Efficient text document clustering with new similarity measures

R. Lakshmi*

Department of Computer Science and Engineering,
K.L.N. College of Engineering,
Sivagangai District, Tamilnadu, India
Email: rlaksh_gopi@yahoo.com
*Corresponding author

S. Baskar

Department of Electrical and Electronics Engineering,
Thiagarajar College of Engineering,
Madurai, Tamilnadu, India
Email: sbeee@tce.edu

Abstract: In this paper, two new similarity measures, namely distance of term frequency-based similarity measure (DTFSM) and presence of common terms-based similarity measure (PCTSM), are proposed to compute the similarity between two documents for improving the effectiveness of text document clustering. The effectiveness of the proposed similarity measures is evaluated on reuters-21578 and WebKB datasets for clustering the documents using K-means and K-means++ clustering algorithms. The results obtained by using the proposed DTFSM and PCTSM are significantly better than other measures for document clustering in terms of accuracy, entropy, recall and F-measure. It is evident that the proposed similarity measures not only improve the effectiveness of the text document clustering, but also reduce the complexity of similarity measures based on the number of required operations during text document clustering.

Keywords: document clustering; similarity measures; accuracy; entropy; recall; F-measure.

Reference to this paper should be made as follows: Lakshmi, R. and Baskar, S. (2021) 'Efficient text document clustering with new similarity measures', *Int. J. Business Intelligence and Data Mining*, Vol. 18, No. 1, pp.49–72.

Biographical notes: R. Lakshmi received her BSc degree in Mathematics from Madurai Kamaraj University, Madurai, India in 1995, her MCA degree in Computer Applications from Bharathidhasan University, Trichy, India in 1998 and her ME degree in Computer Science and Engineering from Anna University, Chennai, India in 2007. She worked for five years as a Lecturer at EMG Yadava women's College and 2.5 years at Raja College of Engineering and Technology, Madurai. She is currently working as an Associate Professor in Computer Science Engineering at K.L.N. College of Engineering from 2007. Her research interests include document categorisation, machine learning, and natural language processing.

S. Baskar received his BE and PhD degrees from Madurai Kamaraj University, Madurai, India in 1991 and 2001, respectively, and his ME degree from Anna University, India in 1993. He is working as a Professor and Head in Department of Electrical and Electronics Engineering, Thiagarajar College of Engineering, Madurai. He did his post-doctoral research in Evolutionary Computation at NTU, Singapore. He is a senior member of IEEE, Fellow of Institution of Engineers (India) and life member of the Indian Society for Technical Education. He was the recipient of the Young Scientists BOYSACAST Fellowship during 2003–2004 supported by the Department of Science and Technology, Government of India. He is the reviewer for *Evolutionary Computation, Soft Computing and Engineering Optimization* journal. He has published over 80 papers in journals in the area of evolutionary optimization and applications. His research interests include the development of new evolutionary algorithms and applications to engineering.

1 Introduction

Organisation of the huge amount of text documents is a major problem in the areas of information retrieval, data mining, text mining, web search and online learning (Han and Kamber, 2006; Za'in et al., 2017). Text document clustering and classification are the most important techniques for effectively organising the text documents. The clustering algorithms (Han and Kamber, 2006; D'hont et al., 2010), that are unsupervised learning methodology, partition a document collection into several groups such the documents within the same group like one another and dissimilar to each of the other groups. Classification algorithms (Han and Kamber, 2006; Pratama et al., 2018; Lin et al., 2014; JooEr et al., 2016) are supervised learning methodology, classify new documents based on the training dataset and class labels. Several clustering and classification methods are used to organise text documents (Lin et al., 2014). This research work is focused only the K-means and K-means++ clustering algorithms along with two proposed similarity measures for organising the text documents. The quality of clustering is based on the similarity measure used in the clustering algorithm (D'hont et al., 2010; Lin et al., 2014; JooEr et al., 2016; Jiang et al., 2011; Basu and Murthy, 2015; Bouakkaz et al., 2015). The main task of text document clustering technique is to determine the similarity (Lin et al., 2014; Basu and Murthy, 2015; Bouakkaz et al., 2015) of documents for grouping. Two documents are said to be similar, if both the documents have similar terms with equal frequencies. A 'term' is the word form which is shown in the documents. A 'term frequency' is the number of occurrences of a particular term in a document. In order to increase the performance of clustering and classification of text documents, varieties of similarity measures have been proposed for computing the similarity of two documents (D'hont et al., 2010; Lin et al., 2014; JooEr et al., 2016; Jiang et al., 2011; Basu and Murthy, 2015; Bouakkaz et al., 2015).

The Euclidean distance (Lin et al., 2014; Cha, 2007), which is a default distance measure of similarity-based method, is one of the similarity measures taken from Euclidean geometry. Cosine similarity (D'hont et al., 2010; Jiang et al., 2011; Cha, 2007; Kaneko et al., 2013), which is mostly used in text processing and information retrieval, takes the measure of the cosine of the angle between two vectors. One of the similarity measures, Jaccard coefficient (Lin et al., 2014; Cha, 2007) is defined as the size of

intersection divided by the size of union of samples of datasets. The Dice's coefficient (Lin et al., 2014; Cha, 2007) is defined as two times size of the intersection divided by the size of square of samples of datasets.

D'hont et al. (2010) implemented a cosine-based pairwise adaptive similarity for text document clustering. Lin et al. (2014) proposed similarity measure for text processing (SMTP) based on three cases, the features which appear in both documents, the features appear in only one document and the features appear in none of the documents. Basu and Murthy (2015) used a threshold value for computing the distance between two documents. Bouakkaz et al. (2015) adopted Google similarity measure to calculate similarity between online documents. Luo et al. (2009) proposed the concept of neighbours and link in addition to cosine similarity for identifying the similarity between two documents. Aliguliyev (2009) has introduced a modified cosine similarity measure, which is calculated by subtracting the corresponding term average weight from each co-measured pair of documents. Yuan and Sun (2005) proposed ontology-based structured cosine similarity for text document processing.

Various similarity measures are used for document clustering and classification, however; this still is a thrust area for the researchers to propose novel similarity measures. An improper similarity measure may produce bad clusters which will directly affect the organisation of the documents. Moreover, a specific similarity measure gives better performance in some datasets and poor performance in other datasets. Thus, it is essential to propose a similarity measure for organising all types of text document datasets efficiently.

In this paper, two similarity measures (DTFSM and PCTSM) are proposed for computing similarity between two documents. Difference and scalar multiplication of individual term frequencies between two documents are considered for the proposed DTFSM. The proposed similarity measure PCTSM is derived from the Dice coefficient similarity measure. For computing the similarity of two documents, the addition of the square of the length of each document is used as the denominator of PCTSM whereas in the traditional Dice coefficient similarity measure, sum of the square of each individual term frequencies for each document used in its denominator part. The proposed similarity measures are applied in both K-means and K-means++ clustering algorithms and the results obtained are compared with other similarity measures in terms of accuracy, entropy, recall and F-measure.

The rest of this paper is organised as the following ways. The related works are discussed in Section 2. The text document clustering is outlined in Section 3. The proposed measures are introduced in Section 4. Experimental results are presented in Section 5 and conclusion part is presented in Section 6.

2 Related work

A similarity measure is a task of computing the degree of similarity between any two text documents. Let D be set of n documents with m features t_1, t_2, t_3, …, t_m. Then, every document d_i, $1 \leq i \leq n$, can be represented as $\{t_{i1}, t_{i2}, \ldots, t_{im}\}$ in a m-dimensional vector space. If $d_1 = \{t_{11}, t_{12}, t_{13}, \ldots, t_{1m}\}$ and $d_2 = \{t_{21}, t_{22}, t_{23}, \ldots, t_{2m}\}$ are two document vectors, then the popular similarity measures adopted for computing the similarity between two documents as discussed below.

