Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023

DOCUMENT CLUSTERING USING TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY, HASH VECTORIZER, LATENT SEMANTIC ANALYSIS (LSA) AND K-MEANS CLUSTERING

P Saidesh Kumar, Research Scholar, University College of Engineering, Osmania University, Hyderabad, India : saideshp@gmail.com
Dr P Vijayapal Reddy, Prof., HOD CSE, Matrusri Engineering College, Hyderabad : drpvijayapalreddy@gmail.com

Abstract
Document clustering is one of the most essential strategies when it comes to arranging files in an unsupervised way. A document clustering approach that makes use of Term Frequency – Inverse Document Frequency (TF-IDF), Hash vectorizer, Latent Semantic Analysis (LSA), and K-Means clustering is presented in this paper. Text vectorization is accomplished by using both TF-IDF and Hash vectorizer in tandem. Through the use of LSA, the dimensionality of the data may be decreased. K-Means clustering is then applied to the smaller collection of characteristics in order to cluster them. The proposed methodology of document clustering is both quick and effective by making use of language models and artificial intelligence algorithms that reduce the distances between documents that are contained within the same cluster while increasing the distances between clusters. The clustering time recorded is 0.12 seconds. The homogeneity is 0.4, the compactness is 0.451, the V-measure is 0.424.

Keywords: Document Clustering, Term Frequency – Inverse Document Frequency (TF-IDF), Hash vectorizer, Latent Semantic Analysis (LSA), K-Means clustering

I. Introduction

Document clustering is one of the most essential strategies when it comes to arranging files in an unsupervised way. The clustering algorithms may be used even when the documents in question are just represented as set of words [1]. The dimensions of the document space are always rather vast, and they might range anywhere from a few hundred to several thousand. Due to the fact that dimensionality is a curse, it is appropriate to project the documents initially into a lower dimensional subspace in which the semantic structure of the document space becomes obvious. This will reduce some of the issues caused by the curse of dimensionality. In low-dimensional semantic spaces, one may make use of the more conventional clustering procedures.

The effort of arranging a collection of documents, the categorization of which is uncertain, into meaningful groupings (clusters) that are homogenous according to some idea of closeness (distance or similarity) among documents. This work is referred to as document clustering. The term "document clustering" refers to the process of organizing a huge quantity of text documents into a limited number of relevant clusters, where each cluster is intended to represent a certain subject area [2]. Document The technique of grouping a set of document collections into a variable number of groups depending on the degree to which the document contents are similar is known as clustering [3]. The practice of grouping documents with similar characteristics into partitions, with the goal of having papers within the same partition demonstrate a greater degree of similarity among themselves than they do to any other document in any other partition.

Clustering is the digital equivalent of physically sorting your papers into different boxes and grouping them [4], with the goal of ensuring that items are only placed in the same box if they should be there. This gives you the ability to navigate and manage your documents by browsing through a relatively limited collection of boxes, often known as clusters, rather than delving directly through the considerably larger data set of documents themselves. It does not rely on any preconceived notions or search phrases in order to arrange the documents in accordance with the structure that occurs

organically. It does this by assigning a set of keywords to each cluster, which then provides a concise summary of the cluster. In addition to this, it defines a "representative document" that may be substituted in place of the cluster as necessary.

1.1 Applications of document clustering

Clustering is the most prevalent kind of unsupervised learning and is a key technique that is used in a number of applications across a wide variety of commercial and scientific disciplines [5]. In the following, we will provide a brief overview of the primary applications of clustering.

- **Finding Similar Documents:** This function is used rather often when the user has identified one "excellent" document in a set of search results and wants other documents that are similar to the one they have identified. Clustering is able to uncover papers that are conceptually identical, in contrast to search-based techniques, which are only able to detect whether the documents have many of the same terms. This is an intriguing quality that clustering has, and it sets it apart from other methods.
- **Organizing Large Document Collections:** Document retrieval focuses on locating documents that are relevant to a given query; nevertheless, it does not address the challenge of making sense of a huge number of documents that have not been classified. The aim here is to arrange these papers in a taxonomy that is similar to the one that humans would build if given sufficient time, and then utilize that taxonomy as a browsing interface to access the original collection of documents.
- **Detection of Duplicate Content:** There is a need, present in a great variety of applications, to locate duplicates or near-duplicates in a huge number of documents. Clustering is used for the detection of plagiarism, the grouping of news pieces that are related, and the reordering of the ranks of search results (to assure higher diversity among the topmost documents). Take note that the description of clusters is only required seldom in applications of this kind.
- **Recommendation System:** Within the context of this program, a user will get suggestions for more articles to read depending on the articles that user has previously perused. The articles are grouped together, which not only makes it feasible to do so in real time but also significantly enhances the quality.
- **Search Optimization:** The user query can be first compared to the clusters rather than comparing it directly to the documents, and the search results can also be easily arranged by clustering, which is a huge help in improving the quality and efficiency of search engines. Clustering also provides a lot of assistance in enhancing the efficiency of search engines.

