Decision Analytics
a SpringerOpen Journal

RESEARCH Open Access

Correlated concept based dynamic document clustering algorithms for newsgroups and scientific literature

Jayaraj Jayabharathy[1*] and Selvadurai Kanmani[2]

* Correspondence:
bharathyraja@pec.edu
[1]Department of Computer Science & Engineering, Pondicherry Engineering College, Puducherry 605014 India
Full list of author information is available at the end of the article

Abstract

Increase in the number of documents in the corpuses like News groups, government organizations, internet and digital libraries, have led to greater complexity in categorizing and retrieving them. Incorporating semantic features will improve the accuracy of retrieving documents through the method of clustering and which will also pave the way to organize and retrieve the documents more efficiently, from the large available corpuses. Even though clustering based on semantics enhances the quality of clusters, scalability of the system still remains complicated. In this paper, three dynamic document clustering algorithms, namely: Term frequency based MAximum Resemblance Document Clustering (TMARDC), Correlated Concept based MAximum Resemblance Document Clustering (CCMARDC) and Correlated Concept based Fast Incremental Clustering Algorithm (CCFICA) are proposed. From the above three proposed algorithms the TMARDC algorithm is based on term frequency, whereas, the CCMARDC and CCFICA are based on Correlated terms (Terms and their Related terms) concept extraction algorithm. The proposed algorithms were compared with the existing static and dynamic document clustering algorithms by conducting experimental analysis on the dataset chosen from 20Newsgroups and scientific literature. F-measure and Purity have been considered as metrics for evaluating the performance of the algorithms. The experimental results demonstrate that the proposed algorithm exhibit better performance, compared to the four existing algorithms for document clustering.

Keywords: Static and dynamic document clustering; MAximum resemblance data labeling (MARDL) technique; Term frequency; Inverse document frequency (TFIDF); Concepts; Semantic similarity

Background

Tremendous growth in the volume of text documents available from various sources like the Internet, digital libraries, news sources, and company-wide intranets has led to an increased interest in developing methods that can help users to effectively navigate, summarize, and organize information, with an ultimate goal of helping the users to find what they are looking for. In this context, fast and high-quality document clustering algorithms play an important role, as they have shown to provide both an intuitive navigation/browsing mechanism, by organizing large amounts of information into a small number of meaningful clusters, as well as to greatly improve the retrieval performance either by cluster-driven dimensionality reduction, term-weighting Tang et al. (2005), or by query expansion Sammut and Webb (2010). As today's search engine does just string matching, documents retrieved may not be so relevant to the user's query. Thus, a good document clustering approach if available and implemented will assist in organizing the document corpus automatically into a meaningful cluster hierarchy for efficient browsing and navigation. Further, it will also help to overcome the inherent deficiencies associated with traditional information retrieval methods.

Document clustering has been investigated for use in a number of different areas of text mining and information retrieval. Initially, document clustering was investigated for improving the precision or recall in information retrieval systems and as an efficient way of finding the nearest neighbors of a document Van Rijsbergen (1989 and Kowalski and Maybury 2002, Buckley and Lewit 1985). Then clustering was used in browsing a collection of documents or in organizing the results returned by a search engine in response to a user's query Cutting et al. (1992; Zamir et al. 1997). Document clustering was also been used to automatically generate hierarchical clusters of documents Steinbach et al. (2000). For example, a web search engine often returns thousands of pages in response to a broad query, making it difficult for users to browse or to identify relevant information.

Clustering methods can be used to automatically group the retrieved documents into a list of meaningful categories, as is achieved by Enterprise Search engines such as: Northern Light and Vivisimo Andrews and Fox (2007). However, in this case scalability becomes a big issue as the number of documents increases day-by-day, thereby necessitating the need to cluster documents dynamically, without disturbing the formulated clusters. By clustering documents dynamically, the time and effort taken for clustering is drastically reduced, as dynamic algorithms processes the new document and assigns it into the meaningful clusters directly, instead of re-clustering the entire document in the corpus. Though some document clustering methods exist for clustering documents in a dynamic environment which are based on terms Wang et al. (2011) or Synonyms and Hypernyms Nadig et al. (2008), they are not best suited for documents that are technically related. To overcome to above limitations, a model for dynamic document clustering based on Term frequency and Correlated Terms (Terms and their related terms) as concepts in Scientific literature and Newsgroups data set, is proposed in this paper. The three new algorithms, namely, Term frequency based MAximum Resemblance Document Clustering (TMARDC), Correlated Concept based MAximum Resemblance Document Clustering (CCMARDC) and Correlated Concept based Fast Incremental Clustering Algorithm (CCFICA) are proposed and the performance of the above have been compared with four existing algorithms, namely, Semantic Similarity based Histogram based Incremental Document Clustering (SHC), Concept-Based Mining Model (CBM), Incremental Algorithm for Clustering Search Results (ICA) and Enhanced Similarity Histogram Clustering using Intra Centroid Vector Similarity (ESHC-IntraCVS) on the same datasets, and results are presented.

The remaining part of this paper is organized as follows. Section "Related work" reviews related work on static and dynamic document clustering. Section "Overview of existing document clustering considered for comparative analysis", outlines the general model for dynamic document clustering, also, the need for considering correlated terms are briefly stated in that section. In section 4 presents, the new clustering algorithms, namely, TMARDC, CCMARDC and CCFICA clustering algorithms have been described in detail. In Section 5, the experimental setup and data set descriptions have been discussed, followed by analysis of results. Finally salient conclusions are presented in section “Experimental results”.

Methods

We have conducted systematic and structured reviews to identify the issues in the existing dynamic document clustering algorithms. To overcome the issues in the exiting work, three algorithms namely Term frequency based MAximum Resemblance Document Clustering (TMARDC), Correlated Concept based MAximum Resemblance Document Clustering (CCMARDC) and Correlated Concept based Fast Incremental Clustering Algorithm (CCFICA) have been proposed. To justify the potential of the proposed algorithm experiments are conducted on two dataset. The performance of the proposed algorithm shows better results compared to the existing algorithm.

Related works

Most of the existing document clustering methods are based on the Vector Space Model (VSM) which is a widely used data representation for text classification and clustering Aas and Eikvil (1999). In VSM the document is represented as a feature vector of the terms in the document. Each feature vector contains term-weights of the terms in the document. Term Frequency–Inverse Document Frequency (TF-IDF) is a weight used which is a statistical measure, is used as a weight to evaluate ‘how important a word is’ to a document in a collection or corpus Salton and Buckley (1998). The importance increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus. The similarity between the documents is measured by one of several similarity measures that are based on such a feature vector Huang (2008). Common ones include the cosine measure and the Jaccard measure.

Li and Zhu (2011) proposed a new method for Document Clustering in Research Literature, based on Negative Matrix Factorization (NMF) and topic discovery based on Test or theory. This method, clusters research literature documents comprising NMF and Test or theory. The NMF method is most prominent for high dimensionality reduction in text data and clustering them. Test or theory is used to discover topics for the documents clustered by the NMF method, by constructing learning matrix and comparison matrix. Using the above, a case study has been provided for the automatic classification of conference proceedings in Chinese. The combination of NMF and Test or theory provides effective results. Many document clustering algorithms are based on term frequency Kumar and Srinathan (2009; Luo et al. 2009; Ni et al. 2010). Several researchers have proposed clustering based on synonyms and hypernyms Bharathi and Vengatesan (2012; Pessiot et al. 2010; Li et al. 2008; Danushka et al. 2011; Kaiser et al. 2009; Shehata 2010; Baghel and Dhir 2010). An overview of a recent survey of existing dynamic document clustering algorithms, along with the details of document representation, similarity measure and the dataset used for experimental analysis, are presented in Table 1.