2.1 Euclidean distance measure

The Euclidean distance (Lin et al., 2014; Cha, 2007) is a basic way of computing distance between two samples. It computes the distance between two samples directly, based on the magnitude of changes in the sample levels. For any two documents d_1 and d_2 with m terms, the Euclidean distance and similarity between them can be described in equations (1) and (2).

$$Euclidean(d_1, d_2) = \sqrt{\sum_{i=1}^{m} |d_{1i} - d_{2i}|^2} \tag{1}$$

$$Euclidean\ similarity = 1 - Euclidean(d_1, d_2) \tag{2}$$

When the Euclidean distance of any two documents is 0, the two documents are identical, and if 1, there is nothing common between these two documents.

2.2 Cosine similarity measure

Cosine similarity is one of the most popular similarity measures in text document processing (D'hont et al., 2010; Bouakkaz et al., 2015; Kaneko et al., 2013). For two documents d_1 and d_2, the cosine similarity measure between them is given in equation (3).

$$Cosine(d_1, d_2) = \frac{d_1 \cdot d_2}{\|d_1\| \cdot \|d_2\|} = \frac{\sum_{i=1} (d_{1i} \cdot d_{2i})}{\sqrt{\sum_{i=1}^{m} d_{1i}^2} \times \sqrt{\sum_{i=1}^{m} d_{2i}^2}} \tag{3}$$

When the cosine similarity is 1, the two documents are identical, and if 0, there is nothing common between these two documents.

2.3 Extended Jaccard coefficient

The extended Jaccard coefficient (Cha, 2007) or Jaccard similarity coefficient is a measure used for comparing the similarity and diversity of the sample set. That is, the Jaccard similarity coefficient compares the members of two sets to see which members are shared and which are distinct. The extended Jaccard coefficient has been mathematically defined in equation (4).

$$\begin{aligned} Jaccard\ similarity\ coefficient(d_1, d_2) &= \frac{d_1 \cdot d_2}{\|d_1\| + \|d_2\| - (d_1 \cdot d_2)} \\ &= \frac{\sum_{i=1}^{m} (d_{1i} \cdot d_{2i})}{\sum_{i=1}^{m} d_{1i}^2 + \sum_{i=1}^{m} d_{2i}^2 - \sum_{i=1}^{m} (d_{1i} \cdot d_{2i})} \end{aligned} \tag{4}$$

2.4 Dice's similarity coefficient

The Dice's similarity coefficient measure (Lin et al., 2014) is defined as two times intersection of the sample set divided by the sum of square of terms. The definition of Dice similarity coefficient is mathematically shown in equation (5).

$$Dice's\ similarity\ coefficient\left(d_1, d_2\right) = \frac{2 \times d_1 \cdot d_2}{\|d_1\| + \|d_2\|} = \frac{2\sum_{i=1}^{m}\left(d_{1i} \cdot d_{2i}\right)}{\sum_{i=1}^{m} d_{1i}^2 + \sum_{i=1}^{m} d_{2i}^2} \tag{5}$$

2.5 Similarity measure for text processing

The SMTP (Lin et al., 2014) considers three cases for computing similarity between two documents. The first case deals the features that appear in both documents. The second case deals the features that appear in only one document and the third case deals the features that appear in none of the documents. Mathematical descriptions of SMTP are given in equations (6) to (9).

$$SMTP\left(d_1, d_2\right) = \frac{F\left(d_1 \cdot d_2\right) + \lambda}{1 + \lambda} \tag{6}$$

where

$$F\left(d_1 \cdot d_2\right) = \frac{\sum_{i=1}^{m} N_*\left(d_{1i}, d_{2i}\right)}{\sum_{i=1}^{m} N_\cup\left(d_{1i}, d_{2i}\right)} \tag{7}$$

where

$$N_*\left(d_{1i}, d_{2i}\right) = \begin{cases} 0.5\left(1 + \exp\left\{-\left(\dfrac{d_{1i} - d_{2i}}{\sigma_j}\right)^2\right\}\right), \text{if } d_{1i}, d_{2i} > 0 \\ 0, \text{if } d_{1i}, = 0 \text{ and } d_{2i} = 0 \\ -\lambda, \text{otherwise} \end{cases} \tag{8}$$

$$N_\cup\left(d_{1i}, d_{2i}\right) = \begin{cases} 0, \text{if } d_{1i} = 0 \text{ and } d_{2i} = 0 \\ 1, \text{otherwise} \end{cases} \tag{9}$$

Among all the existing similarity measures, cosine similarity is applied in several text applications, including text document classification and clustering. Even though Euclidean distance is a default similarity measure used in K-means and K-means++ clustering algorithms, it is not suitable for text document processing (Lin et al., 2014).

3 Text document clustering

3.1 Document representation

Document representation deals with how textual documents are represented for various text processing techniques. A document is usually represented as a vector (Yuan and Sun, 2005; Al-Mubaid and Umair, 2006; Zhang et al., 2012) in which each component indicates the value of the corresponding term frequency in the individual document. Normally, the dimensionality of a document is large and the resulting vector is sparse (Dhillon and Modha, 2001; Riaz et al., 2017; Sharmili and Chilambuchelvan, 2016), i.e., most of the components in the vectors are zero. Such sparsity and high dimensionality are major challenges for the similarity measures to identify the similarity of documents (Han and Kamber, 2006; Lin et al., 2014). The most widely used method is the *term frequency inverse document frequency weighting* (*tf – idf*) method to represent the documents for text processing techniques (Za'in et al., 2017; Lin et al., 2014; Aliguliyev, 2009). The *tf – idf* method assumes that the importance of a term in a document is reflected by its frequency of appearance in documents (Za'in et al., 2017). The *term frequency* (tf_i) (Lin et al., 2014; Gong et al., 2011) in a document is the number of occurrences of the i^{th} term which is divided by the length of the document. That is, $tf_i = \frac{\sum_i f_i}{n}$, where n is the length of the document. The *inverse document frequency* (idf_i) is calculated by $1+\log\left(\frac{N}{df_i}\right)$, where N is the total number of documents and df_i is the number of documents with the term i. The calculation of $tf_i - idf_i$ for the term i in a document is $tf_i \times idf_i$.

3.2 K-means clustering algorithm

K-means is the most important flat clustering algorithm (Lin et al., 2014; Bouakkaz et al., 2015; Sharmili and Chilambuchelvan, 2016; Gong et al., 2011; Artur and Vassilvitskii, 2007; Aubaidan et al., 2014). This algorithm is used to cluster n documents into K partitions. The traditional K-means clustering algorithm is given in Algorithm 1.

Algorithm 1 The traditional k-means clustering

Input: D-document-by-term matrix, n-number of documents, K-number of clusters

Output: K clusters of given dataset

1. Randomly choose K documents from document set as initial centroids.
2. Calculate the similarity between each document and cluster centroids.
3. Assign each document of the document set to the cluster whose similarity between the document and cluster centroid is maximum of all the cluster centroids.
4. Recalculate the new cluster centroids using the mean value of each cluster.
5. Recalculate the similarity between each document and new obtained cluster centroids.
6. If no document was reassigned then stop, otherwise repeat from step 3.

3.3 K-means++ clustering algorithm

K-means++ algorithm (Artur and Vassilvitskii, 2007; Aubaidan et al., 2014) provides a way to choose initial centroids for the K-means algorithm. Let D be a set of document collection, and K be the number of specified seeds for cluster. Let $d(x)$ be the shortest distance from document x to the closest centroids. K-means++ algorithm is described in Algorithm 2.

Algorithm 2 K-means++ clustering algorithm

Phase I: choose a set of k initial centres from a document set.