Although the method of document clustering has been the focus of study for many decades, the problem is still far from being straightforward or addressed. The following constitute the challenges:

1. The process of determining which aspects of the texts are significant and might be employed in the clustering procedure.
2. Determining the most effective method for quantifying the degree to which two publications are similar.
3. Selecting an efficient method for clustering by making use of the similarity measure that was introduced previously in this paragraph.
4. Implementing the clustering approach in a way that is not only successful but also feasible in terms of the amount of memory and processing power that it demands.
5. Finding a number of methods to measure how successful the clustering that was done really was.

This paper presents a document clustering method using TF-IDF, Hash vectorizer, LSA and K-Means clustering. Text vectorization is performed using a combination of TF-IDF and Hash vectorizer. The dimensionality of the data is reduced by using LSA. The reduced set of features are clustered using K-Means clustering. Section II presents the literature review. Section III discusses the proposed methodology. Section IV presents the experimental analysis followed by conclusion and references.

II. Literature

Sakib Al Hasan et al [6], provided the most up-to-date findings from research carried out on Bangla Document Clustering. The final purpose of this study is to accomplish the target of testing K-Means clustering and Mini-Batch K-Means clustering algorithms and analyzing the performance of these algorithms for Bangla news text data using silhouette score and homogeneity score.

Wasseem N. Ibrahem Al-Obaydy et al [7], a technique to the categorization of documents is offered, with the goal of clumping the text documents of research papers into expressive groups that cover the same kind of scientific ground. In the process of constructing the suggested strategy, the primary focus and scopes of target groups were established; Each group includes a number of different themes. The word tokens were recovered one at a time from different subjects that were connected to the same group. The term frequency-inverse document frequency (TF-IDF) numerical statistic is used to determine a document's weight, and the frequency with which word tokens occur several times inside a document has an effect on the weight of the document. In order to carry out the classification process, the suggested method makes use of the paper's title, abstract, and keywords, in addition to the themes that are associated with the categories. For the purpose of categorizing and grouping the texts into key categories, we made use of a technique called the K-means clustering method. In order to establish the cluster centers, the K-means method makes use of the category weights (or centroids).

Zakky Nilem Sanjifa et al [8], intends to detect and evaluate community comments via the official government site using the K-means clustering and Latent Semantic Analysis (LSA) techniques in order to provide subjects of development concerns that are currently occurring. After obtaining the subject matter, the next step is to interpret it into categories in accordance with the requirements of a program known as RKPD. In the process of clustering using k-means clustering and LSA, 17 clusters were formed. These clusters were divided into the following five categories: public service, infrastructure, city utilities, living environment, transportation system and mass transport, and education service. Public service occupied the first position in this cluster.

A clustering-based classifier for Arabic text documents is proposed by Arun Kumar Sangaiah et al [9]. After document preparation, which involves deleting stop words and obtaining the root for each term in each document, we use k-means, incremental k-means, Threshold + k-means, and k-means with dimensionality reduction. Then, use a strategy that weights terms to determine the significance of each phrase in relation to its respective text. Then, use a technique for measuring similarity that may be applied to each document and its relationship to the other papers. In addition, we will be calculating accuracy using the F-measure, entropy, and support vector machine (SVM).

Naveen Kumar et al [10], suggested results in an increase in both the effectiveness and precision of the document clustering. The work that was suggested in this research included the following phases: preprocessing, a term document matrix, and the use of a clustering method. After displaying a flowchart initially, it goes on to compute the TF, IDF, and TF-IDF values, respectively. After that, K-means clustering is performed in order to divide the data into several groups according to their degree of similarity.

Jing ZHU et al [11], K-Means clustering based on TF-IDF was suggested, and tests were carried out using 24 software requirement papers written in Chinese language. In the modern day, there is no way to automatically acquire the function points when employing the approach known as function point analyze (FPA), and this is particularly true for the requirement papers that are written in Chinese. When words are segmented in Chinese, it is necessary to divide them accurately so that subsequent entity recognition and disambiguation can be carried out in a smaller range. This lays a solid foundation for the efficient automatic extraction of function points.

Peng Yang et al [12] proposed that a probabilistic model to calculate the prior Latent Dirichlet Allocation (LDA) distribution, find the latent topics, and achieve better clustering. This model would integrate various degrees of information on the popularity of the subject. To be more specific, worldwide subject popularity is being implemented to lessen the possibility of distraction in local cluster popularity, while local cluster popularity is being implemented to focus more emphasis on

certain aspects of global topic popularity. Both measures of popularity give information that is complimentary to one another, and their combination might cause statistical parameters of the model to be dynamically adjusted.

R. Janani et al [13], a brand-new approach called Spectral Clustering with Particle Swarm Optimization (SCPSO) is proposed with the intention of improving the clustering of text documents. The randomization process is carried out using the starting population while taking into account both the global and the local optimization function. In order to manage the vast quantity of text documents, the purpose of this line of study is to investigate the possibility of combining spectral clustering with swarm optimization. The suggested algorithm SCPSO is tested using the benchmark database in comparison to the other methods that are already in use. Comparisons are made between the Spherical K-means method, the Expectation Maximization Method (EM), and the traditional PSO Algorithm using the newly developed algorithm, SCPSO.