Table 1 Survey of recent document clustering algorithms

Algorithm name with author(s)	Technical abbreviation	Representation	Similarity measure	Data set used
Threshold Resilient Online Algorithm Chou and Chen (2008)	IPLSI(Incremental Probabilistic Latent Semantic Indexing)	Latent Semantic Variables	A latent variable is introduced between documents and terms, Cosine function	NIST TDT Corpora
Efficient Phrase Based Indexing Hammouda and Kamel (2004)	Uses DIG(Document Index Graph) for Web Clustering	Document Index Graph (Phrase Based Representation)	Phrase Based Similarity measure	USENET News Groups
Component-Based Clustering Algorithms Boris et al. (2012)	IR(Initial Representative), MD(Measure Distance), UR(Update Representatives), EC(Evaluate Clusters), SC(Stop Criterion)	Object-Based Software Representation	CITY,CORREL, COSINE, ELUCID	10 UCI Datasets
Temporal Queries and Version Management Zaniolo and Wang (2008)	XML Techniques	V-Document (XML Document)	——	W3C, World Fact Book
Density –Based Methods for Hierarchical Clustering Chehreghani and Abolhassani (2008)	3-Phases: Insertion Phase, Extraction Phase, Combination Phase	M-Tree Structure	Relative distance between objects	DMOZ, NEWS, REUTERS
XML Schema Matching Algorithm Alsayed et al. (2009)	NPS(Number Prufer Sequences), LPS(Label Prufer Sequences)	Prufer Sequences, Schema Trees	The distance between two nodes in the schema tree	XCBL, OAGIS
Novel Web User Clustering Method Ling et al. (2009)	A 3Phase COWES Algorithm	A Web Session Subtree	DoC(Degree of Change), FoC(Frequency of Change) and SoC(Significance of Change)	Internet Traffic Archive
Multi-label Document Clustering Algorithm Chen et al. (2010)	FMDC(Fuzzy Based Multi-label Document Clustering) – Fuzzy Association Rule + Existing Ontology	Terms and Hypernyms Representation of documents	Membership Functions and Document Term Matrix	Classic, Re0, R8, and WebKB
Incremental Construction of Multilingual Topic Maps Ellouze et al. (2012)	CITOM(Construction Incremental Topic Map)	Topic Map Model Representation	Topic Map Pruning Process	Multilingual corpora
Feature Extraction Algorithm Yan et al. (2011)	TOFA(Trace-Oriented Feature Analysis)	Bag Of Words Model(BOW)	Latent Semantic Indexing(LSI)	20NG, RVCI, ODP
Correlation Similarity Measure Space Zhang et al. (2011)	CPI(Correlation Preserving Index)	Terms and related terms	Correlation similarity	20NG
Contextual Document Cluster Rooney et al. (2006)	CDC(Contextual Document Cluster)	Term Document Representation	Adjacent Document Similarity	RCVI
Framework of Wikipedia-Based Clustering Hu et al. (2009)	Exact-match and Relatedness-match	Concept feature vector and Category feature vector	Complete Linkage as cluster distance measure	20-newsgroup, TDT2, LA Times

Critical analysis of the recent document clustering algorithms, as presented in Table 1 reveals that document are represented (i) based on phrase or pair-wise concept, where in the similarity relationship between the sentences are identified as used Hammouda and Kamel (2004; Lam and Hwuang 2009); (ii) using tree representation and similarity between two objects or nodes are identified and clustered Zaniolo and Wang (2008; Chehreghani and Abolhassani 2008; Alsayed et al. 2009); (iii) component based clustering algorithm which makes use of object – based software representation for modeling the document and cosine and Euclid measure for document clustering Boris et al. (2012); (iv) identifying the semantic relations and representing the documents based on Terms and Related terms Zhang et al. (2011); (v) as concept and feature vector Hu et al. (2009). Most of the above works are based on web page information representation, tracking and retrieval.

Prathima and Supreethi (2011) presented a survey of concept based clustering algorithms, and concluded that most of the clustering techniques use TF-IDF method. This method has the following issues:

- It fails to differentiate the degree of semantic importance of each term;
- It assign weights without distinguishing between semantically important and unimportant words within the document and
- It does not consider synonyms, polysemous, etc.

Based on the critical analysis of published literature, it is inferred that more than 60% of clustering techniques is based on term frequencies. About 30% of clustering techniques and annotation tools use synonyms and hypernyms for predicting the concepts. Moreover, the synonyms and Hypernyms are extracted by means of WordNet lexical database Miller (1995). Since scientific literature and many tracks of news documents consist of purely domain-specific technical terms, the performance of synonyms and hypernyms based clustering may not always yield better results. In order to enhance the quality of the cluster for the above mentioned document sets, the focus of the present study is on clustering the document based on terms and their technically related terms. In this regard, a domain- specific dictionary has been developed by the authors to extract the related terms as concepts.

Overview of existing document clustering considered for comparative analysis

Three existing algorithms that have been chosen for the comparative analysis (with that of the proposed algorithms) are briefly described below.

Semantic similarity histogram based incremental document clustering (SHC) algorithm

Gad and Kamel (2010) proposed an incremental clustering algorithm based on Phrase-Semantic Similarity Histogram (PSSM). This algorithm integrates the text semantic to the incremental clustering process. The clusters are represented using semantic histogram which measures the distribution of semantic similarities within each cluster. The PSSM which is based on single word analysis and phrase analysis, assigns and adjusts the term weight (word/phrase) based on its relationships with semantically similar terms that occur together in the document. As soon as the new document is incrementally added to the cluster, the semantic histogram ratio is calculated and the insertion order problem is addressed by making bad documents that reduce the cluster cohesiveness to leave, and reassign them to a more appropriate cluster.

Enhanced similarity histogram clustering using intra centroid vector similarity (ESHC-intra CVS) algorithm

Gavin and Yue (2009) proposed an enhanced incremental clustering approach to develop a better clustering algorithm that helps to organize the information available on the internet in an incremental fashion in a better way. This enhanced algorithm works with the idea that the cluster that contains a large number of similar documents to the current document being clustered will have a centroid vector that has a high similarity to the current document. Therefore, the cluster whose centroid vector is most similar to the document's vector representation is the one that most likely to contain the maximum number of documents that are more similar to the current document. Adding the new document to this cluster (when possible) will probably give the greatest benefit to that cluster and the entire dataset. This approach uses the same pair-wise document similarity representation and distribution approach and also uses additional information about the cluster to determine the best cluster to place the new document.

Concept-based mining model (CBM)

Shehata et al. (2010) proposed a Concept- based Mining Model for Enhancing Text Clustering Mining model. The proposed concept-based mining model consists of sentence-based concept analysis, document-based concept analysis, corpus-based concept-analysis, and concept-based similarity measure. By combining the factors affecting the weights of concepts on the sentence, document, and corpus levels, a concept-based similarity measure that is capable of accurate calculation of pair-wise documents, is formulated. This allows performing concept matching and concept-based similarity calculations among documents in an accurate way. The quality of text clustering achieved by this model significantly surpasses the traditional single term-based approaches like: (i) Hierarchical Agglomerative Clustering (HAC), (ii) Single-Pass Clustering, and (iii) k-Nearest Neighbor (k-NN).

An incremental algorithm for clustering search results (ICA)

Liu et al. (2008) proposed an incremental clustering algorithm based on Cluster Average Similarity Area (CASA), which was used to score the degree of coherency of a cluster. The cohesiveness quality information of a cluster was computed based on its CASA. The above algorithm works by processing data objects one at a time, incrementally assigning data objects to their respective clusters while they progress.

A model for dynamic document clustering

Figure 1 shows the sequence of steps involved in dynamic document clustering. The document which is in unstructured format are preprocessed and converted to a structured format. The details of each module involved in the model namely, preprocessing, static clustering and dynamic document clustering are discussed below.

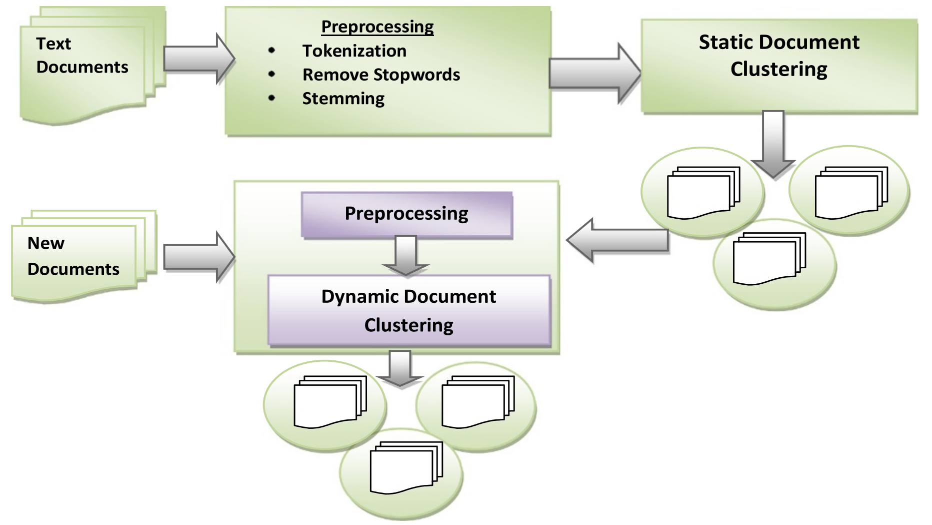


Figure 1 **A general model for dynamic document clustering.**

Preprocessing

Preprocessing involves: tokenization, removing stopwords and stemming.