Input: D-document-by-term matrix, n-number of documents, K-number of clusters

Output: K initial centroids

1. Randomly take one document $c_1 \in D$ as the first centroid.
2. Take a new centroid c_i, choosing $x \in D$ with the highest probability $\frac{d(x)^2}{\sum_{x \in D} d(x)^2}$.
3. Repeat step 2 until all the K centroids are taken.

Phase II: Standard K-means clustering algorithm

Input: K initial centroids

Output: K clusters of the given dataset

1 Taking the K centroids which is the output of the Phase I, proceed with the standard K-means algorithm for clustering.

4 Proposed similarity measures

Two similarity measures (DTFSM and PCTSM) are proposed for computing similarity of two documents d_1 and d_2 as well as improving the quality of text document clustering.

4.1 Distance of terms frequency-based similarity measure (DTFSM)

The concept of standard Euclidean distance has been used to compute similarity between documents for several decades. It computes the distance between two documents directly, based on the difference between the individual term frequencies. In general, Euclidean is not a good measure of computing the similarity in text document categorisation problems, due to the high dimensionality of the documents (Lin et al., 2014; Dhillon and Modha, 2001). In order to handle high dimensional datasets, the appropriate similarity measures, cosine similarity, Jaccard coefficient similarity, Dice's coefficient similarity, etc., are used. All the above said similarity measures are different variations of normalised scalar product of two documents. Mostly, the length of term frequency is used to normalise the scalar product.

The proposed DTFSM considers the normalised difference between the individual term frequencies. This enhances the effectiveness of the proposed method. The two time scalar product of two documents is used to normalise the difference between the individual term frequencies. That is, the proposed DTFSM is embedded with the difference between the individual term frequency and scalar multiplication of the two

documents. In the proposed DTFSM, the sum of difference of individual term frequency between two documents for m number of terms is calculated and it is divided by two time scalar multiplication of two documents' term frequency. Finally, the square root of whole value is subtracted from 1. The proposed DTFSM is mathematically expressed in equation (10).

$$DTFSM\left(d_1, d_2\right) = 1 - \sqrt{\frac{\sum_{i=1}^{m} \left|d_{1i} - d_{2i}\right|}{2\sum_{i=1}^{m} \left|d_{1i} \cdot d_{2i}\right|}} \tag{10}$$

4.2 Presence of common terms-based similarity measure

Presence of common terms-based similarity measure (PCTSM) is derived from traditional Dice coefficient. For computing the similarity of two documents, the addition of the square of the length of each document is used as the denominator of PCTSM whereas in the traditional Dice coefficient similarity measure, sum of the square of each individual term frequencies for each document used in its denominator part. The idea used in the proposed PCTSM increases the effectiveness of the clustering and at the same time reduces the time complexity based on the number of operations used compared to the traditional Dice coefficient similarity measure.

The PCTSM is defined as two time scalar multiplication of two documents' term frequency divided by the addition of square of length of each document. The proposed PCTSM is specified in mathematical form as shown in equation (11).

$$PCTSM\left(d_1, d_2\right) = \frac{2\sum_{i=1}^{m} \left(d_{1i} \cdot d_{2i}\right)}{\left(\sum_{i=1}^{m} d_{1i}\right)^2 + \left(\sum_{i=1}^{m} d_{2i}\right)^2} \tag{11}$$

4.3 Computational complexity of DTFSM and PCTSM

The computational complexity of any similarity or dissimilarity measure is expressed as the number of required operations (Li et al., 2008). Let d_1 and d_2 be two documents with m terms. Then, the comparison of computational complexity of DTFSM and PCTSM with other standard similarity measures for calculating similarity between the two documents is described in Table 1.

Table 1 Comparison of computational complexity DTFSM and PCTSM with other similarity measures

Similarity measures	*Computational complexity*
Cosine	$6m + 1$
Jaccard coefficient	$6m$
Dice's similarity coefficient	$6m$
DTFSM	$4m + 2$
PCTSM	$4m + 4$

SMTP (Lin et al., 2014) consists of several operations, including calculation of standard deviation for all terms to perform similarity in between two documents. All the similarity measures described in Table 1 takes $O(m)$ time complexity. But, the proposed similarity measures (DTFSM and PCTSM) need less time complexity based on the number of operations needed to perform similarity between two documents.

5 Experimental results

In this section, the effectiveness of the proposed similarity measures (DTFSM and PCTSM) with K-means and K-means++ clustering algorithms is analysed. The performance of DTFSM and PCTSM are compared with other four similarity measures namely, cosine, extended Jaccard, Dice's similarity coefficient and SMTP.

5.1 Datasets

5.1.1 Reuter-21578

Reuters-21578 document collection (http://www.daviddlewis.com/resources/testcollections/reuters21578/) is used to evaluate the effectiveness of cluster performance. The reuters-21578 ModeApt'*e* split text categorisation test collection contains thousands of documents collected from Reuters newswire in 1987. Majority researchers have used the version of reuters-21578 ModeApt'*e*, which contains 90 categories and 12,902 documents. In the proposed work, eight most frequent categories among the 90 categories are used for the work and all the documents with less than or more than one topic are removed. The resulting dataset is named Reuters-8 in which 5,485 of the documents are pre-destinated for training and the other 2,189 documents are pre-destinated for testing. Table 2 shows the distribution of the documents in each class for training and testing. The dataset can be obtained from http://www.cs.umb.edu/~smimarog/textmining/datasets/. There are 17,745 features involved in the dataset.

Table 2 Distribution of documents in each class of Reuters-8

Class	*Number of training data*	*Number of testing data*	*Subtotal of data*
Acq	1,596	696	2,292
Crude	253	121	374
Earn	2,840	1,083	3,923
Grain	41	10	51
Interest	190	81	271
Money-fix	206	87	293
Ship	108	36	144
Trade	251	75	326
Total	5,485	2,189	7,674

5.1.2 WebKB

The documents in the WebKB dataset are the web pages collected by the world wide knowledge base project of the CMU text learning group and the documents in the dataset are manually classified into different classes. This dataset can be obtained from http://www.cs.umb.edu/~smimarog/textmining/datasets/. The documents in this dataset are not classified as training or testing datasets. They are divided randomly into training and testing datasets. Table 3 shows the distribution of the documents in each class randomly selected for training and testing, respectively. The number of features involved in this dataset is 7,786.

Table 3 Distribution of documents in each class of WebKB

Class	*Number of training data*	*Number of testing data*	*Subtotal of data*
Project	336	168	504
Course	620	310	930
Faculty	750	374	1,124
Student	1,097	544	1,641
Total	2,803	1,396	4,199

5.2 Evaluation of performance

The proposed similarity measures are applied in both K-means and K-means++ clustering algorithms and their performances are compared with other similarity measures described in Section 2 based on the measures (Chim and Deng, 2008) such as accuracy, recall, entropy and F-measure. If the accuracy, recall and F-measure of cluster are 1 and the entropy is 0, then the cluster is considered as a very efficient cluster.

- *Accuracy* – the accuracy (Lin et al., 2014) for clustering of text documents is mathematically defined in equation (12).

$$Accuracy = \frac{\sum_{i=1}^{K} most_i}{n} \tag{12}$$

  where $most_i$ is the majority number of documents with identical class labels in the i^{th} cluster, n is the total number of documents and K is the number of clusters.

- *Entropy* – the entropy (Lin et al., 2014; Chim and Deng, 2008) for clustering of text documents is mathematically defined in equation (13).

$$Entropy = \frac{\sum_{i=1}^{K} n_i \left(\sum_{j=1}^{p} -\frac{n_i^j}{n_i} \log \frac{n_i^j}{n_i} \right)}{(\log p) n} \tag{13}$$

  where n_i is the number of documents in the i^{th} cluster, and n_i^j is the number of documents with label j in the i^{th} cluster.