Laith Mohammad Abualigah et al [14], MHKHA stands for multi-objective hybrid KH algorithm, and it is offered as a solution to the issue of text document clustering. The objective functions are combined with hybrid KH algorithms. The starting solutions of the KH algorithm are inherited from the k-mean clustering method in this implementation, and the choice on clustering is based on two combined objective functions. The performance of the suggested algorithms is evaluated using nine text standard datasets obtained from the Laboratory of Computational Intelligence (LCI). Accuracy, precision, recall, the F-measure, and convergence behavior are the five metrics that are used in the assessment process. Comparisons are made between the variants of the KH algorithm that have been suggested and thirteen other algorithms that have been published and are known to be effective in the field of clustering.

Ammar Kamal Abasi et al [15], a solution to the text FS issue is presented in the form of a binary grey wolf optimizer (BGWO) algorithm. Selecting relevant characteristics from the text is the first step in this novel version of the GWO algorithm, which is introduced by this approach. The clustering method known as k-means is used to do an analysis of these useful qualities in order to cut down on the amount of time required for the clustering algorithm's execution while simultaneously increasing its overall effectiveness. The effectiveness of BGWO is evaluated using six different published datasets, which are Tr41, Tr12, Wap, Classic4, 20Newsgroups, and CSTR respectively. Based on the evaluation's measures, the findings revealed that the output of the BGWO algorithm performed better than any of the other algorithms that were tested, including GA and BPSO.

Qi-Zhu Dai et al [16], offered a brand-new clustering technique that uses the natural reverse nearest neighbor structure as its foundation. We name this approach the RNN-NSDC. In the first step of the method, "core objects" are extracted by using the "reverse nearest neighbors" technique. Second, in order to cluster, our approach makes advantage of the knowledge on the core objects' neighbor structures. As long as noise effects are taken into account, core sets are able to accurately depict the structure of clusters. Therefore, the RNN-NSDC is able to determine the ideal cluster numbers for the datasets even when they comprise clusters of various forms and outliers.

III. Proposed Model

The input documents are first vectorized using a hybrid model of Term Frequency-Inverse Document Frequency (TF-IDF) and Hash Vectorizer. The extracted features are then sent to Latent Semantic Analysis (LSA) for dimensionality reduction. The reduced set of features are sent to K Means clustering.

1. in the below figure, after clustering box, please add box related to output
2. in the below figure, all are rectangles. Flowchart should use different types of boxes

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023

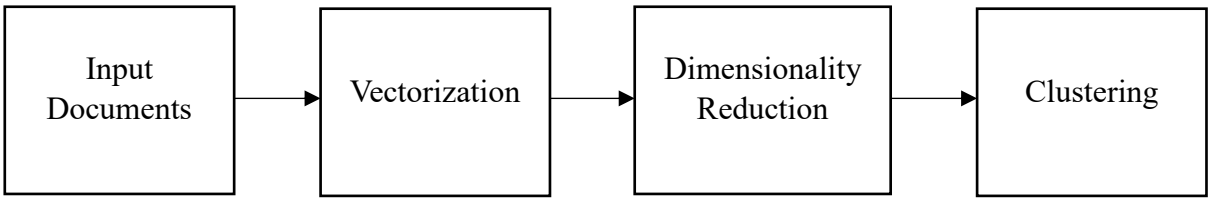

Figure 1: Proposed model

3.1 Text Vectorization

The process of converting text data into numerical data requires certain clever methods, which are collectively referred to as vectorization. In the field of natural language processing (NLP), this process is known as word embeddings. The process of transforming text data to numerical vectors is referred to as vectorization. Another name for this process is word embedding. Following that, such vectors are included into a variety of machine learning models. In this sense, we refer to this as the extraction of characteristics from text with the assistance of the goal of building various natural languages, processing models, and so on. In the next paragraphs of this post, you will learn about the many methods we have available to transform the text input into numerical vectors.

section 3.2, 3.3, 3.4 and 3.5 are old ones only. No need to explain elaborately. Cut short.

3.2 TF-IDF

In natural language processing and information retrieval, one of the most used statistical methods is called Term Frequency - Inverse Document Frequency (TF-IDF for short). It determines the importance of a term inside a document in comparison to the importance of the word within a collection of documents (i.e., relative to a corpus). A text vectorization technique will turn the words inside a text document into numerical representations of their relative value. There are several different scoring strategies for text vectorization, with TF-IDF being one of the most prominent. However, there are many more scoring techniques available.

The Term Frequency-Inverse Document Frequency (TF-IDF) algorithm vectorizes and scores a word by multiplying the Term Frequency (TF) of the word with the Inverse Document Frequency (IDF).

