum_{i=1}^z (u_i - v_i)^2}$$

1010

**Table 1.** Similarity measure (in terms of time complexity and n is the number of dimensions of X and Y) [8]

Distance Measure	Time Complexity	Advantage	Disadvantage	Application
Euclidean Distance	O(n)	Very common, calculates smoothly and works well with compact or remote clusters.	Sensitive to outliers.	Used in K-means algorithm, Fuzzy C-means algorithm.
Pearson coefficient	O(2n)	It gives best result using hierarchical single-link algorithms.	It is Used only high dimensional datasets only.	Partitioning and hierarchical clustering algorithm.
Dice coefficient	O(n)	Dice rating isn't always simplest a degree of what number of positives you find, however it additionally penalizes for the fake positives that the technique finds, much like precision. so it's miles greater much like precision than accuracy.	Dice rating is likewise penalizing for the positives that your algorithm could not find.	Dice coefficient is useful for ecological community data



Jaccard similarity	$O(n^2)$	It is good for cases where duplication does not matter.	It could dominate average score taken over multiple sets	Used to compute the similarity between two objects, such as two text documents.
Cosine Similarity	$O(3n)$	It is independent of the length of the vector and invariant with respect to the rotation.	-	Mostly used in document similarity application.

### Proposed Cluster Identification Methodology

Using various techniques to discover the "clusters," cluster analysis combines comparable things together. These clusters are latent variables, which means they are inferred from the relationships between the items rather than being explicitly measured. When determining how closely items, content, or services connect from the viewpoint of

users, cluster analysis is the method employed. Knowing how your users differ and are similar in terms of their demographics, psychographics, and behaviour is a crucial component of knowing them. To describe the traits that each group shares as a whole, these groups are sometimes referred to as segments or clusters.