- *F-measure* – in data mining and information retrieval, recall is a measure of how many truly relevant results are returned, while precision is a measure of result relevancy. That is, precision is the ratio of the number of relevant documents to the total number of documents retrieved.
- *Recall* – recall is the ratio of the number of relevant documents retrieved to the total number of relevant documents in the entire collection. The precision and recall (Lin et al., 2014; Chim and Deng, 2008) can be defined as $Precision\ P(i,\ j) = \frac{n_{ij}}{n_j}$ and $Recall\ R(i,\ j) = \frac{n_{ij}}{n_i}$, where n_i is the number of members in class i, n_j is the number of members in the cluster j and n_{ij} is the number of members in class i of the cluster j. F-measure is the harmonic mean of precision and recall (Lin et al., 2014; Zhang et al., 2012; Li et al., 2008) and it is mathematically defined in equation (14).

$$F\text{-}measure(i,\ j) = F(i,\ j) = \frac{2 \times P(i,\ j) \times R(i,\ j)}{P(i,\ j) + R(i,\ j)} \tag{14}$$

- The final F-measure for the entire clusters is calculated by using the equation (15).

$$F\text{-}measure = \sum_i \frac{n_i}{n} \times \max\{F(i,\ j)\} \tag{15}$$

where n is the number of documents.

5.3 Results and discussion

The proposed DTFSM and PCTSM are applied in both K-means and K-means++ clustering algorithms for Reuters-21578 and WebKB datasets. All the results are taken by implementing separate programs for each similarity measure in MATLAB 7.10. The K-means and K-means++ algorithms runs under different K values (8, 16, 24 and 32) for Reuters-8 and with K values (4, 8, 12, and 16) for WebKB to measuring the accuracy (Ac), entropy (En), recall (Re) and F-measure (FM). The results shown from Tables 4 to 7 are the average of ten independent runs by K-means and K-means++ clustering algorithms with respective similarity measures.

5.3.1 Result of DTFSM for text document clustering

The result of DTFSM on testing data of Reuters-8 and WebKB is shown in Tables 4 and 5 and the same are represented from Figures 1 to 8.

In all the cases of K-means and K-means++, the proposed DTFSM provides better accuracy, recall and F-measure values compared to cosine, extended Jaccard coefficient, Dice's coefficient and SMTP similarity measures on both Reuters-8 and WebKB. In K-means, the proposed similarity measure DTFSM reduces the entropy in all the cases except for K = 32 on Reuters-8 and K = 16 on WebKB. Only in the above two cases (K = 32 on Reuters-8 and K = 16 on WebKB), the proposed DTFSM is in the second position based on the performance. Similarly, the proposed DTFSM provides better results on K-means++.

Table 4 Accuracy and recall values by K-means and K-means++ with different measures on testing data of Reuters-8 and WebKB

Algorithm	*Dataset*	*K*	*Cosine*		*Jaccard*		*Dice*		*SMTP*		*DTFSM*	
			Ac	*Re*	*Ac*	*Re*	*Ac*	*Re*	*Ac*	*Re*	*Ac*	*Re*
K-means	Reuters-8	8	0.8580	0.4699	0.8478	0.4548	0.8516	0.4697	0.7978	0.3856	*0.8615*	*0.5222*
		16	0.8761	0.2958	0.8827	0.3189	0.8691	0.2957	0.8325	0.2136	*0.8894*	*0.3216*
		24	0.8834	0.2170	0.8842	0.2330	0.8890	0.2307	0.8503	0.1666	*0.8922*	*0.2334*
		32	0.8919	*0.1957*	0.8849	0.1716	0.8875	0.1749	0.8602	0.1268	*0.8950*	0.1828
	WebKB	4	0.6491	0.5604	0.6494	0.5718	0.6431	0.5655	0.5334	0.4466	*0.6814*	*0.6036*
		8	0.6508	0.2952	0.6433	0.2865	0.6393	0.2808	0.5972	0.2595	*0.6711*	*0.2996*
		12	0.6273	0.1768	0.5701	0.1934	0.6465	*0.2409*	0.6288	0.1880	*0.6686*	0.2052
		16	0.6393	0.1415	0.6329	0.1409	0.6388	0.1408	0.6467	0.1428	*0.6498*	*0.1477*
K-means++	Reuters-8	8	0.8727	0.5302	0.8588	0.4993	0.8626	0.5045	0.8228	0.4334	*0.8751*	*0.5584*
		16	0.9013	0.3382	0.8991	0.3326	0.8814	0.3166	0.8538	0.2214	*0.9031*	*0.3402*
		24	0.8985	0.2323	0.8851	0.2486	0.8992	*0.2491*	0.8675	0.1924	*0.9063*	0.2444
		32	0.8936	0.1908	0.9014	0.1825	0.8971	0.1874	0.8835	0.1437	*0.9018*	*0.1945*
	WebKB	4	0.6650	0.5877	0.6802	0.6109	0.6771	0.6066	0.6046	0.6267	*0.7152*	*0.6795*
		8	0.6558	0.2792	0.6673	0.2961	0.6728	0.3024	0.6511	0.2953	*0.7003*	*0.3312*
		12	0.6582	0.1919	0.6791	0.1983	0.6649	0.1988	0.6640	0.1930	*0.7003*	*0.2130*
		16	0.6417	0.1412	0.6417	0.1419	0.6592	0.1488	0.6769	0.1558	*0.6903*	*0.1570*

Table 5 Entropy and F-measure values by K-means and K-means++ with different measures on testing data of Reuters-8 and WebKB

Algorithm	*Dataset*	*K*	*Cosine*		*Jaccard*		*Dice*		*SMTP*		*DTFSM*	
			En	*FM*	*En*	*FM*	*En*	*FM*	*En*	*FM*	*En*	*FM*
K-means	Reuters-8	8	0.1851	0.5896	0.1900	0.5789	0.1853	0.5586	0.3346	0.5305	*0.1767*	*0.6126*
		16	0.1577	0.4303	0.1553	0.4269	0.1636	0.4294	0.2673	0.3921	*0.1463*	*0.4790*
		24	0.1473	0.3470	0.1544	0.3590	0.1482	0.3724	0.2383	0.2978	*0.1438*	*0.3826*
		32	*0.1376*	0.3009	0.1533	0.3025	0.1474	0.3045	0.2088	0.2453	0.1391	*0.3302*
	WebKB	4	0.6161	0.6138	0.6235	0.613	0.627	0.5994	0.7842	0.4996	*0.5877*	*0.6420*
		8	0.6022	*0.4830*	0.6207	0.4254	0.6246	0.4166	0.7187	0.4207	*0.5887*	0.4500
		12	0.6420	0.3470	0.6251	0.3411	0.6189	0.3448	0.6658	0.3461	*0.5977*	*0.3499*
		16	0.6277	0.3039	0.6286	0.2769	0.6304	0.2684	0.6265	*0.3081*	*0.6134*	0.2779
K-means++	Reuters-8	8	0.1695	0.5402	0.1849	0.5139	0.1801	0.5166	0.2975	0.4603	*0.1663*	*0.5569*
		16	0.1375	0.3885	0.1398	0.3831	0.1601	0.3607	0.2474	0.3038	*0.1329*	*0.3954*
		24	0.1400	0.2838	0.1510	0.2883	0.1337	*0.2952*	0.2257	0.2677	*0.1329*	0.2938
		32	0.1417	0.2344	0.1347	0.2347	0.1406	0.2339	0.1854	0.2043	*0.1346*	*0.2434*
	WebKB	4	0.5902	0.5799	0.5886	0.6257	0.5906	0.6237	0.6859	0.5881	*0.5316*	*0.6752*
		8	0.5959	0.3434	0.6097	0.3845	0.5806	0.3850	0.7670	0.3678	*0.5559*	*0.4167*
		12	0.6017	0.2587	0.5806	0.2838	0.5979	0.2892	0.6465	0.2833	*0.5576*	*0.3031*
		16	0.6205	0.2057	0.6272	0.2200	0.6048	0.2266	0.6167	0.2242	*0.5667*	*0.2415*