Tokenization (Christopher et. al. http://nlp.stanford.edu/software/tokenizer.shtml), is the process of splitting the sentences into separate tokens. For example, "this is a paper about document clustering" is split as: this\is\paper\about\document\clustering. Stop words are frequently occurring words that have little or no discriminating power, such as: \a", \about", \all", etc., or other domain-dependent words. Stop words are often removed. Stemming is the process of removing the affixes in the words and producing the root word known as the stem Frakes and Fox (2003). Typically; the stemming process is performed to transform the words into their root form. For example: connected, connecting and connection would be transformed into 'connect'. Most widely used stemming algorithms are the ones proposed by Porter (1998), Lovins (1968), and S-removal Harman (1991).

Static document clustering

The processed documents are clustered using a Bisecting K-means clustering algorithm in order to group similar documents. Cluster analysis or clustering is the assignment of a set of observations into subsets (called clusters) so that observations of the same cluster are similar in some sense. The Bisecting K-means method will split a large cluster into two sub-clusters and this step will be repeated for several times, until the K numbers of clusters are formed with high similarity Steinbach et al. (2000).

Dynamic document clustering

Dynamic Document Clustering is the process of inserting the newly arrived documents to the appropriate existing cluster such that the formulated cluster will have a high intra- cluster similarity, and less inter-cluster similarity. At first the new documents are preprocessed and then it is clustered based on the dynamic technique. The issues that are to be addressed are:

- Effectiveness: How accurately the newly arrived documents are inserted to the existing clusters.
- Insertion Order Issue: Pattern of arrival of new documents should not affect the correctness of the clusters.

The new documents are assigned to the existing cluster, one by one in recursive steps. The new documents are assigned to a cluster dynamically at run time without the need for re-clustering. As a result the existing clusters are updated and the final clusters are obtained. In this study, the three newly proposed algorithms TMARDC, CCMARDC and CCFICA are experimented for clustering the documents dynamically. The details of these three algorithms are discussed in the next section.

Proposed algorithms for dynamic document clustering

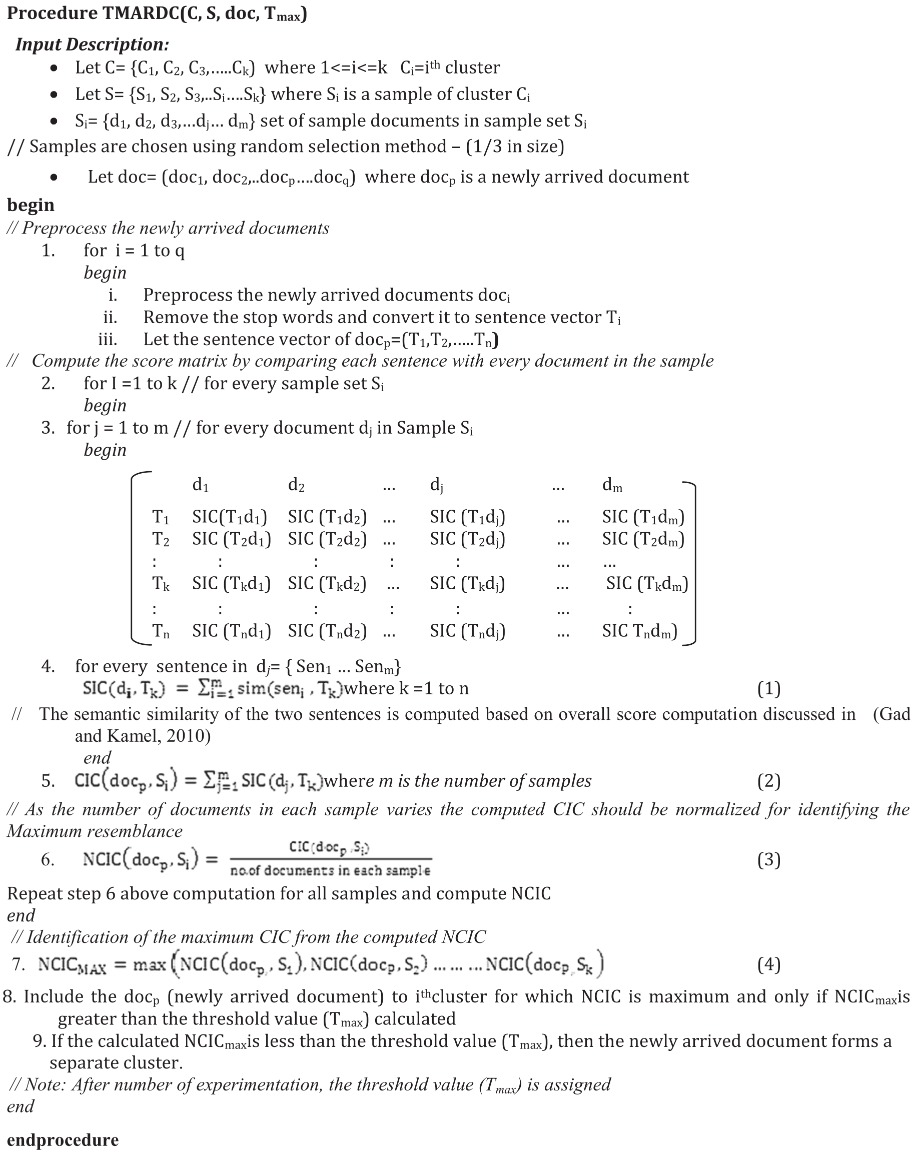
This section describes the proposed Term frequency based MAximum Resemblance Document Clustering (TMARDC) algorithm, Correlated Concept based MAximum Resemblance Document Clustering (CCMARDC) algorithm, and Correlated Concept based Fast Incremental Clustering Algorithm (CCFICA) for dynamic clustering.

Term frequency based maximum resemblance document clustering (TMARDC)

This algorithm adopts the core concept of MARDL i.e. Maximum Resemblance technique Chen et al. (2008). This algorithm is purely based on a bag of words representation. This dynamic algorithm starts with the set of clusters which is obtained as the result of bisecting K-Means clustering. Initially, the sample set is constructed for each cluster set. One third of the documents are chosen randomly as samples from the set of documents in each cluster. The samples chosen should be unique and should not be replica's of documents in samples. The new documents are preprocessed first which includes stop word removal process and stemming process. The new documents are stemmed using a stemming algorithm. After preprocessing of the new document, the new document is compared with samples based on *Sentence Importance computation (SIC), Cluster set Importance computation (CIC)* and the influence of the new document in each cluster termed as *Frequency Value (FV*) is calculated. The CIC should be normalized to obtain the *FV*, because the number of documents in each sample may vary.

Then the dynamic algorithm assigns the new document to the cluster with the high *FV*, provided, the *FV* is within the threshold value. The threshold value is maintained for clustering process to make a document to form a new cluster or assigning a document to the appropriate cluster. If all the clusters result in *FV* less than the threshold value, then, the new document forms a separate cluster. The threshold value is calculated through a series of experiments on all worst, average and best case inputs and it is termed as Threshold value (T_{max}). A newly arrived document, if it's *FV* falls less than the T_{max} it forms a separate cluster, thus ensuring that no document goes without clustering, even it doesn't patches with any of the existing clusters.

Algorithm TMARDC



```
Procedure TMARDC(C, S, doc, $T_{max}$)
 Input Description:
    • Let C= {$C_1$, $C_2$, $C_3$,.....$C_k$)  where 1<=i<=k   $C_i$=$i^{th}$ cluster
    • Let S= {$S_1$, $S_2$, $S_3$,..$S_i$....$S_k$} where $S_i$ is a sample of cluster $C_i$
    • $S_i$= {$d_1$, $d_2$, $d_3$,...$d_j$... $d_m$} set of sample documents in sample set $S_i$
// Samples are chosen using random selection method – (1/3 in size)
    • Let doc= ($doc_1$, $doc_2$,..$doc_p$....$doc_q$)  where $doc_p$ is a newly arrived document
begin
// Preprocess the newly arrived documents
    1.  for  i = 1 to q
        begin
            i.   Preprocess the newly arrived documents $doc_i$
            ii.  Remove the stop words and convert it to sentence vector $T_i$
            iii. Let the sentence vector of $doc_p$=($T_1$,$T_2$,.....$T_n$)
//  Compute the score matrix by comparing each sentence with every document in the sample
    2.  for I =1 to k // for every sample set $S_i$
        begin
    3.  for j = 1 to m // for every document $d_j$ in Sample $S_i$
        begin
```