### Workflow of System Design

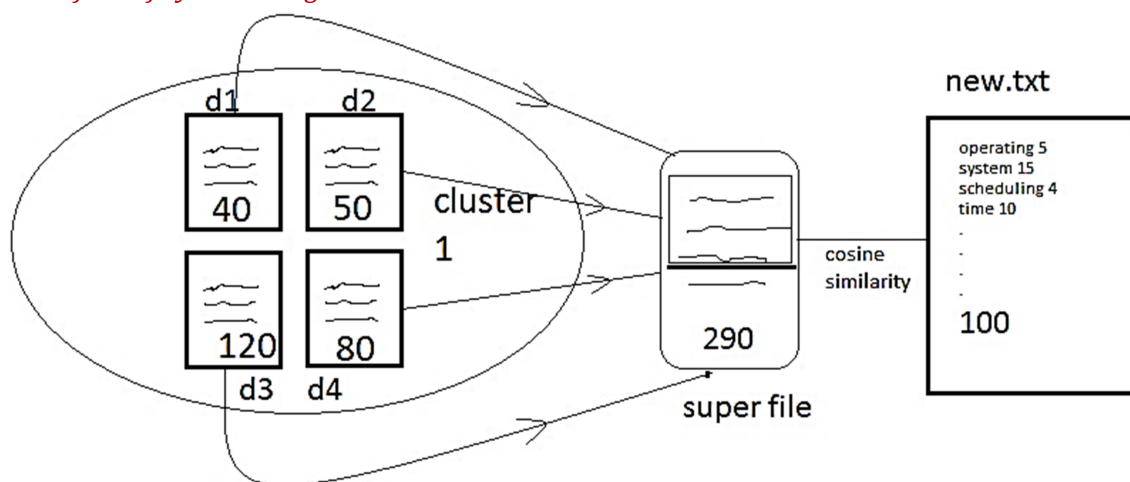


Figure 2. Work Flow of Proposed Methodology

1011

### Workflow of Keyword Selection

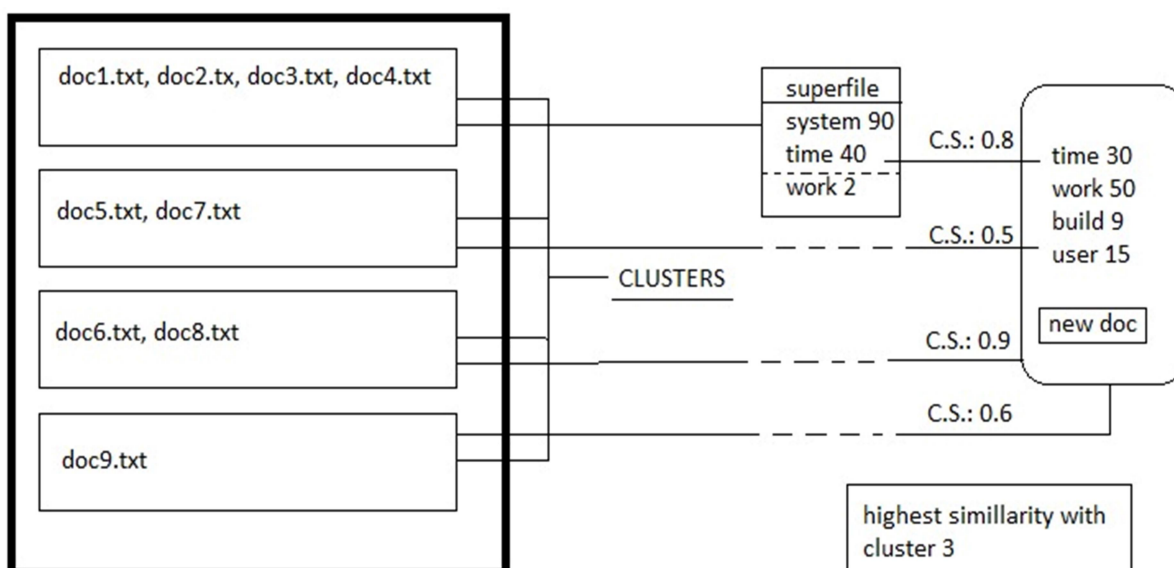


Figure 3. Overview of cluster selection



Explanation: In the diagram shown above (figure 3), there are 4 clusters. Among these clusters all documents are divided by their matching affinity. Example doc4.txt matches most with doc1, doc2, doc3 that is cluster 1, hence it lies in cluster 1. Each cluster contains a super file that contains all prominent keywords in cluster documents. When a new document "new.txt" appears and is to be sorted into one of the clusters or in a new cluster it's affinity with all clusters is matched. Affinity is decided by cosine similarity function whose values lie between 0 to 1. Here it is 0.8,0.5,0.9,0.6 in clusters (1,2,3,4); it is the highest with cluster 3 (0.9) that means the document is almost same as cluster 3 documents and shows similarity above 70%. Therefore new.txt is sorted into cluster 3.

## Modules Description

### Document Repository

Files pertaining to a specific project component, person, or organization can be stored in a document.

### Document Fetcher

Document Fetcher is a desktop search program that is both free and open source. It runs on Windows, Mac and Linux OS X and is developed in Java. The application's graphical user interface is built with the Standard Widget Toolkits. The application is an indexing search tool, which means that instead of looking through all of your files, it analyses a local database of file content. This implies that the application must be running at all times to keep track of changes, but search results are immediate. Apache Lucene is used to power the search engines an open-source search engine that is extensively used. The algorithm of the above module is given below. [13]

### Algorithm for DOCUMENT\_FETCHER

1. Start
2. Open and read (Buffered Reader) the new document with the given address (main directory).
3. Propagate to the fixed directory (main) containing all required documents' clusters as sub directories.
4. Make directory index of all clusters.
5. Pop from directory index as i.

- a) Propagate to selected  $i^{\text{th}}$  directory.
- b) Open and read super file (named super\_file.csv).
- c) Read each element from csv (selected stack) into a hash map "hm" with count as its property.
- d) Perform document similarity matching algorithm.
- e) Store results of document affinity.
- f) If directory index is null,
  - i. End.
  - ii. Else, go to step 4.

### Algorithm for Keyword Selection

1. Start.
2. Open and read the new document (whose keywords are to be added).
3. Filter general terms from the text and store all keywords in an Array List.
4. From the available hash map of super file, search each keyword from array list. 1012
5. If keyword found in map,
  - a. Increase hash map count of keyword by 1.
  - b. Else, create a new element in hash map, assigning count as 1.
6. Create formulae for selection:
  - a. Calculate total count in hash map (sum of all keyword counts) as 'total'.
  - b. Calculate total number of keywords in hash map as 'n'.
  - c. Average count per word (avg) = total / n.
  - d. Threshold =  $k * \text{avg}$ . 'k' is the normalizing factor.
7. For all keywords in hash map.
  - a. If word frequency (count) is greater than threshold, push into selected stack.
  - b. Else push in rejected stack.
8. Create new super file with selected and rejected stacks in csv format.
9. End.