Figure 1 Accuracy of DTFSM on testing data Reuters-8 (see online version for colours)

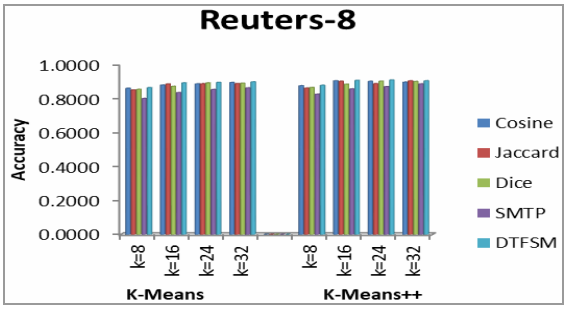


Figure 2 Recall of DTFSM on testing data Reuters-8 (see online version for colours)

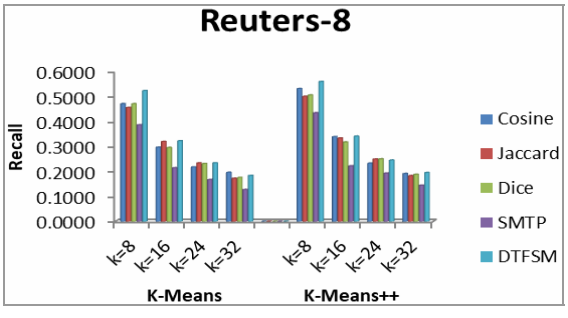


Figure 3 Entropy of DTFSM on testing data Reuters-8 (see online version for colours)

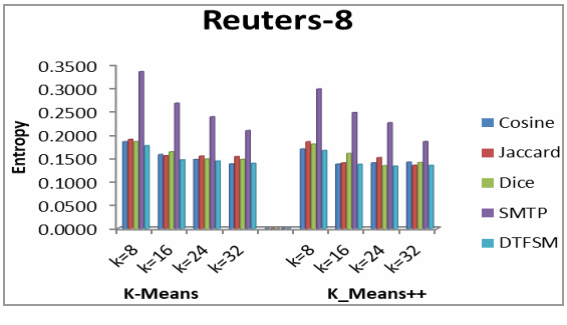


Figure 4 F-measure of DTFSM on testing data Reuters-8 (see online version for colours)

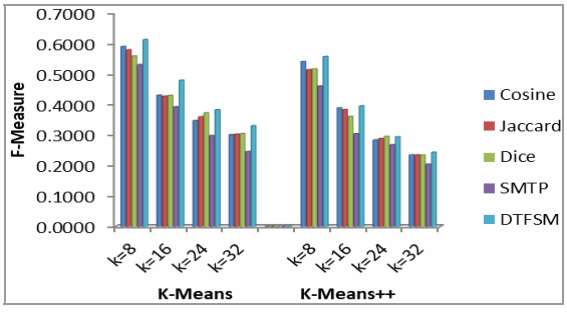


Figure 5 Accuracy of DTFSM on testing data WebKB (see online version for colours)

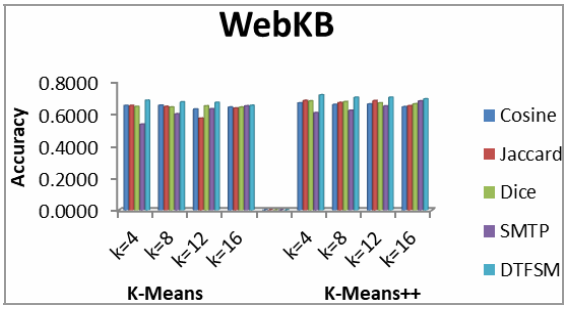


Figure 6 Recall of DTFSM on testing data WebKB (see online version for colours)

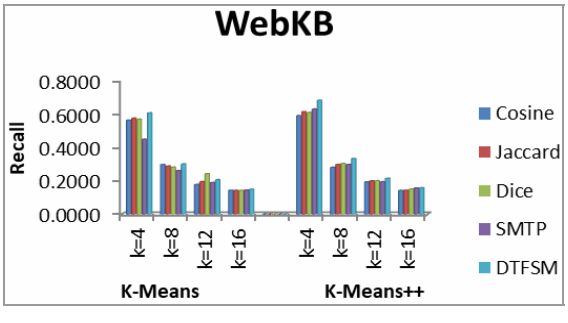


Figure 7 Entropy of DTFSM on testing data WebKB (see online version for colours)

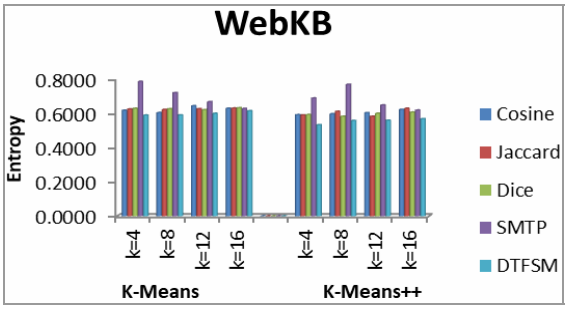


Figure 8 F-measure of DTFSM on testing data WebKB (see online version for colours)

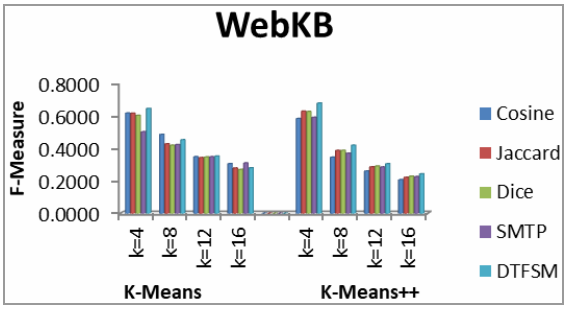


Table 6 Accuracy and recall values by K-means and K-means++ with different measures on testing data of Reuters-8 and WebKB

Algorithm	*Dataset*	*K*	*Cosine*		*Jaccard*		*Dice*		*SMTP*		*PCTSM*	
			Ac	*Re*	*Ac*	*Re*	*Ac*	*Re*	*Ac*	*Re*	*Ac*	*Re*
K-means	Reuters-8	8	0.8580	0.4699	0.8478	0.4548	0.8516	0.4697	0.7978	0.3856	*0.8633*	*0.5396*
		16	0.8761	0.2958	0.8827	0.3189	0.8691	0.2957	0.8325	0.2136	*0.8968*	*0.3452*
		24	0.8834	0.2170	0.8842	0.2330	0.8890	0.2307	0.8503	0.1666	*0.8974*	*0.2359*
		32	0.8919	*0.1957*	0.8849	0.1716	0.8875	0.1749	0.8602	0.1268	*0.9078*	0.1911
	WebKB	4	0.6491	0.5604	0.6494	0.5718	0.6431	0.5655	0.5334	0.4466	*0.6683*	*0.5902*
		8	0.6508	0.2952	0.6433	0.2865	0.6393	0.2808	0.5972	0.2595	*0.6991*	*0.3241*
		12	0.6273	0.1768	0.5701	0.1934	0.6465	*0.2409*	0.6288	0.1880	*0.6719*	0.2013
		16	0.6393	0.1415	0.6329	0.1409	0.6388	0.1408	0.6467	0.1428	*0.6682*	*0.1464*
K-means++	Reuters-8	8	*0.8727*	0.5302	0.8588	0.4993	0.8626	0.5045	0.8228	0.4334	0.8706	*0.5793*
		16	0.9013	0.3382	0.8991	0.3326	0.8814	0.3166	0.8538	0.2214	*0.9051*	*0.3439*
		24	0.8985	0.2323	0.8851	0.2486	0.8992	*0.2491*	0.8675	0.1924	*0.9037*	0.2447
		32	0.8936	0.1908	0.9014	0.1825	0.8971	0.1874	0.8835	0.1437	*0.9072*	*0.1937*
	WebKB	4	0.6650	0.5877	0.6802	0.6109	0.6771	0.6066	0.6046	0.6267	*0.6990*	*0.6292*
		8	0.6558	0.2792	0.6673	0.2961	0.6728	0.3024	0.6511	0.2953	*0.6837*	*0.3130*
		12	0.6582	0.1919	0.6791	0.1983	0.6649	0.1988	0.6640	0.1930	*0.6864*	*0.2094*
		16	0.6417	0.1412	0.6417	0.1419	0.6592	0.1488	*0.6769*	*0.1558*	0.6609	0.1480