$$\left[\begin{array}{lllllll} & d_1 & d_2 & \dots & d_j & \dots & d_m \\ T_1 & SIC(T_1d_1) & SIC\ (T_1d_2) & \dots & SIC\ (T_1d_j) & \dots & SIC\ (T_1d_m) \\ T_2 & SIC\ (T_2d_1) & SIC\ (T_2d_2) & \dots & SIC\ (T_2d_j) & \dots & SIC\ (T_2d_m) \\ : & : & : & : & : & \dots & \dots \\ T_k & SIC\ (T_kd_1) & SIC\ (T_kd_2) & \dots & SIC\ (T_kd_j) & \dots & SIC\ (T_kd_m) \\ : & : & : & : & : & \dots & : \\ T_n & SIC\ (T_nd_1) & SIC\ (T_nd_2) & \dots & SIC\ (T_nd_j) & \dots & SIC\ T_nd_m) \end{array}\right]$$

```
    4.  for every  sentence in  $d_j$= { $Sen_1$ ... $Sen_m$}
        $SIC(d_i, T_k) = \sum_{i=1}^{m} sim(sen_i , T_k)$where k =1 to n                     (1)
//  The semantic similarity of the two sentences is computed based on overall score computation discussed in   (Gad
        and Kamel, 2010)
        end
    5.  $CIC(doc_p, S_i) = \sum_{j=1}^{m} SIC(d_j, T_k)$where m is the number of samples           (2)
// As the number of documents in each sample varies the computed CIC should be normalized for identifying the
Maximum resemblance
    6.  $NCIC(doc_p, S_i) = \frac{CIC(doc_p, S_i)}{no.of\ documents\ in\ each\ sample}$                   (3)
Repeat step 6 above computation for all samples and compute NCIC
end
 // Identification of the maximum CIC from the computed NCIC
 7. $NCIC_{MAX} = max\left(NCIC(doc_p, S_1), NCIC(doc_p, S_2) \dots\dots\dots NCIC(doc_p, S_k\right)$     (4)
8. Include the $doc_p$ (newly arrived document) to $i^{th}$cluster for which NCIC is maximum and only if $NCIC_{max}$is
        greater than the threshold value ($T_{max}$) calculated
    9. If the calculated $NCIC_{max}$is less than the threshold value ($T_{max}$), then the newly arrived document forms a
       separate cluster.
 // Note: After number of experimentation, the threshold value ($T_{max}$) is assigned
end

endprocedure
```

Correlated concept based maximum resemblance document clustering (CCMARDC)

Incorporation of semantic features, improve the quality of document clustering and also the accuracy of information extraction techniques. In this study, concept extraction algorithm introduced by Jayabharathy et al. (2011), which itself is a modification of the existing semantic-based model proposed by Shehata (2009) has been adopted. The model proposed by Shehata (2009) aims to cluster documents by meaning. The semantic-based similarity measure is used for the two CCMARDC and CCFICA algorithms, proposed in this study. In order to extract concepts, a domain-specific dictionary consisting of scientific terms and terms related to newsgroup tracks are created unlike the work of Shehata (2009), where in Word Net lexical database Miller (1995) was used for Synonyms/Hypernyms extraction. Domain-specific dictionary for scientific and Newsgroups are used for concept extraction, as it eliminates the need for word sense disambiguation (WSD) Banerjee and Pedersen (2002; http://en.wikipedia.org/wiki/Word_sense_disambiguation), which is not the scope of the present study.

Why correlated terms?

There are many existing clustering algorithms that take synonyms and hypernyms for vector representation. In this study, the authors have considered *crtv* as concepts for clustering to improve the efficiency of clustering the documents both statically and dynamically. The idea of considering terms and related terms as concepts based on semantic similarity has been carried out for extracting topic from the clustered documents Jayabharathy et al. (2011). The proposed technique CCMARDC takes this idea of considering *crtv* as concepts for static clustering and applies the same concept for clustering the document dynamically. Considering terms or synonyms and hypernyms for information extraction leads the following issues:

Case 1: Words have multiple meanings, hence diversifies the information extraction.
E.g. Bat : represents the cricket bat or a kind of a bird.
Case 2: Considering terms or synonyms of the terms limits the search space of the domain.
E.g. wireless: first sense medium of communication.

Similarly, synonyms of the term "*wireless*" is extracted from WordNet as: "*first sense medium of communication*", whereas, taking related terms like "*wireless*", "*communication*", "*protocol*" "*mobile communication*" etc. will be extracted as concepts, which gives better accuracy and improves the efficiency of information extraction. For example, *sports article* contains terms like: a *ball, bat, wicket, run, batsman, over* etc. Taking synonyms/hypernyms as concept, will not give better performance since the meaning of these terms are not literally same. If we consider the technically related terms i.e. *crtv,* all the above mentioned terms will be grouped together as a single concept which refers sports related to the concept – *cricket*. Similarly the synonym for the term "*farmer"* from WordNet is extracted as: *"a person Title who operates a farm*". But using the proposed model the concept will be extracted as *"farmer", "crops", "fertilizer", "land" and "farm".* Clustering the document using this extraction procedure would improve the performance of the resulting cluster, than that of the cluster generated by existing works.

Concept extraction algorithm: description

Considering the extraction of Synonyms/Hypernyms as concepts degrades the efficiency of the results in the case of scientific literature and news group dataset because of the fact that the documents speak more about scientific or technical terms. Concept extraction is based on our previous work Jayabharathy et al. (2011), where Correlated concepts are nothing but the terms and their related terms. For Concept extraction, domain specific dictionary is used where terms related to each domain is kept along with the meaning of the term. For e.g. the terms A and B are taken as a concept; if term A is in the definition of term B or vice versa combines A and B as a single concept else add the definition of A and B as separate concept to the concept list. E.g. Considering share market as the term in the news documents, the related terms are share, shareholder, money, market. The documents containing these words are grouped together as share market which forms the cluster.

The framework of the proposed correlated concept based maximum resemblance document clustering (CCMARDC)

The Figure 2 illustrates the processes involved in the proposed Correlated Concept based MAximum Document clustering (CCMARDC). This algorithm is similar to the TMARDC algorithm, the main difference is that the documents are represented as correlated concepts for clustering instead of term frequency. In addition, a new module is integrated, which is meant for concept extraction and interaction with domain-specific dictionary. Not only that, instead of computing the sentence similarity between the new document and documents in the sample set, the semantic similarity between the new document and the document(s) in the sample set S_i is computed. The above mentioned process of TMARDC algorithm is repeated for clustering process and for inclusion of new document based on correlated concept.

Similarity measure

The semantic-based similarity between two documents d_1 and d_2 is calculated. This similarity measure is a function of the following factors Shehata (2009):

- The number of matching concepts, (mc), in each document (d);
- The total number of the labeled verb-argument structures (*v*), in each sentence (*st*);

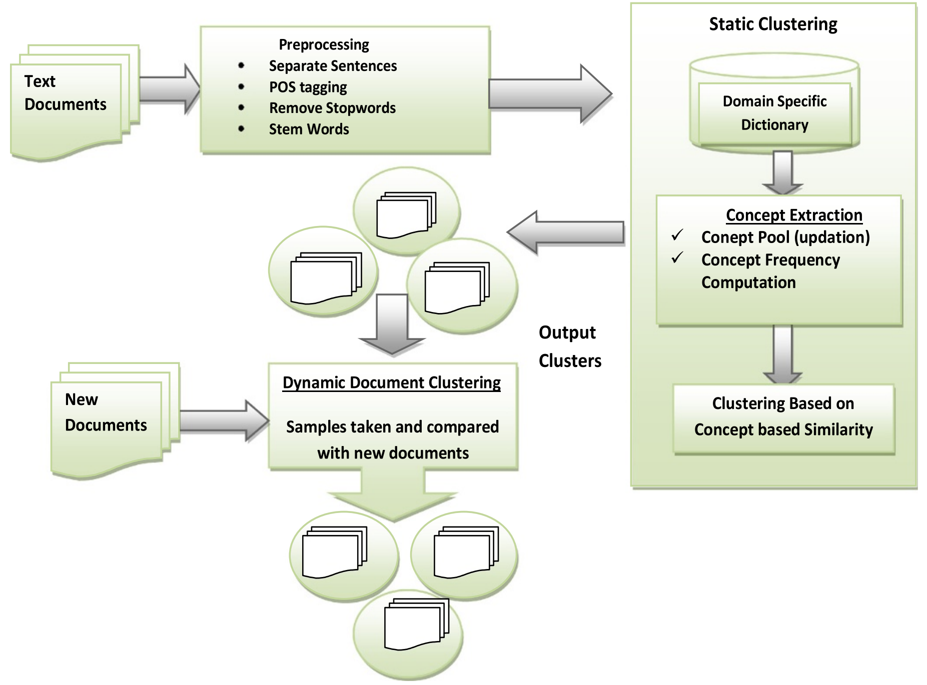

Text Documents
Preprocessing
Separate Sentences
POS tagging
Remove Stopwords
Stem Words
Static Clustering
Domain Specific Dictionary
Concept Extraction
Conept Pool (updation)
Concept Frequency Computation
Clustering Based on Concept based Similarity
Output Clusters
New Documents
Dynamic Document Clustering
Samples taken and compared with new documents

Figure 2 Correlated concept based MAximum resemblance document clustering.