The term frequency (TF) of a term or word refers to the ratio of the number of times the term occurs in a document to the total number of words in the document. TF may also refer to the number of times a word appears in a document.

$$TF = \frac{number\ of\ times\ the\ term\ appears\ in\ the\ document}{total\ number\ of\ terms\ in\ the\ document}$$

The Inverse Document Frequency (IDF) of a phrase indicates the percentage of total documents in the corpus that are comprised of documents that include the term. Words that are only found in a limited proportion of papers (for example, terms used in technical jargon) are given significance ratings that are greater than words that are included in all publications (e.g., a, the, and).

$$IDF = \log\left(\frac{number\ of\ the\ documents\ in\ the\ corpus}{number\ of\ documents\ in\ the\ corpus\ contain\ the\ term}\right)$$

When determining a term's TF-IDF, the scores for both TF and IDF are multiplied together.

$$TF - IDF = TF * IDF$$

Vectorization of text, which may then be utilised for feature mining, is accomplished by the use of the Term Frequency' – Inverse Document Frequency' algorithm. TF-IDF is composed of two distinct elements. First, there is the term frequency, which is the total number of times a particular term appears in the text document in comparison to the total number of words in the text. Next, there is the inverse document frequency, which determines how much information a single word contributes to the overall document. It determines how important a certain word is to the overall significance of the work. IDF displays the frequency with which a certain term appears in all of the texts. Calculating TF-IDF is as simple as multiplying TF by IDF.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023

$$TF(t_1) = \frac{c_{i,j}}{\sum_k c_{i,j}}$$

$c_{i,j}$ is the sum of all of the occurrences of word t_1 in a given text.
$\sum_k c_{i,j}$ is the total number of words counted across the whole of the text

$$IDF = \log(\frac{N}{df_t})$$

N – number of documents taken
df_t – number of documents that have term t_1
When translated into plain English, the significance of a word is high when it appears a lot in a certain document but infrequently in others. Other documents include it less often. In a nutshell, the ratio of rareness across documents, as measured by IDF, counterbalances the commonality, as measured by TF, inside a document. The TF-IDF score that was calculated as a consequence conveys the significance of a phrase for a given text within the corpus. The TF-IDF algorithm has various applications in the field of natural language processing. For instance, Search Engines employ TF-IDF to rate the relevance of a page to a query based on the amount of times the document is mentioned in the query. Text categorization, text summarization, and topic modelling are three further applications of the TF-IDF algorithm.

3.3 Hash Vectorization

A hashing vectorizer is a type of vectorizer that looks for the token string name that corresponds to the integer index mapping by employing the hashing trick. This vectorizer is responsible for converting text documents into matrices. More specifically, it takes a collection of documents and transforms them into a sparse matrix that contains the token occurrence counts. The illustration of how the Hash Vectorizer works may be seen down below in figure 2.

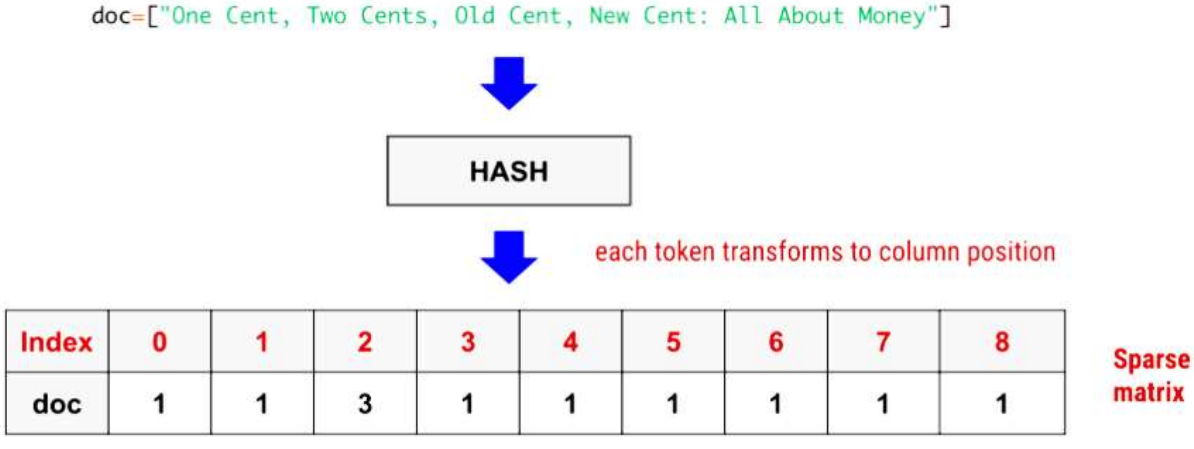

Figure 2: Hash Vectorizer Working

The resultant Hash Vectorizer object, when saved, would be considerably smaller and, therefore, much quicker to load back into memory when it was required to do so. This is because the vocabulary would not have to be stored. The fact that it will be unable to get the actual token given its location in the column is one of the drawbacks of acting in this manner. This would be of utmost significance in activities such as keyword extraction, in which it is necessary to obtain and make use of the real tokens. The hashing vectorizer offers the following benefits, because there is no need to store the vocabulary dictionary in the memory, it is extremely low memory scalable. This makes it suitable for use with huge data sets. It is possible to utilise it in a streaming or parallel pipeline since there is no state throughout the fitting process.