Table 7 Entropy and F-measure values by K-means and K-means++ with different measures on testing data of Reuters-8 and WebKB

Algorithm	*Dataset*	*K*	*Cosine*		*Jaccard*		*Dice*		*SMTP*		*PCTSM*	
			En	*FM*	*En*	*FM*	*En*	*FM*	*En*	*FM*	*En*	*FM*
K-means	Reuters-8	8	0.1851	0.5896	0.1900	0.5789	0.1853	0.5586	0.3346	0.5305	*0.1752*	*0.6636*
		16	0.1577	0.4303	0.1553	0.4269	0.1636	0.4294	0.2673	0.3921	*0.1412*	*0.5088*
		24	0.1473	0.3470	0.1544	0.3590	0.1482	0.3724	0.2383	0.2978	*0.1422*	*0.4185*
		32	0.1376	0.3009	0.1533	0.3025	0.1474	0.3045	0.2088	0.2453	*0.1306*	*0.3624*
	WebKB	4	0.6161	0.6138	0.6235	0.613	0.627	0.5994	0.7842	0.4996	*0.5900*	*0.6428*
		8	0.6022	0.4830	0.6207	0.4254	0.6246	0.4166	0.7187	0.4207	*0.5545*	*0.4839*
		12	0.6420	0.3470	0.6251	0.3411	0.6189	0.3448	0.6658	0.3461	*0.5912*	*0.3528*
		16	0.6277	0.3039	0.6286	0.2769	0.6304	0.2684	0.6265	*0.3081*	*0.5942*	0.2897
K-means++	Reuters-8	8	0.1695	0.5402	0.1849	0.5139	0.1801	0.5166	0.2975	0.4603	*0.1639*	*0.5696*
		16	*0.1375*	0.3885	0.1398	0.3831	0.1601	0.3607	0.2474	0.3038	0.1382	*0.3892*
		24	0.1400	0.2838	0.1510	0.2883	*0.1337*	0.2952	0.2257	0.2677	0.1352	*0.2956*
		32	0.1417	0.2344	0.1347	0.2347	0.1406	0.2339	0.1854	0.2043	*0.1324*	*0.2377*
	WebKB	4	0.5902	0.5799	0.5886	0.6257	0.5906	0.6237	0.6859	0.5881	*0.5900*	*0.6413*
		8	0.5959	0.3434	0.6097	0.3845	0.5806	0.3850	0.7670	0.3678	*0.5682*	*0.3970*
		12	0.6017	0.2587	0.5806	0.2838	0.5979	0.2892	0.6465	0.2833	*0.5716*	*0.2982*
		16	0.6205	0.2057	0.6272	0.2200	0.6048	0.2266	0.6167	0.2242	*0.6030*	*0.2268*

Figure 9 Accuracy of PCTSM on testing data Reuters-8 (see online version for colours)

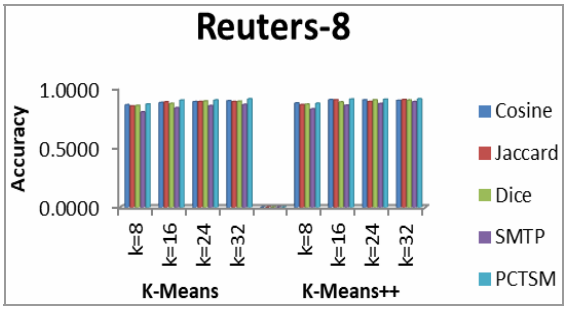


Figure 10 Recall of PCTSM on testing data Reuters-8 (see online version for colours)

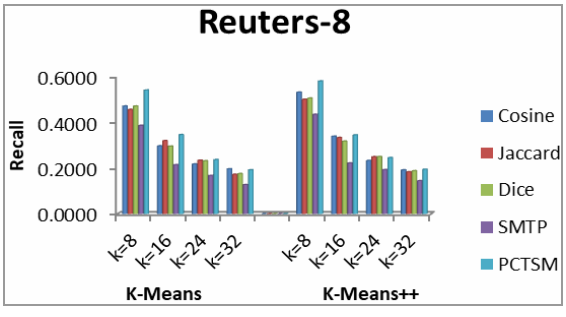


Figure 11 Entropy of PCTSM on testing data Reuters-8 (see online version for colours)

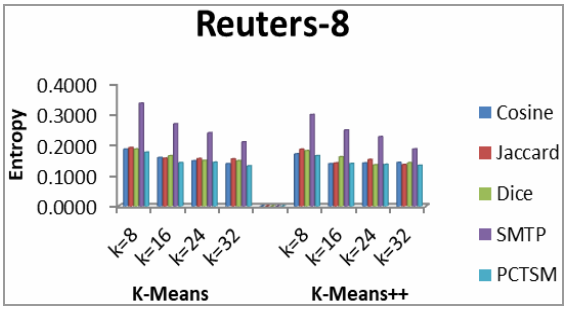


Figure 12 F-measure of PCTSM on testing data Reuters-8 (see online version for colours)

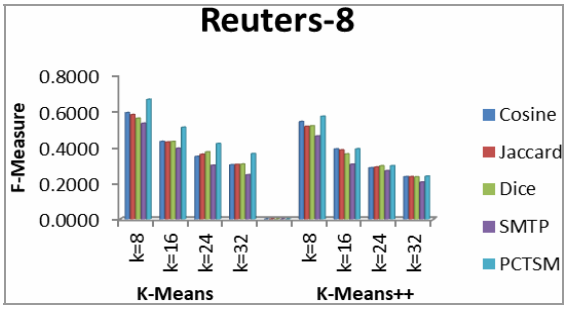


Figure 13 Accuracy of PCTSM on testing data WebKB (see online version for colours)

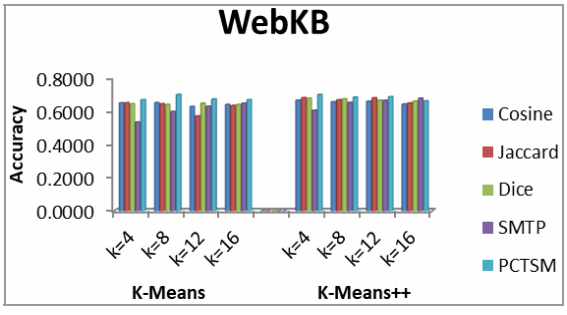


Figure 14 Recall of PCTSM on testing data WebKB (see online version for colours)