- The ctf_i of each concept c_i in st for each document d, where $i = 1, 2, \ldots, mc$ *and*
- The $c\,f_i$ of each concept c_i in each document d, where $i = 1, 2, \ldots, mc$

$$\mathrm{sim_s}\left(\mathrm{doc_p}, \mathrm{d_j}\right) = \sum\nolimits_{i-1}^{mc} \mathrm{weight_{i1}} * \mathrm{weight_{i2}} \tag{5}$$

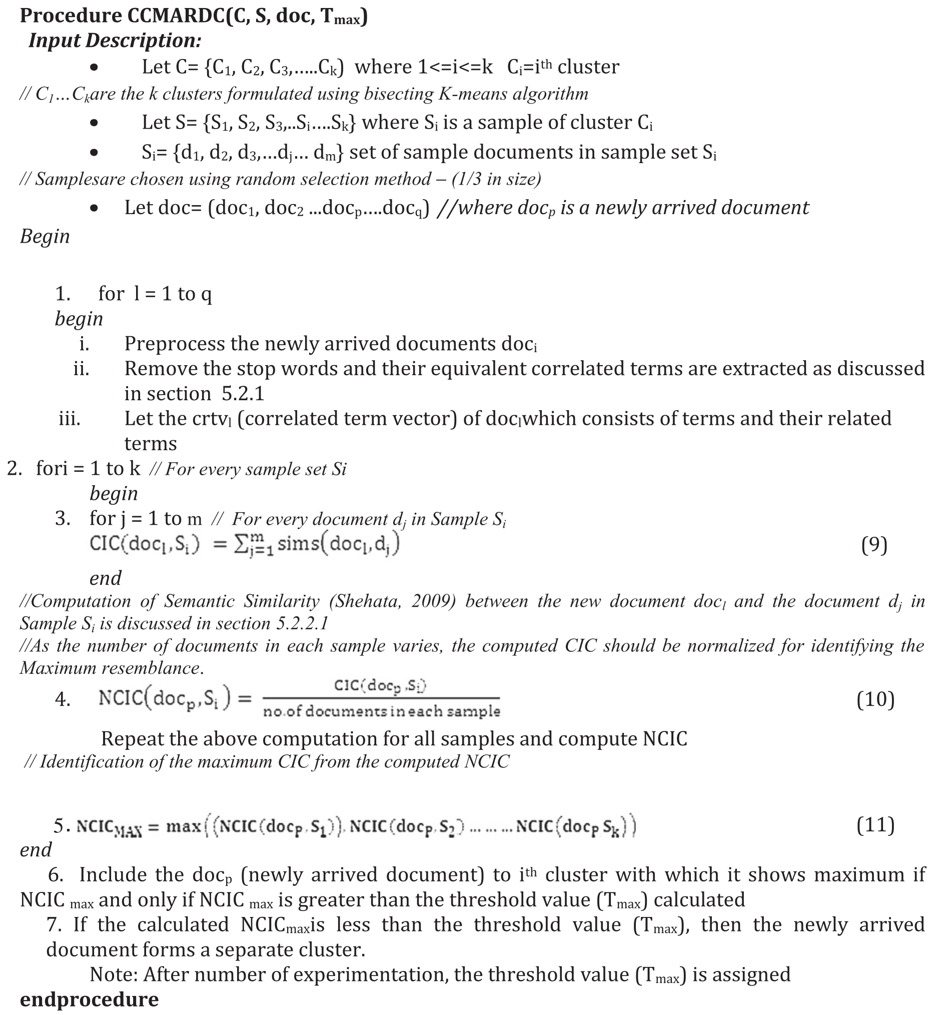
$$\mathrm{weight_i} = \mathrm{cf\ weight_i} + \mathrm{ctf\ weight_i} \tag{6}$$

$$\mathrm{cf\ weight_i} = \mathrm{cf_{ij}} / \left(\sum\nolimits_{j=1}^{cn} \left(\mathrm{cf_{ij}}\right)^2\right)^{1}/_{2} \tag{7}$$

$$\mathrm{ctf\ weight_i} = \mathrm{ct\ f_{ij}} / \left(\ \sum\nolimits_{j-1}^{cn} \left(\mathrm{ct\ f_{ij}}\right)^2\ \right)^{1}/_{2} \tag{8}$$

Where *cn* is the total number of concepts which have a conceptual term frequency value in document *d*.

Algorithm CCMARDC



```
Procedure CCMARDC(C, S, doc, $T_{max}$)
 Input Description:
        •   Let C= {$C_1$, $C_2$, $C_3$,.....$C_k$)  where 1<=i<=k   $C_i$=$i^{th}$ cluster
// $C_1$...$C_k$are the k clusters formulated using bisecting K-means algorithm
        •   Let S= {$S_1$, $S_2$, $S_3$,..$S_i$....$S_k$} where $S_i$ is a sample of cluster $C_i$
        •   $S_i$= {$d_1$, $d_2$, $d_3$,...$d_j$... $d_m$} set of sample documents in sample set $S_i$
// Samplesare chosen using random selection method – (1/3 in size)
        •   Let doc= ($doc_1$, $doc_2$ ...$doc_p$....$doc_q$) //where $doc_p$ is a newly arrived document
Begin

    1.   for  l = 1 to q
    begin
      i.     Preprocess the newly arrived documents $doc_i$
     ii.     Remove the stop words and their equivalent correlated terms are extracted as discussed
             in section  5.2.1
    iii.     Let the $crtv_l$ (correlated term vector) of $doc_l$which consists of terms and their related
             terms
 2.  fori = 1 to k  // For every sample set Si
          begin
    3.    for j = 1 to m  //  For every document $d_j$ in Sample $S_i$
          $CIC(doc_l, S_i) = \sum_{j=1}^{m} sims(doc_l, d_j)$                          (9)
          end
//Computation of Semantic Similarity (Shehata, 2009) between the new document $doc_l$ and the document $d_j$ in
Sample $S_i$ is discussed in section 5.2.2.1
//As the number of documents in each sample varies, the computed CIC should be normalized for identifying the
Maximum resemblance.
    4.    $NCIC(doc_p, S_i) = \frac{CIC(doc_p, S_i)}{no.of\ documents\ in\ each\ sample}$                    (10)
            Repeat the above computation for all samples and compute NCIC
 // Identification of the maximum CIC from the computed NCIC

    5. $NCIC_{MAX} = max((NCIC(doc_p, S_1)), NCIC(doc_p, S_2) \ldots \ldots \ldots NCIC(doc_p S_k))$   (11)
end
    6.  Include the $doc_p$ (newly arrived document) to $i^{th}$ cluster with which it shows maximum if
NCIC $_{max}$ and only if NCIC $_{max}$ is greater than the threshold value ($T_{max}$) calculated
    7. If the calculated $NCIC_{max}$is less than the threshold value ($T_{max}$), then the newly arrived
    document forms a separate cluster.
          Note: After number of experimentation, the threshold value ($T_{max}$) is assigned
endprocedure
```

Correlated concept based fast incremental clustering algorithm (CCFICA)

Xiaoke et al. (2009) proposed Fast Incremental Clustering Algorithm (FICA) an increment data clustering algorithm for mushroom data set. The main objective of this algorithm is to cluster the categorical data into the K number of clusters using incremental method. The existing algorithm uses dissimilarity measure for finding the distance between the new object and the existing cluster. The core idea of the above algorithm is considered in the CCFICA proposed here. The FICA algorithm is modified for clustering the documents for dynamic document corpuses, based on semantic similarity. For every cluster, the top correlated concepts from each document are extracted and are maintained as a concept pool. Instead of computing the dissimilarity between document clusters and the new document, the semantic similarity between the new document and the concept pool is computed, which reduces the computation overhead.