3.4 Latent Semantic Analysis (LSA)

Latent Semantic Analysis, often known as LSA, is a method for producing vector-based representations of texts that are said to capture the semantic content of the texts they are analysing.

The fundamental objective of LSA is to compare the vector representations of two texts in order to determine how similar those texts are to one another. It has been shown that this very simple similarity measure closely matches human capacities on a range of tasks, and it has been located within a psychological theory of the meaning of text. This article traces the evolution of LSA by providing a historical backdrop, demonstrating how LSA computes and makes use of its vector representations, and finally providing instances of the theoretical and empirical support for LSA as well as its current research prospects.

LSA, which was formerly known as Latent Semantic Indexing, was created for the problem of information retrieval, which entails picking, from a huge collection of documents, a few relevant documents that fit a given query. Originally, LSA was known as Latent Semantic Indexing. Matching keywords, weighted keyword matching, and vector-based representations based on the frequency with which words appear in texts were some of the previous methods used for tackling this challenge. LSA is an extension of the vector-based method that reconfigures the data using a technique called Singular Value Decomposition (SVD). The specifics of this process will be laid out in the following sections, but the overarching concept is that there is a group of underlying latent variables that encompasses all of the possible interpretations that may be communicated using a given language. It is expected that these variables do not interact with one another (and therefore orthogonal in the vector space). The SVD algorithm is a method from the field of matrix algebra that, in essence, re-orients and ranks the dimensions present in a vector space.

Because the dimensions in a vector space computed by SVD are ordered from most significant to least significant, if some of the less significant dimensions are ignored, the reduced representation is guaranteed to be the best possible for that dimensionality. This is because the order of the dimensions is determined by the SVD algorithm. The assumption that just the top few hundred dimensions (out of tens or even hundreds of thousands) are effective for capturing the meaning of texts is a common one in LSA. Because the representations are based on fewer dimensions, words that often appear in situations that are quite similar to one another will have vectors that are very similar to one another and will thus be given a high similarity grade. It is presumed that the dimensions that were thrown out were the result of noise, coincidental correlations, or some other feature that was deemed unnecessary. It should not come as much of a surprise that LSA fared better in terms of information retrieval than its competitors' techniques. The fact that it can replicate human behaviour so well across a wide range of language activities is even more astonishing. However, before discussing them, a more in-depth explanation of the LSA approach will be presented.

The following are the most often performed steps, however there are several variations:

- Gather a substantial amount of content that is relevant to the area and organise it into "documents." The assumption that the material included inside a paragraph is likely to be cohesive and connected underpins the practise of treating each paragraph as its own independent document in the majority of applications.
- Create a co-occurrence matrix of documents and words when this step is complete. The number of occurrences of the phrase y in document x is stored in the cell of this matrix that corresponds to document x and the term y. A word is considered to be a term if it is found in more than one document and no attempts at stemming or other morphological analysis are made to unite the many variants of the same word. If there are m terms and n documents, this matrix may be thought of as offering a representation that contains an m-dimensional vector for each document and an n-dimensional vector for each term. This interpretation is valid if there are both numbers of terms and documents.
- It is possible to provide different weights to the data contained inside each cell in order to mitigate the impact of frequently occurring terms across the corpus. The "log entropy" approach, which is derived from Information Theory and involves multiplying the value by its information gain, is a typical way for valuing data.

- The SVD algorithm is called with a parameter called k, which indicates the number of dimensions that should be used. (In theory, the SVD would be calculated using all of the dimensions to form three matrices that, when multiplied together, would recreate the original data. However, this is not possible because of the amount of memory that would be required to do so. Instead, the methods that are now being employed are optimised for working with sparse data spaces, and they only calculate the k dimensions of the matrices that are the most significant.

The processing that was done above resulted in the creation of three matrices. One has a k-dimensional vector for each individual document, a k-dimensional vector for each individual phrase in the corpus, and the k singular values. The first two matrices each define their own unique vector space, which is distinct from both of the spaces that were previously defined by the first matrix. A vector may be transformed from one space to another using the singular values as the transforming factors. The use of these matrices is dependent on the application being carried out.

The LSA representation of each document is included in the document vectors so that it may be retrieved from the database. In the document vector space, a query may be transformed into a "pseudo doc" by merging the vectors that correspond to the phrases in the query and then dividing the result by the singular values. The cosine between two vectors is often computed in order to make a comparison between them. (Other distance measures may be used by some applications.) The documents that are most similar to the query in terms of their meaning are represented by vectors that are closer together in the document vector space (according to LSA). In the vast majority of the remaining applications, the original texts are only used during the training process, also known as the generation of the semantic space. Combining the word vectors in the way that was just explained allows for the comparison of new texts. Because of the comparison that takes place in the term space, there is no need for any manipulation with the singular values in this scenario.