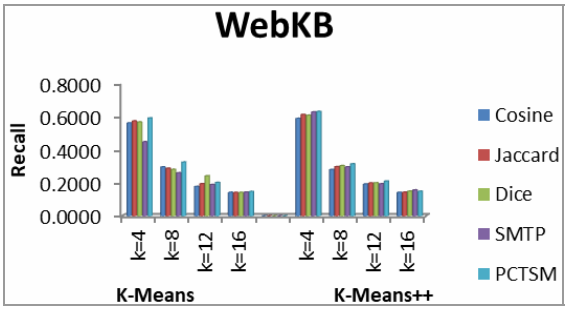


Figure 15 Entropy of PCTSM on testing data WebKB (see online version for colours)

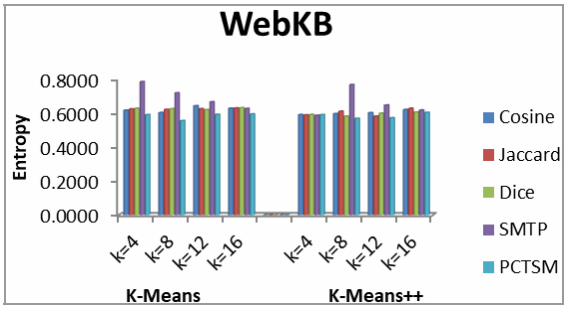
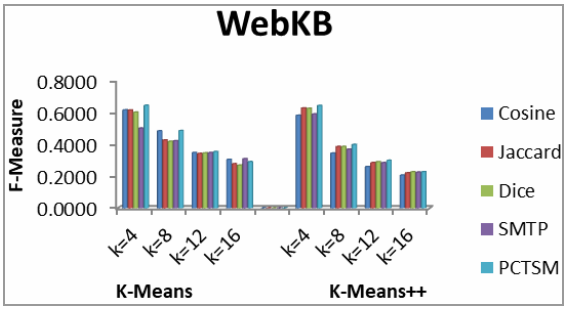


Figure 16 F-measure of PCTSM on testing data WebKB (see online version for colours)



5.3.2 Result for PCTSM for text document clustering

In this section, the results and the analysis of the proposed PCTSM are discussed. The result of PCTSM on the dataset of Reuters-8 and WebKB is shown in Tables 6 and 7 and the same are depicted from Figures 9 to 16.

Tables 6 and 7 and Figures 9 to 16, the clustering performance of PCTSM for Reuter-8 and WebKB is significant better than others in both K-means and K-means++ clustering algorithms. It is not only better than the traditional Dice's coefficient similarity, but also better than all other measures. The two proposed methods show better performance on two different datasets in terms of accuracy, recall, entropy and F-measure. The time complexity is also reduced in the proposed methods. Thus, the proposed measures can be used for similarity-based clustering algorithm for different datasets with reduced computation time.

5.3.3 Comparison of proposed similarity measures DTFSM and PCTSM

Figures 17 to 20 show the results of DTFSM and PCTSM for the testing data of Reuters-8 and Figures 21 to 23 show the results of DTFSM and PCTSM for the testing data of WebKB.

Figure 17 Comparison of accuracy between DTFSM and PCTSM on Reuters-8 (see online version for colours)

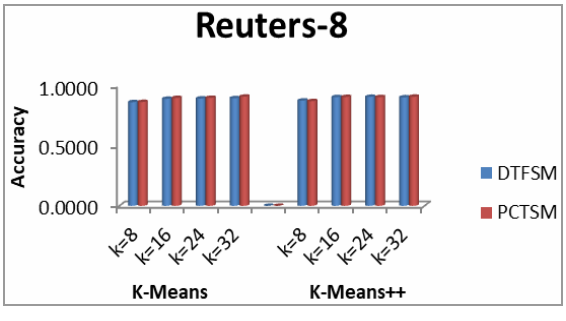


Figure 18 Comparison of recall between DTFSM and PCTSM on Reuters-8 (see online version for colours)

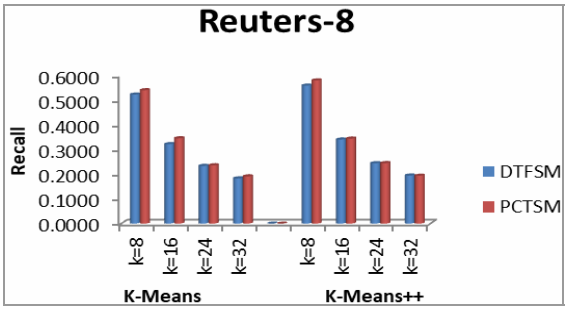


Figure 19 Comparison of entropy between DTFSM and PCTSM on Reuters-8 (see online version for colours)

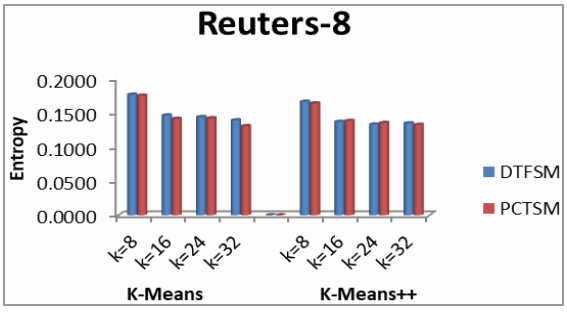


Figure 20 Comparison of F-measure between DTFSM and PCTSM on Reuters-8 (see online version for colours)

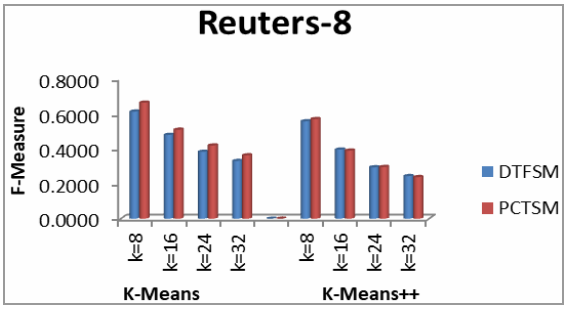


Figure 21 Comparison of accuracy between DTFSM and PCTSM on WebKB (see online version for colours)

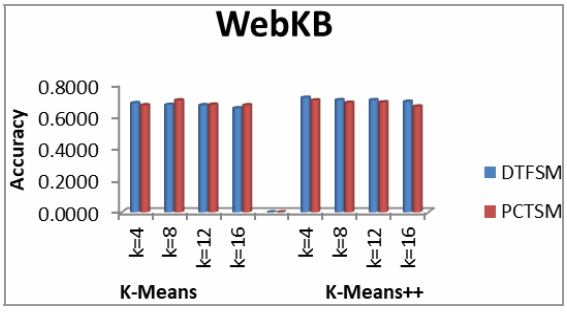


Figure 22 Comparison of recall between DTFSM and PCTSM on WebKB (see online version for colours)

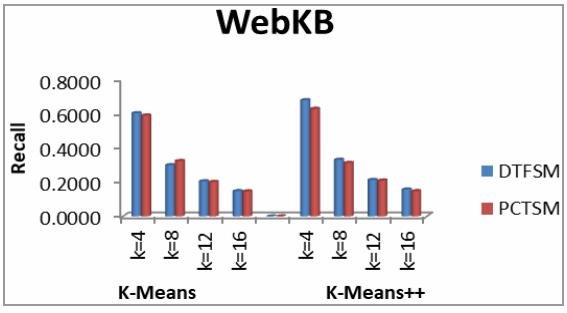


Figure 23 Comparison of entropy between DTFSM and PCTSM on WebKB (see online version for colours)