Algorithm CCFICA

```
Procedure CCFICA ( DS, k,vt)
Input Description:
                • Newly arrived document Set DS ;
                • The number of clusters k ;
                • The threshold value vt;    n=0;
    1. If  n=0; no. of formulated clusters is 0
       begin
                 i. Initialize CS as an empty set , and read a new document $doc_p$ from DS
                ii. Create a cluster with the document and place it into the collection CS
               iii. Create Concept pool CP for the clusters in CS where $CP_i$ is the Concept pool for cluster set i
                iv.Update n=n+1
// since new cluster is formulated
end
    2. if n> k  // the number of the clusters is more than k;
begin
       for every cluster pool compute pairwise similarity
               begin             //Compute the semantic based similarity between concept pool(s)
               Compute $Sim_s$ ($CP_i$, $CP_j$)
               end
// Semantic similairy between concept pool i and j
// Concept pool is a collection of top correlated terms for each document in the cluster set
                  i.   Identify,  two  $CP_i$, $CP_j$such that  Sim($CP_i$, $CP_j$) is maximum and is greater than the threshold  vt
                 ii.   Merge the two clusters $C_i$ and $C_j$
                iii.   Merge both of  their concept pool $CP_i$ and  $CP_j$
                 iv.   Update n=n-1
       end
    3.   If n<K  and DS not empty
// If the number of the clusters is less than k;
       begin
            i.     read a new document $doc_p$
           ii.     Compute the similarity between $doc_p$and each cluster
                         for j= 1 to n beginCompute Sim ($CP_{i,}doc_p$)end
          iii.    Select and  add new document $doc_p$ to the cluster j with maximum similarity
           iv.     Update the cluster j concept pool $CP_j$. Go to step 4.
            v.     If $doc_p$doesn't belong to any existing cluster then create new cluster and the cluster pool.
           vi.    Update n=n+1
       end
     4: If DS is empty, Stop else goto step 1
 end
 endprocedure
```

Experimental results

Data set

The data set used for the experimental analysis contains 500 abstract articles collected from the Science Direct digital library. The articles are classified according to the Science Direct classification system into four major categories: computer networks and communications, nuclear and high energy physics, economics and econometrics, and civil and structural engineering. In addition, to that 20 Newgroups is considered as another data, set for the result analysis which consists of more than 1000 news articles related to Sports, Political and Share market tracks.

Performance metrics

F-measure and Purity are the performance measures used to evaluate the quality of document clustering. F-measure combines the Precision and Recall from information retrieval process Steinbach et al. (2000). Each cluster is treated as if it were the result of a query, and each class as if it were the desired set of documents, for a query. The recall and precision of that cluster for each given class are calculated. More specifically, F-measure for cluster *j* and class *i*is calculated as follows:

$$\mathrm{Recall(i,j)} = \frac{\mathrm{n_{ij}}}{\mathrm{n_i}} \tag{12}$$

$$\mathrm{Precision(i,j)} = \frac{\mathrm{n_{ij}}}{\mathrm{n_j}} \tag{13}$$

$$\mathrm{F(i,j)} = \frac{\mathrm{(2 * Recall(i,j) * Precision(i,j))}}{\mathrm{Presicion(i,j) + Recall(i,j)}} \tag{14}$$

Where n_{ij} is the number of members of the class i in cluster j, n_j is the number of members of cluster j and n_i is the number of members of class i. For each class, only the cluster with highest F-measure is selected. Finally, the overall F-measure of a clustering solution is weighted by the size of each cluster:

$$\mathrm{F(S)} = \frac{1}{\mathrm{n}} \sum\nolimits_{\mathrm{j=1}}^{\mathrm{n}} \frac{\mathrm{n_j}}{\mathrm{max(F(i,j))}} \tag{15}$$

The purity measure evaluates the coherence of a cluster, that is, the degree to which a cluster contains documents from a single class Huang (2008). Given a particular cluster C_i of size n_i the purity of C_i is formally defined as:

$$\mathrm{P(C_i)} = \frac{1}{\mathrm{n}} \mathrm{max}\left(\mathrm{n_i^h}\right) \tag{16}$$

Where $max(n_i^h)$ is the number of documents that are from the dominant class in cluster C_i and n_i^h represents the number of documents from cluster C_i assigned to class h. The overall purity of a clustering solution is:

$$\mathrm{Pur}ity(S) = \frac{1}{n} \sum\nolimits_{\mathrm{i-1}}^{\mathrm{n}} \mathrm{max}\left(n_i^h\right) \tag{17}$$

Implementation procedure

Initially, text documents which have been collected from various sources were accumulated in a database. Then, pre-processing was carried out by considering the various stages like:

tagging by means of Stanford POS tagger tool, stop word removal and stemming, based on Porter Stemmer algorithm and morphological capabilities of WordNet. The above preprocessing is common for both existing and proposed algorithms considered in this study. Then the documents are represented as VSM. These documents are clustered using Bisecting K-means algorithm which generates K number of clusters.

For implementing the existing algorithms the preprocessing as outlined in this work along with dataset chosen for the study were used. The algorithms as originally proposed by the various authors were implemented in the above environment. However, for CBM, the entire model as originally proposed was not considered. Instead, the CBA algorithm and clustering- based concept semantic similarity alone is implemented. For uniformity, only the ICA clustering algorithm as originally proposed by the authors, were used in this study, even though the original ICA algorithm starts with query retrieval and then proceeds to clustering. By varying the number of documents the results of the proposed and existing algorithms are measured. These algorithms are implemented in JDK 1.7 environment using Net Beans IDE.

Results and discussion

The Table 2 describes the document representation, similarity measures and the data set adopted in the existing and the proposed algorithms. From the above Table the variations between the proposed and existing algorithms in terms of representation, similarity measure and the data set can be easily identified. The experiments are conducted by varying the number of new documents from 50 to 500 that are to be inserted in the existing clusters. Though CBA is not an incremental clustering algorithm it has been implemented as it considers the semantic relations between documents. Entire document set and the new document collection are given as input for processing in a static way.

The performance analysis of the existing (SHC, ESHC and CBA algorithms) and the proposed algorithms (TMARDC, CCMARDC and CCFICA) are categorized into three classes:

i) Based on F-measure and Purity analysis for Scientific Literature;
ii) Based on F-measure and Purity analysis for Newsgroup and
iii) Based on pair-wise performance analysis (one to one comparison) for both datasets

Table 2 Techniques adopted in existing and proposed algorithms

Algorithm	Document representation	Similarity measure	Data set
Existing algorithms			
SHC Gad and Kamel (2010)	Term weight (word/phrase relationship)	Semantic Similarity	Reuters-21578 and 20-Newsgroups
ESHC-IntraCVS Gavin and Yue (2009)	Term frequency	Cosine Similarity	UW-CAN dataset, 314 web pages from University of Waterloo
CBA (Shehata (2010); Shehata et al. 2010)	Verb argument structure	Concept similarity Measure	ACM abstract articles, Reuters, Brown corpus, Usenet newsgroups
ICA Liu et al. (2008)	Term occurrencec	Jaccard coefficient	*20NewsGroup* corpus
Proposed algorithms			
TMARDC	Term frequency	MARDL, sentence similarity	*ACM abstract articles, 20Newsgroup*
CCMARC	Correlated terms	Semantic similarity	*ACM abstract articles, 20Newsgroup*
CCFICA	Correlated terms	Semantic similarity	*ACM abstract articles, 20Newsgroup*

F-measure and purity analysis for scientific literature dataset

The quality of the formulated cluster has been assessed based on F-measure and purity as performance metrics. Figures 3 and 4 shows the results of the proposed correlated term based algorithms CCFICA, CCMARDC and TMARDC (term based approach). Both CCFICA and CCMARDC algorithms give better results compared to TMARDC and the three existing algorithms considered in this study. This is because the data set chosen for these experiments are domain-specific documents which consist of more scientific and technical terms compared to English literary terms, contained in the other dataset.

The proposed algorithms perform better than the existing algorithms, as they consider the semantic relation between the documents. In CBA, the comparison is solely based on the semantic structure (subject verb argument) of each sentence only. Though it extracts the most prominent terms in sentences, it fails to capture technical correlation of terms between the sentences and the documents. The other reason is that CBA is a static clustering technique which applies clustering process for all the document clusters including the new document (s). Clustering the entire document set is a time consuming process. Also, extraction of semantic structure (subject verb argument) from the entire document set leads to information loss; as only top sentences are extracted. As the proposed CCMARDC captures the correlated concepts through the concept extraction algorithm, and as it is also devised as a dynamic algorithm, the problem of information loss has been overcome. Hence,the proposed CCMARDC algorithm gives better results, compared to the existing CBA algorithm.

F-measure and purity analysis for newsgroup dataset

The Figures 5 and 6 show the average F-measure and Purity comparison for the existing and proposed algorithms. From these, charts it is inferred that the quality of CCMARDC and CCFICA algorithms are better than the existing CBA algorithm. The proposed TMARDC algorithm gives better performance compared to ICA, SHC and ESHC algorithms. The performance of the clustering is evaluated between two categories of algorithms as: Concept based and Term frequency based algorithms. From the above figures it is also inferred that CCMARDC& CCFICA compared with CBA gives less improvement for newsgroup dataset. The drop in the performance of the 20newsgroup dataset is due to the dominance of English literary terms in the documents, rather than technical terms. Since the above dataset consists of more literary terms, synonyms and

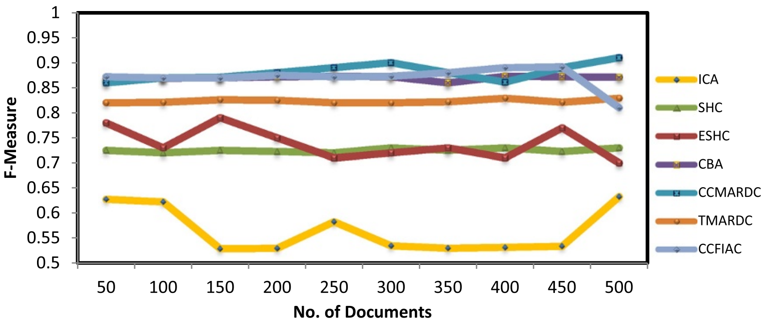


Figure 3 F-measure comparison of dynamic document clustering algorithms for scientific dataset.