Algorithm

- The core idea is to take a matrix of documents and terms and try to decompose it into separate two matrices –
    - A document-topic matrix
    - A topic-term matrix.
- Therefore, the learning of LSA for latent topics includes matrix decomposition on the document-term matrix using Singular value decomposition.

Step-1: The first step is to generate a document-term matrix of shape m x n in which each row represents a document and each column represents a word having some scores.

Step-2: Perform dimensionality reduction on document term matrix, M.

$$M=U*S*V^T$$

Orthogonal matrix, (U)
Diagonal matrix. (S)
Orthogonal matrix Transpose, (V^T)
where S is a diagonal matrix having diagonal elements as the singular values of M.

3.5 K means clustering

The K-means clustering method will loop until it locates the best centroid, after which it will calculate the centroids. It presupposes that one is already familiar with the total number of clusters. Another name for this approach is the flat clustering algorithm. The letter 'K' in the K-means method stands for the number of data clusters that were determined by the algorithm. The data points are put into a cluster using this approach in such a way that the total of the squared distances between the data points and the cluster's centroid is kept to a minimum. This ensures that the process works properly. It is important to realize that having less variety within the clusters will result in more data points that are comparable to one another within the same cluster.

With the aid of the stages that are listed below, we will be able to comprehend how the K-Means clustering method works.

- **Step 1** − Determine how many clusters, denoted by the letter K, will need to be produced in the input features.
- **Step 2** − Choose K clusters at random and then place each data point into one of the clusters. The data should be categorised according to the quantity of data points.
- **Step 3** − Calculate the centroids of each cluster.
- **Step 4** − Continue iterating the following steps until the ideal centroid is identified, which is the assignment of data points to the clusters that are not changing any more –
    - 4.1 − The first step would include computing the sum of the squared distances between the data points and the centroids.
    - 4.2 − Place the each data point in the cluster that is physically located the closest to it relative to the other clusters (centroid).
    - 4.3 − Take the average of all of the data points that belong to a cluster, and then use that number to calculate the centroids for the clusters.

K-means employs a method known as expectation-maximization in order to resolve the issue. The data points are assigned to the cluster that is geographically closest to them via the Expectation phase, and the centroid of each cluster is determined through the Maximization step.

Whenever we are dealing with the K-means algorithm, it is imperative that we pay attention to the following:

- It is advised that standardisation of the data be performed when dealing with clustering algorithms such as K-Means. This is due to the fact that such algorithms employ a distance-based measurement to assess the degree of similarity between data points.
- It's possible that K-Means will become stuck in a local optimum and won't converge to a global optimum since it's an iterative algorithm and the centroids are initially chosen at random. Because of this, it is strongly suggested that several initializations of centroids be used.

IV. Experimental Results

This section presents the experimental analysis carried out to validate the proposed model. The dataset named “20newsgroups” consists of documents that belong to four categories namely:

- "alt.atheism",
- "talk.religion.misc",
- "comp.graphics",
- "sci.space",

The clustering result is validated with the help of the following parameters:

- Clustering Time, the time it takes to do clustering.
- Homogeneity, which quantifies how much clusters contain only members of a single class; It is defined as below

$$h = 1 - \frac{H(C/K)}{H(C)}$$

H(C|K) represents the ratio between the number of samples labelled c in cluster k and the total number of samples in cluster k.

- Completeness, which quantifies how much members of a given class are assigned to the same clusters;

$$c = 1 - \frac{H(K/C)}{H(K)}$$

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023

H(K|C) represents the ratio between the number of samples labelled *c* in cluster *k* and the total number of samples labelled *c*.

- V-measure or Normalised Mutual Index (NMI), the harmonic mean of completeness and homogeneity;

$$NMI = 2 * \frac{h * c}{h + c}$$

Table 1 shows the values obtained by the proposed model. The clustering time recorded is 0.12 seconds. The homogeneity is 0.4, the compactness is 0.451, the V-measure is 0.424.

Table 1: Experimental analysis

Metric	**Value**
Clustering time	0.12 seconds
Homogeneity	0.400
Completeness	0.451
V-measure	0.424

Table 2: Comparative analysis

Evaluation parameters	**Count Vectorizer with K Means**	**Count Vectorizer + Mini Batch K means [6]**	**TF-IDF + K Means [7][10][11]**	**TF-IDF + LSA + K Means [17]**	**Hash Vectorizer + TF-IDF + LSA + K Means**
Clustering Time	0.46 seconds	0.15 seconds	0.39 seconds	0.25 seconds	**0.12 seconds**
Homogeneity	0.007	0.014	0.343	0.393	**0.400**
Completeness	0.031	0.020	0.404	0.404	**0.451**
V-measure	0.011	0.016	0.370	0.398	**0.424**