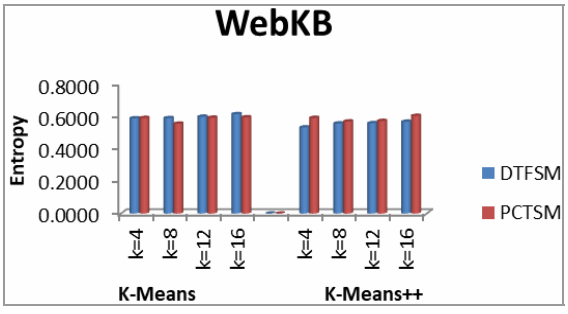
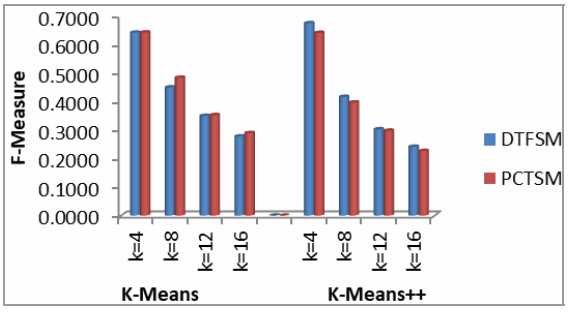


Figure 24 Comparison of F-measure between DTFSM and PCTSM on WebKB (see online version for colours)



In Reuters-8 dataset, compared to DTFSM, PCTSM performs only marginally better in terms of all the three performance measures. But, the computation overhead is significantly smaller for DTFSM. In WebKB dataset, PCTSM performs better in terms of all the four performance measures for most of the cases. In terms of time complexity, DTFSM takes a slightly less number of operations than the PCTSM. Thus, the performance of DTFSM is almost significantly similar to PCTSM with less computational overhead.

6 Conclusions

Two new similarity measures, namely DTFSM and presence of common terms similarity measure (PCTSM), between the two documents are presented. DTFSM is the extended form of Euclidean similarity measure and PCTSM is derived from Dice's coefficient similarity measure. The proposed similarity measures are applied in both K-means and K-means++ clustering algorithms for the testing datasets of Reuters-8 and WebKB. The performances in the terms of accuracy, entropy, recall and F-measure are calculated with K values (8, 16, 24 and 32) for Reuters-8 and with K values (4, 8, 12, and 16) for WebKB. The proposed similarity measures increase the performance of text document clustering for any type of text document datasets as well as decrease the computational complexity of computing the similarity between documents compared with other similarity measures including SMTP (Lin et al., 2014). When the performances of two proposed similarity measures are compared in terms of accuracy, entropy, recall and F-measure, DTFSM is slightly better than PCTSM. The performance of DTFSM is almost equally efficient as PCTSM and is slightly better than PCTSM in the terms of time complexity. The proposed similarity measures are not only suitable for K-means and K-means++ clustering algorithms, but also for any similarity-based clustering and classification algorithms. Hence, they can be used for clustering and classification of all types of text document datasets.

Acknowledgements

We thank both the Management of K.L.N. College of Engineering and Thiagarajar College of Engineering for their great supporting to complete this research work.

References

Aliguliyev, R.M. (2009) 'Clustering of document collection – a weighting approach', *Expert Syst. Appl.*, Vol. 36, No. 4, pp.7904–7916.

Al-Mubaid, H. and Umair, S.A. (2006) 'A new text categorization technique using distributional clustering and learning logic', *IEEE Trans. Knowl.Data Eng.*, Vol. 18, No. 9, pp.1156–1165.

Artur, D. and Vassilvitskii, S. (2007) 'K-means++: the advantages of careful seeding', *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp.1027–1035.

Aubaidan, B., Mohd, M. and Albared, M. (2014) 'Comparative study of k-means and k-means++ clustering algorithms on crime domain', *Journal of Computer Science*, Vol. 10, No. 7, pp.1197–1206.

Basu, T. and Murthy, C.A. (2015) 'A similarity assessment technique for effective grouping of documents', *Inf. Sci.*, Vol. 311. No. 8, pp.149–162.

Bouakkaz, M., Loudcher, S. and Ouinten, Y. (2015) 'OLAP textual aggregation approach using the Google similarity distance', *International Journal of Business Intelligence and Data Mining*, Vol. 11, No. 1, pp.31–48.

Cha, S-H. (2007) 'Comprehensive survey on distance/similarity measures between probability density functions', *Int. J. Math. Models Methods Appl. Sci.*, Vol. 1, No. 4, pp.300–307.

Chim, H. and Deng, X. (2008) 'Efficient phrase-based document similarity for clustering', *IEEE Trans. Knowl. Data Eng.*, Vol. 20, No. 9, pp.1217–1229.

D'hont, J. et al. (2010) 'Pairwise – adaptive dissimilarity measure for document clustering', *Inf. Sci.*, Vol. 180, No. 12, pp.2341–2358.

Dhillon, I.S. and Modha, D.S. (2001) 'Concept decompositions for large sparse test data using clustering', *Mach. Learn.*, Vol. 42, No. 1, pp.143–175.

Gong, L., Zeng, J. and Zhang, S. (2011) 'Text stream clustering algorithm based on adaptive feature selection', *Expert Syst. Appl.*, Vol. 38, No. 3, pp.1393–1399.

Han, J. and Kamber, M. (2006) *Data Mining: Concepts and Techniques*, 2nd ed., Morgan Kaufmann, San Francisco, CA, USA.

Jiang, J.Y. et al. (2011) 'A similarity measure for text processing', in *Int. Conf. Mach. Learn, Cybernetics*, Guilin, China, 10–13 July, pp.1460–1465.

JooEr, M., Yong, Z., Ning, W. and Pratama, M. (2016) 'Attention pooling-based convolutional neural network for sentence modeling', *Information Science*, Vol. 373, No. 12, pp.388–403.

Kaneko, M., Okamoto, S., Kohana, M. and Inayoshi, Y. (2013) 'Document clustering based on web search hit counts', *International Journal of Business Intelligence and Data Mining*, Vol. 8, No. 1, pp.61–73.

Li, Y., Chung, S.M. and Holt, J.D. (2008) 'Text document clustering based on frequent word meaning sequences', *Data Knowl. Eng.*, Vol. 64, No. 1, pp.381–404.

Lin, Y.S., Jiang, J.Y. and Lee, S.J. (2014) 'A similarity measure for text classification and clustering', *IEEE Trans. Knowl. Data Eng.*, Vol. 26, No. 7, pp.1575–1590.

Luo, C., Li, Y. and Chung, S.M. (2009) 'Text document clustering based on neighbors', *Data Knowl. Eng.*, Vol. 68, No. 11, pp.1271–1288.

Pratama, M., Angelov, P.P., Lughofer, E. and Er, M.J. (2018) 'Parsimonious random vector functional link network for data streams', *Information Sciences*, Vols. 430–431, No. 3, pp.519–537.

Riaz, S., Fatima, M., Kamran, M. and Wasif Nisar, M. (2017) 'Opinion mining on large scale data using sentiment analysis and k-means clustering', *Cluster Comput.*, pp.1–16.

Sharmili, K.C. and Chilambuchelvan, A.G. (2016) 'Optimal feature subset selection in high dimensional data clustering', *International Journal of Business Intelligence and Data Mining*, Vol. 11, No. 3, pp.242–263.

Yuan, S-T. and Sun, J. (2005) 'Ontology-based structured cosine similarity in document summarization: with applications to mobile audio-based knowledge management', *IEEE Trans. Syst. Man Cybern. Part B: Cybern.*, Vol. 35, No. 5, pp.1028–1040.

Za'in, C., Pratama, M., Lughofer, E. and Anavatti, S.G. (2017) 'Evolving type-2 web news mining', *Applied Soft Computing*, Vol. 54, pp.200–220.

Zhang, T. et al. (2012) 'Document clustering in correlation similarity measure space', *IEEE Trans. Knowl. Data Eng.*, Vol. 24, No. 6, pp.1002–1013.

Websites

http://www.cs.umb.edu/~smimarog/textmining/datasets/ (accessed 15 July 2017).

http://www.daviddlewis.com/resources/testcollections/reuters21578/ (accessed 10 July 2017).