### Result Analysis



**PURITY:** To compute purity, every cluster is assigned to the elegance that's maximum common within side the cluster, after which the accuracy of this task is measured via way of means of counting the quantity of efficaciously assigned files and dividing via way of means of N.

For example, the quest effects for jaguar includes 3 lessons similar to the 3 senses car, animal, and working system.

**Table 1.** Result Analysis using cluster purity factor

Number of Documents	Number of Clusters generated	Number of Similarities in Cluster							Purity
		1	2	3	4	5	6	7	
50	2	15	20						0.7
100	3	20	30	15					0.65
150	4	30	15	20	25				0.6
200	2	80	25						0.525
250	5	15	60	50	92	28			0.98
300	3	90	82	75					0.82
350	6	75	40	30	25	48	50		0.76
400	4	90	85	87	60				1.01
450	2	90	150						0.53
500	7	80	90	100	48	50	120		0.97
550	3	230	120	180					0.96
600	4	80	90	140	130				0.733
650	6	50	60	85	72	40	90		0.61
700	7	20	48	75	89	95	120	50	0.71
750	5	45	73	78	51	67			0.41
800	2	129	248						0.47
850	3	422	128	343					1.05
900	4	327	189	193	219				1.03
950	5	282	271	378	349	241			1.60
1000	6	121	199	327	233	292	127		1.29

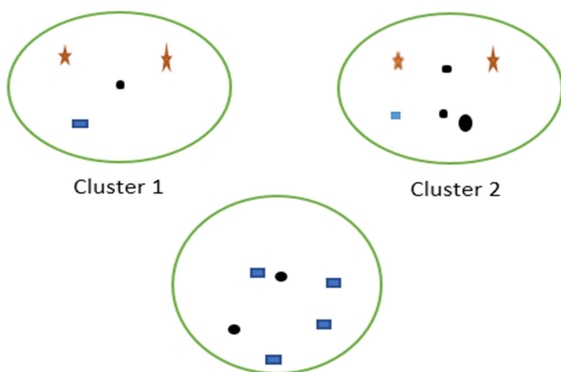


Figure: purity

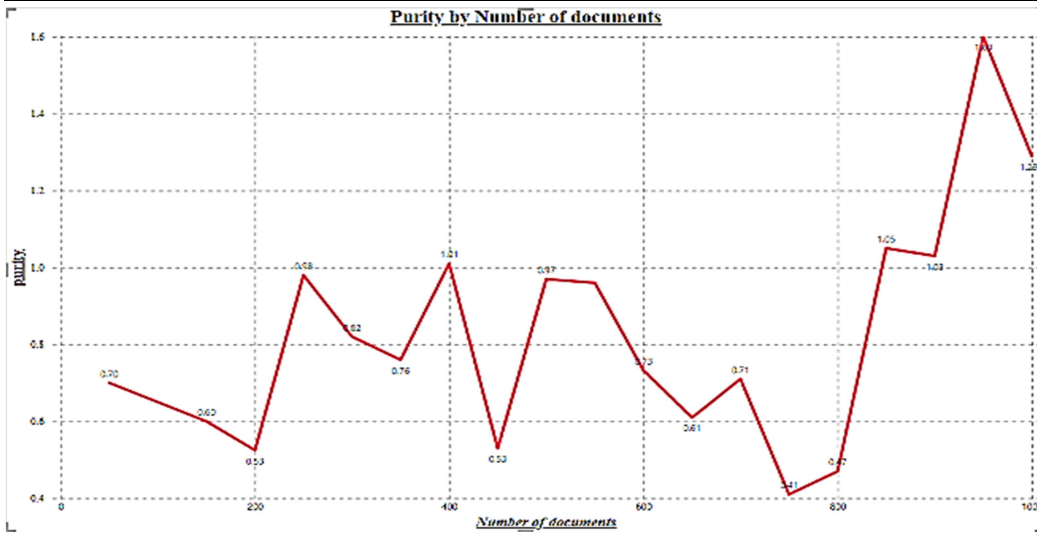
★ 2(cluster1); 3 (cluster2); 4(cluster3)