Table 2 shows the comparative analysis of the proposed model with existing techniques. Clustering Time for Count Vectorizer with K Means, Count Vectorizer + Mini Batch K means, TF-IDF + K Means, TF-IDF + LSA + K Means and Hash Vectorizer + TF-IDF + LSA + K Means is 0.46, 0.15, 0.39, 0.25 and 0.12 seconds respectively. Homogeneity for Count Vectorizer with K Means, Count Vectorizer + Mini Batch K means, TF-IDF + K Means, TF-IDF + LSA + K Means and Hash Vectorizer + TF-IDF + LSA + K Means is 0.007, 0.014, 0.343, 0393 and 0.400 respectively. Completeness for Count Vectorizer with K Means, Count Vectorizer + Mini Batch K means, TF-IDF + K Means, TF-IDF + LSA + K Means and Hash Vectorizer + TF-IDF + LSA + K Means is 0.031, 0.020, 0.404, 0.404 and 0.451 respectively. V-measure for Count Vectorizer with K Means, Count Vectorizer + Mini Batch K means, TF-IDF + K Means, TF-IDF + LSA + K Means and Hash Vectorizer + TF-IDF + LSA + K Means is 0.011, 0.016, 0.370, 0.398 and 0.424 respectively.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023

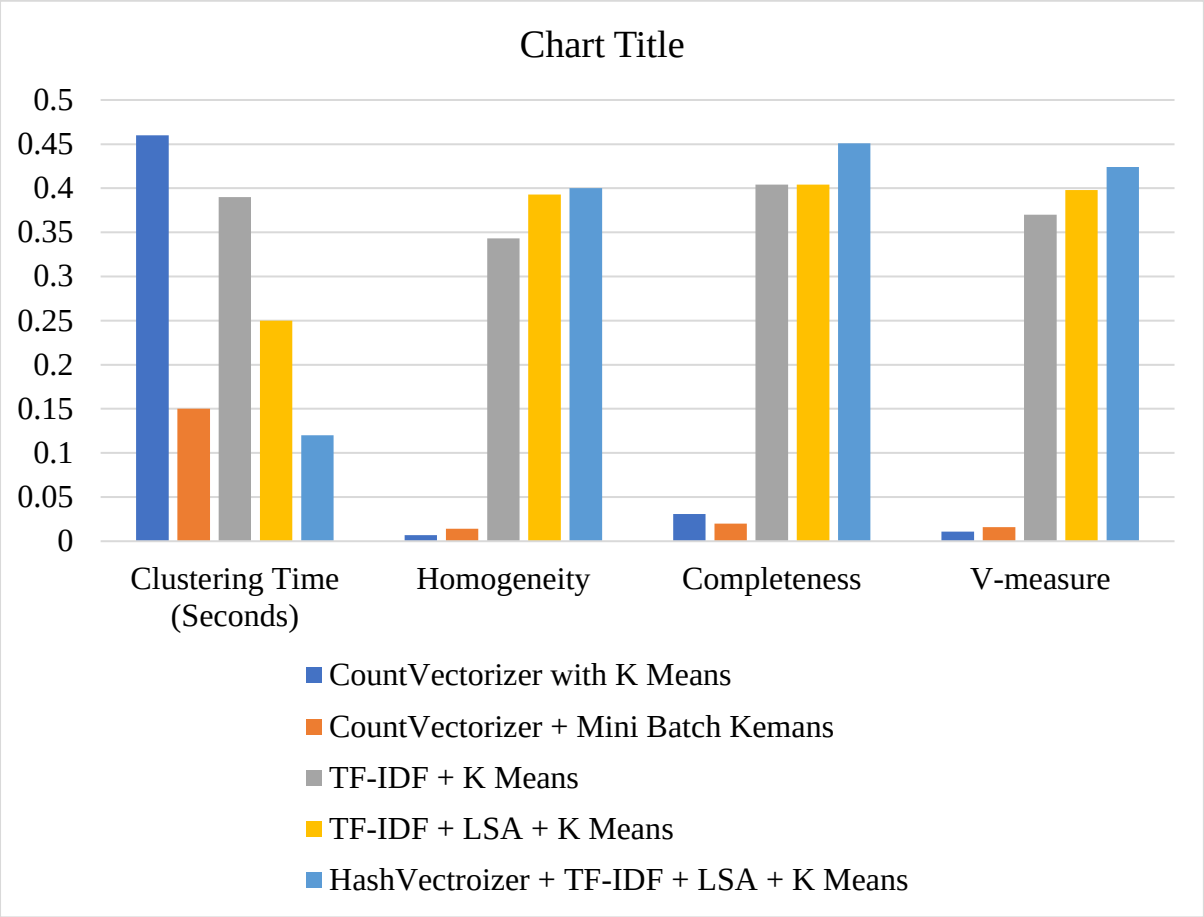

Figure 3: Comparative analysis of the proposed model

V. Conclusion

The process of dividing a set of document collections into a variable number of groups based on the degree to which the document contents are similar is known as document clustering. The process of grouping documents with similar characteristics into partitions, with the goal of having documents within the same partition exhibit a higher degree of similarity among themselves than they do to any other document in any other partition. In this paper, a strategy to document clustering is described that makes use of TF-IDF, Hash vectorizer, Latent Semantic Analysis (LSA), and K-Means clustering. TF-IDF is an acronym that stands for Term Frequency – Inverse Document Frequency. Text vectorization is achieved by using TF-IDF and Hash vectorizer in conjunction with one another in order to get the desired results. The dimensionality of the data is reduced by using LSA. The K-Means clustering method is then used on the reduced set of attributes in order to group them. The time that was reported for clustering was 0.12 seconds. It has a compactness of 0.451, homogeneity of 0.4, a V-measure of 0.424.