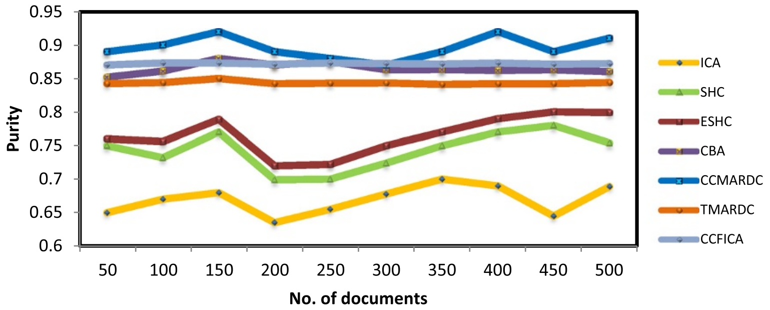


Figure 4 Purity comparison of dynamic document clustering algorithms for scientific dataset.

hypernymns based CBA algorithm works on par with the proposed algorithms. But, TMRARDC algorithm works better compared to the three existing algorithms considered in this study.

Pair-wise performance analysis (one to one comparison) for both datasets

The experiments were conducted on two categories of algorithms:

i) Clustering based on term frequency (TMARDC, ICA, SHC, ESHC)
ii) Clustering based on concepts (CCMARDC, CCFICA, CBA)

The quality of the clustering could be judged properly only when the algorithms of same category are evaluated and analyzed. To justify this statement a comparative analysis between the following pairs CBA&CCMARDC, CBA&CCFICA algorithms have been made, as CBA treats Synonyms and Hypernyms as concepts. Then, the performance evaluation between the term frequency based algorithms (i.e. TMARDC&ICA, TMARDC&SHC, TMARDC&ESHC) were analyzed. In the Figure 7 for simplicity the above pairs as: C1, C2, T1, T2 and T3, where

C1 = CBA &CCMARDL C2 = CBA& CCFICA,
T1 = TMARDC&ICA,T2 = TMARDC&ESHC and T3 = TMARDC&SHC.

Figure 7A and B illustrates the percentage of improvement for scientific literature dataset. Figure 8A and B illustrates the percentage of improvement for 20 Newsgroup

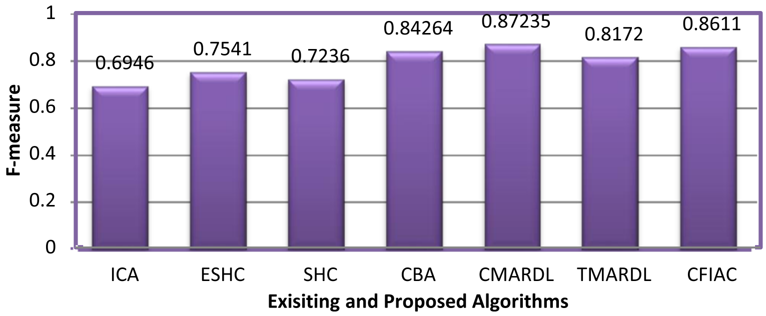


Figure 5 Average F-measure comparison between the proposed and existing algorithms for newsgroup dataset.

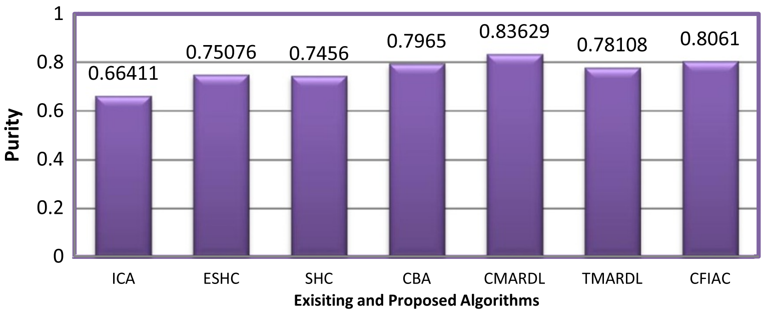


Figure 6 Average purity comparison between the proposed and existing algorithms for newsgroup dataset.

dataset. The performance outcome of the proposed algorithms are consolidated and presented in Table 3. The improvements in the performance of the proposed algorithms are grouped into three classes namely; LOW, MEDIUM and HIGH. LOW corresponds to improvement in the percentage between ±2%, MEDIUM between 0 to +11% and HIGH from +11% to +36%. From the Figure 8A and B it is inferred that CCMARDC& CCFICA when compared to CBA gives less improvement for newsgroup dataset. The drop in performance for newsgroup dataset is due to the dominance of English literary terms in the documents rather than technical terms. That is the reason CBA gives better performance for some situations than the proposed CCMARDC and CCFICA algorithms. However, it is pertinent to note here that the same CCMARDC& CCFICA algorithm when compared with CBA gives MEDIUM improvement for scientific dataset. This is because the proposed algorithms consider correlated terms, whereas, CBA takes the only Synonyms/Hypernyms contained in the scientific and 20newsgroup dataset.

The term based algorithms are also experimented with the same set of document collections and the results obtained are summarized in Table 3. It can be clearly stated that the quality of clustering based on TMARDC gives appreciable performance compared to the existing term based SHC, ESHC and ICA algorithms. This is because of identifying the prominence of each sentence of the newly arrived document with the documents of the samples using *SIC* and the relevancy of the new document against the each sample set, using CIC and NCIC, thus leading to better quality improvement. Computing the similarity between the samples and the new document(s) helps to choose a prominent cluster for inserting the newly arrived document, rather than re-clustering the entire set. Whereas,

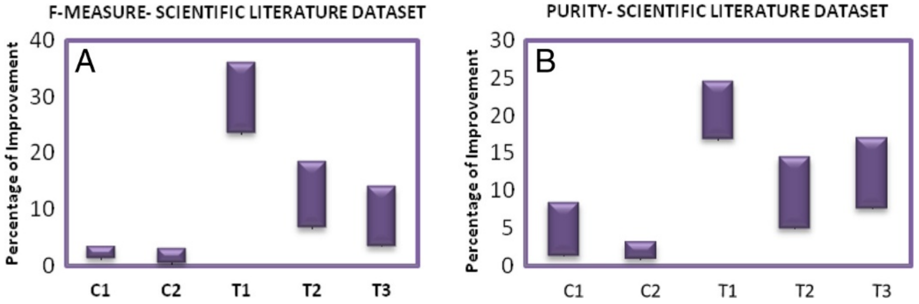


Figure 7 F-measure & purity improvement of proposed and existing algorithms for scientific literature dataset. Figure 7**A** and **B**.

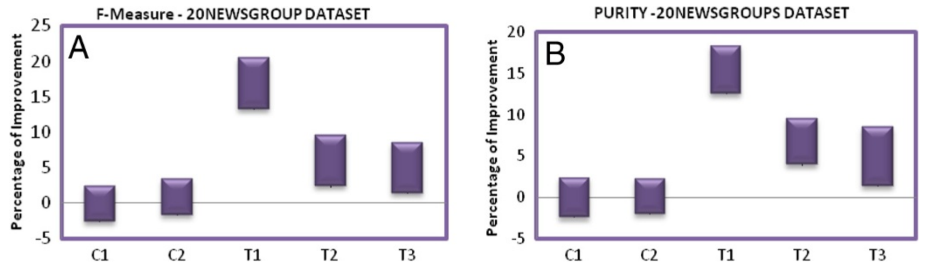


Figure 8 F-measure & purity improvement of proposed and existing algorithms for 20 newsgroup dataset. Figure 8**A** and **B**.

most of the incremental clustering algorithm works based on applying similarity measure on the entire cluster and on the new document, the proposed algorithms basically compute the similarity between the samples and new document(s) top concepts or terms. The computation overhead is thus minimized to a greater extend, as these parameters are computed against the new document and the sample set only, but not for the entire cluster. Instead of choosing random samples, choosing the documents around cluster centroid may also improve the quality.