Purity is  $(1/16) \times (2+3+4) = 0.56$

$$\text{Purity} (\Omega, C) = 1/N \sum_k^j \max | \omega_k \cap C_j |$$

Where:  $\Omega = \{\omega_1, \omega_2, \omega_3, \dots, \omega_k\}$  is the set of clusters and  $C = \{c_1, c_2, c_3, \dots, c_j\}$  set of classes.[14]





**Graph 1.** Purity by number of Documents

**Table 2.** Result analysis by considering time taken to generated cluster

No. of Documents	Cluster generation	Time taken for assigning proper cluster (by traditional method) * (in milli second)	Time taken for assigning proper cluster (by proposed method) (in milli second)
50	2	1.99	1.97
100	3	2.96	2.92
150	4	3.58	3.32
200	2	1.99	1.97
250	5	4.89	4.41
300	3	2.96	2.92
350	6	5.89	5.71
400	4	3.67	3.61
450	2	1.98	1.92
500	7	6.89	6.21
550	3	2.41	2.37
600	4	2.99	2.83
650	6	4.98	4.41
700	7	5.98	5.91
750	5	4.92	4.90
800	2	1.97	1.91
850	3	2.83	2.48
900	4	3.89	3.71
950	5	4.89	4.47
1000	6	5.51	5.43

1014

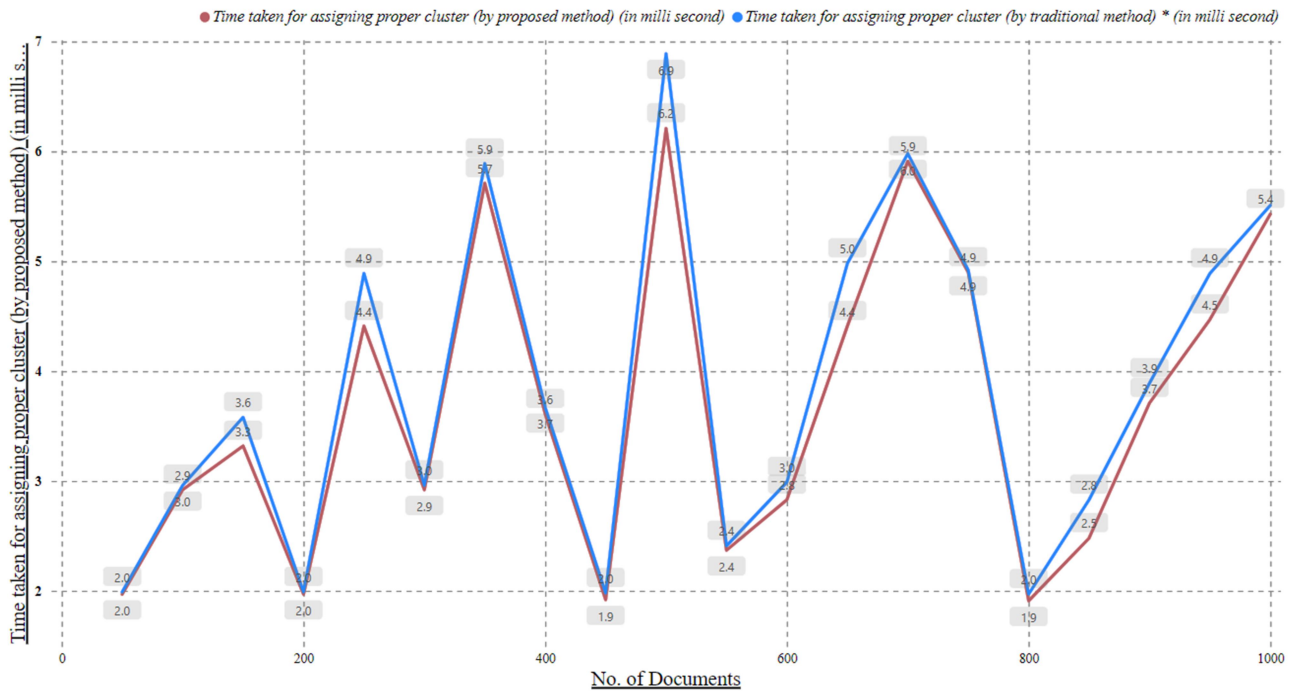
\*Traditional method

Suppose two cluster are there and these are 100,50 documents each of the cluster deciding proper cluster for a new incoming document is a tds job

because the similarity of the new document is matched with all the document of a cluster.