References

[1] Cozzolino, Irene, and Maria Brigida Ferraro. "Document clustering." Wiley Interdisciplinary Reviews: Computational Statistics (2022): e1588.
[2] Weißer, Tim, Till Saßmannshausen, Dennis Ohrndorf, Peter Burggräf, and Johannes Wagner. "A clustering approach for topic filtering within systematic literature reviews." MethodsX 7 (2020): 100831.
[3] Curiskis, Stephan A., Barry Drake, Thomas R. Osborn, and Paul J. Kennedy. "An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit." *Information Processing & Management* 57, no. 2 (2020): 102034.
[4] Berry, Michael W., Azlinah Mohamed, and Bee Wah Yap, eds. Supervised and unsupervised learning for data science. Springer Nature, 2019.

[5] Sarker, Iqbal H. "Data science and analytics: an overview from data-driven smart computing, decision-making and applications perspective." SN Computer Science 2, no. 5 (2021): 1-22.
[6] Al Hasan, Sakib, Wang Ruiqin, and Md Gulzar Hussain. "Clustering Analysis of Bangla News Articles with TF-IDF & CV Using Mini-Batch K-Means and K-Means." In 2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), pp. 17-22. IEEE, 2022.
[7] Al-Obaydy, Wasseem N. Ibrahem, Hala A. Hashim, Yassen AbdulKhaleq Najm, and Ahmed Adeeb Jalal. "Document classification using term frequency-inverse document frequency and K-means clustering." Indonesian Journal of Electrical Engineering and Computer Science 27, no. 3 (2022): 1517-1524
[8] Sanjifa, Zakky Nilem, Surya Sumpeno, and Yoyon Kusnendar Suprapto. "Community Feedback Analysis Using Latent Semantic Analysis (LSA) to Support Smart Government." In 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA), pp. 428-433. IEEE, 2019.
[9] Intani, Sheila Maulida, Bahrul Ilmi Nasution, Muhammad Erza Aminanto, Yudhistira Nugraha, Nurhayati Muchtar, and Juan Intan Kanggrawan. "Automating Public Complaint Classification Through JakLapor Channel: A Case Study of Jakarta, Indonesia." In 2022 IEEE International Smart Cities Conference (ISC2), pp. 1-6. IEEE, 2022.
[10] Kumar, Naveen, Sanjay Kumar Yadav, and Divakar Singh Yadav. "An Approach for Documents Clustering Using K-Means Algorithm." In Innovations in Information and Communication Technologies (IICT-2020), pp. 453-460. Springer, Cham, 2021.
[11] Zhu, Jing, Song Huang, Yaqing Shi, Kaishun Wu, and Yanqiu Wang. "A Method of K-Means Clustering Based on TF-IDF for Software Requirements Documents Written in Chinese Language." IEICE Transactions on Information and Systems 105, no. 4 (2022): 736-754.
[12] Dounia, Rahhali, Khaider Yasssine, and En Nahnahi Noureddine. "Exploring text representation impact on K-means based arabic text documents clustering." In 2022 International Conference on Intelligent Systems and Computer Vision (ISCV), pp. 1-5. IEEE, 2022.
[13] Janani, R., and S. Vijayarani. "Text document clustering using spectral clustering algorithm with particle swarm optimization." Expert Systems with Applications 134 (2019): 192-200.
[14] Abualigah, Laith Mohammad, Ahamad Tajudin Khader, and Essam Said Hanandeh. "A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis." Engineering Applications of Artificial Intelligence 73 (2018): 111-125.
[15] Abasi, Ammar Kamal, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, Syibrah Naim, Sharif Naser Makhadmeh, and Zaid Abdi Alkareem Alyasseri. "An improved text feature selection for clustering using binary grey wolf optimizer." In *Proceedings of the 11th national technical seminar on unmanned system technology 2019*, pp. 503-516. Springer, Singapore, 2021.
[16] Dai, Qi-Zhu, Zhong-Yang Xiong, Jiang Xie, Xiao-Xia Wang, Yu-Fang Zhang, and Jia-Xing Shang. "A novel clustering algorithm based on the natural reverse nearest neighbor structure." *Information Systems* 84 (2019): 1-16.
[17] Ratna, Anak Agung Putri, Naiza Astri Wulandari, Aaliyah Kaltsum, Ihsan Ibrahim, and Prima Dewi Purnamasari. "Answer categorization method using K-means for Indonesian language automatic short answer grading system based on latent semantic analysis." In 2019 16th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering, pp. 1-5. IEEE, 2019.