**Time taken for assigning proper cluster (by Proposed method) and (by Traditional methoy No. of Documents**



**Graph 2.** Time taken for assigning cluster (by Proposed method) and (by Traditional method) by number of Documents

**Conclusion**

There are numerous ways to assign documents to a cluster. In this method, we start with a cluster of many documents, combine them all into one super file, and use the cosine similarity measure technique to match a new document to this super file. This paper focussed on deriving a formula to filter super file's keywords such that it gives enhanced similarity result, that is similar and non-similar clusters of documents are polarised for differentiation. Filtering keywords of super File allows a faster algorithm for matching a new document than the conventional method by reducing document loading, processing and search time by reducing data size and keeping the most effective keywords only. The future scope in this paper can be to identify better formulae which filters in prominent features of the cluster in super File, speeding up the process and reducing the large size of super file.

**References**

Thompson, Victor U., Christo Panchev, and Michael Oakes. "Performance evaluation of similarity measures on similar and dissimilar text retrieval." 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K). Vol. 1. IEEE, 2015.

Manicassamy, Jayanthi, and P. Dhavachelvan. "Rank based clustering for document retrieval from biomedical databases." arXiv preprint arXiv:0912.2307 (2009).

Lahitani, Alfirna Rizqi, Adhistya Erna Permanasari, and Noor Akhmad Setiawan. "Cosine similarity to determine similarity measure: Study case in online essay assessment." 2016 4th International Conference on Cyber and IT Service Management. IEEE, 2016.

Niwattanakul, Suphakit, et al. "Using of Jaccard coefficient for keywords similarity." Proceedings of the international multicongference of engineers and computer scientists. Vol. 1. No. 6. 2013.

Dice, Lee R. "Measures of the amount of ecologic association between species." *Ecology* 26.3 (1945): 297-302. Dice, Lee R. "Measures of the amount of ecologic association between species." *Ecology* 26.3 (1945): 297-302.

Obilor, Ezezi Isaac, and Eric Chikweru Amadi. "Test for significance of Pearson's correlation coefficient." *International Journal of Innovative Mathematics, Statistics & Energy Policies* 6.1 (2018): 11-23.

<http://rosalind.info/glossary/euclidean-distance/>

Shirkhorshidi, Ali Seyed, Saeed Aghabozorgi, and Teh Ying Wah. "A comparison study on similarity and dissimilarity measures in clustering continuous data." *PloS one* 10.12 (2015): e0144059.

Nicholas O. Andrews and Edward A. Fox, "Recent Developments in Document Clustering", thesis, Deptt. Of CS, Virginia Tech, October 16, 2007.

Seung-sikh, "Keyword based document clustering", report, school of cs, kookim university.seoul, korea.

Jaccard Similarity - an overview | ScienceDirect Topics.

Cosine Similarity - an overview | ScienceDirect Topics.

Hill, Emily, David Shepherd, and Lori Pollock. "Exploring the use of concern element role information in feature location evaluation." 2015 IEEE 23rd International Conference on Program Comprehension. IEEE, 2015.

Sripada, Satya Chaitanya, and M. Sreenivasa Rao. "Comparison of purity and entropy of k-means clustering and fuzzy c



means clustering." Indian journal of computer science and engineering 2.03 (2011).

N. Lal, M. Singh, S. Pandey, and A. Solanki, "A Proposed Ranked Clustering Approach for Unstructured Data from Dataspace using VSM," 2020 20th International Conference on Computational Science and Its Applications (ICCSA), Cagliari, Italy, 2020, pp. 80-86.  
DOI: 10.1109/ICCSA50381.2020.00024.

Niranjan Lal, Shamimul Qamar, Monika Kalra, "K- Mean Clustering Algorithm Approach for Data Mining of Heterogeneous Data" Information and Communication Technology for Sustainable Development (ICT4SD), LNNS, Springer Proceeding, Volume 10, pp. 61-70, 2017.