Conclusions

The emphasis of the present work is Dynamic Document Clustering based on Term frequency and Correlated based Concept algorithms, using semantic-based similarity measure. The core idea of Data mining algorithms MARDL and FICA is adopted for the proposed algorithms TMARDC, CCMARDC and CCFICA. In general the documents are represented as TF-IDF, whereas, in this study the documents are represented by means of correlated term vector (crtv). This representation helps the user to capture the technical correlation between the documents. The proposed algorithms are compared with the existing term frequency and synonyms/hypernyms based incremental document clustering algorithms considering scientific literature and newsgroup dataset. From the comparative analysis it can concluded that considering crtv representation for dynamic document clustering leads to promising results especially for scientific literature. Sometimes the results from the Newsgroup dataset are not promising, due to the need for relatively more English literary terms, rather technical terms. In future, it is proposed to extend

Table 3 Result outcome improvement classes of proposed algorithms

DATASET	ALGORITHM COMPARISON	PURITY	F-MEASURE
20Newsgroup dataset	CCMARDC & CBA	LOW	LOW
	CFICA & CBA	LOW	LOW
	TMARDC & ICA	MEDIUM	MEDIUM
	TMARDC & ESHC	MEDIUM	MEDIUM
	TMARDC & SHC	MEDIUM	MEDIUM
Scientific literature dataset	CCMARDC & CBA	MEDIUM	MEDIUM
	CCFICA & CBA	MEDIUM	MEDIUM
	TMARDC & ICA	HIGH	HIGH
	TMARDC & SHC	HIGH	HIGH
	TMARDC & ESHC	HIGH	HIGH

concept extraction based on significant phrases in documents, and also by incorporating semantic relations like hyponymy, holonymy, and meronymy.

Competing interests

All authors declare that they have no competing interests.

Authors' contributions

JJ carried out the systematic reviews, identified the issues in the existing work. JJ and SK designed architecture and implementation of the proposed algorithms. The dataset collection, experiments and result analysis are conducted by both JJ and SK. The format of the manuscript was decided by JJ and SK. The manuscript was prepared by JJ, corrections and reviews are made by SK. Both authors read and approved the final manuscript.

Authors' information

J. Jayabharathy received her B.Tech (CSE) from Pondicherry Engineering College, Puducherry, India and M.Tech (CSE) from Pondicherry University, Puducherry, India. She is currently working as Assistant Professor in the Department of Computer Science & Engineering at Pondicherry Engineering College. She has published nearly 15 research papers. She is currently pursuing her Ph.D in Document Mining. Her areas of interests include Data mining and Distributed Computing.

Dr. S. Kanmani received her B.E (CSE) and M.E (CSE) from Bharathiar University, Coimbatore, India and Ph.D from Anna University, Chennai, India. She is working as Professor in the Department of Information Technology at Pondicherry Engineering College. She has published nearly 63 research papers. She is currently a supervisor guiding 8 Ph.D scholars. She is an expert in Software Testing. Her areas of interests include Software Engineering, Genetic algorithms and Data Mining.

Acknowledgements

We appreciate the insightful comments from the three anonymous reviewers. Their comments were very helpful for us to improve the paper. We also express our thanks to Pondicherry Engineering College for their support in performing this research.

Author details

[1]Department of Computer Science & Engineering, Pondicherry Engineering College, Puducherry 605014 India.
[2]Department of Information Technology, Pondicherry Engineering College, Puducherry 605014 India.

Received: 11 April 2013 Accepted: 5 September 2013
Published: 19 February 2014

References

Aas, K, & Eikvil, L. (1999). *Text Categorisation: A Survey. Technical Report 941*. Oslo Norway: Norwegian Computing Center. iteseer.ist.psu.edu/aas99text.html.

Alsayed, A, Eike, S, & Saake, G. (2009). Improving XML schema matching performance using prüfer sequences. *Data Knowledge Engineering, 68*, 728–747.

Andrews, NO, & Fox, EA. (2007). *Recent developments in document clustering* (pp. 1–25). Technicalreport: Published by Citeseer.

Baghel, R, & Dhir, R. (2010). A frequent concept based document clustering algorithm. *International Journal of computer Applications, 4*(5), 0975–8887.

Banerjee, S, & Pedersen, T. (2002). Adapted lesk algorithm for word sense disambiguation using WordNet. In *Computational linguistics and intelligent text processing* (pp. 136–145). London: Springer.

Bharathi, G, & Vengatesan, D. (2012). Improving information retrieval using document clusters and semantic synonym extraction. *Journal of Theoretical and Applied Information Technology, 36*(2), 167–173.

Boris, D, Milan, V, Milos, J, & Kathrin, K. (2012). An architecture for component-based design of representative-based clustering algorithms. *Data Knowledge Engineering, 75*, 78–98.

Buckley, C, & Lewit, AF. (1985). *Optimizations of inverted vector searches, SIGIR'85* (pp. 97–110).

Chehreghani, MH, & Abolhassani, H. (2008). Improving density-based methods for hierarchical clustering of Web pages. *Data & Knowledge Engineering, 67*, 30–50.

Chen, HL, Chuang, KT, & Chen, MS. (2008). On data labeling for clustering categorical data. *IEEE Transactions On Knowledge And Data Engineering, 20*(11), 1458–1472.

Chen, CL, Tseng, FSC, & Liang, T. (2010). An integration of WordNet and fuzzy association rule mining for multi-label document clustering. *Science Direct Data & Knowledge Engineering, 69*, 1208–1226.

Chou, TC, & Chen, MC. (2008). Using incremental PLSI for threshold-resilient online event analysis. *IEEE Transactions on Knowledge And Data Engineering, 20*(3), 289–299.

Cutting, DR, Karger, DR, Pedersen, JO, & Tukey, JW. (1992). *Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections, SIGIR '92* (pp. 318–329).

Danushka, B, Yutaka, M, & Ishizuka, M. (2011). A Web search engine-based approach to measure semantic similarity between words. *IEEE Transactions on Knowledge And Data Engineering, 23*(7), 977–990.

Ellouze, N, Lammari, N, & Métaism, E. (2012). CITOM: an incremental construction of multilingual topic maps. *Data & Knowledge Engineering, 74*, 46–62.

Frakes, WB, & Fox, CJ. (2003). *Strength and Similarity of Affix Removal Stemming Algorithms* (pp. 26–30). ACMSIGIR Forum.

Gad, WK, & Kamel, MS. (2010). Incremental clustering algorithm based on phrase- semantic similarity histogram. *Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, 11*(14), 2088–2093.

Gavin, S, & Yue, X. (2009). Enhancing an incremental clustering algorithm for Web page collections. *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, 81–84.

Hammouda, KM, & Kamel, MS. (2004). Efficient phrase-based document indexing for web document clustering. *IEEE Transactions on Knowledge And Data Engineering, 16*(10), 1279–1296.

Harman, D. (1991). How effective is suffixing. *Journal of the American Society for Information Science, 42*(1), 7–15.

Hu, X, Zhang, X, Lu, C, Park, EK, & Zhou, X. (2009). *Exploiting Wikipedia as External Knowledge for Document Clustering* (pp. 389–396). France: Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'09).

Huang, A. (2008). *Similarity Measures for Text Document Clustering* (Proceedings of the New Zealand Computer Science Research Student Conference (NZCRSC'08), pp. 49–56).

Jayabharathy, J, Kanmani, S, & AyeshaaParveen, A. (2011). Document clustering and topic discovery based on semantic similarity in scientific literature. In *2nd International Conference on Data Storage and Data Engineering (DSDE 2011), 2* (pp. 425–429).

Kaiser, F, Schwarz, H, & Jakob, M. (2009). *Using Wikipedia-Based Conceptual Contexts to Calculate Document Similarity* (*Proceedings of Third International Conference on Digital Society*, IEEE, pp. 322–327).

Kowalski, G, & Maybury, MT. (2002). *Information Retrieval Systems – Theory and Implementation* (IIth ed.). Kluwer Academic Publishers. ebook ISBN: 0-306-47031-4.

Kumar, N, & Srinathan, K. (2009). *A New Approach for Clustering Variable Length Documents* (*Proceedings of the Advanced computing Conference*, IEEE, pp. 982–987).

Lam, W, & Hwuang, R. (2009). An active learning framework for semi-supervised document clustering with language modeling. *Data & Knowledge Engineering, 68*, 49–67.

Li, F, & Zhu, Q. (2011). Document clustering in research literature based on NMF and testor theory. *Journal of Software, 6*(1), 78–82.

Li, Y, Chung, SM, & Holt, JD. (2008). Text document clustering based on frequent word meaning sequences. *Journal on Data & Knowledge Engineering, 64*(1), 381–404.

Ling, C, Bhomwick, SS, & Wolfgang, J. (2009). COWES: Web user clustering based on evolutionary Web sessions. *Data & Knowledge Engineering, 68*, 867–885.

Liu, Y, Ouyang, Y, Sheng, H, & Xiong, Z. (2008). *An Incremental Algorithm for Clustering Search Results, IEEE International Conference on Signal Image Technology and Internet Based Systems* (pp. 112–117).

Lovins, JB. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics, 11*, 22–31.

Luo, C, Li, Y, & Chung, SM. (2009). Text document clustering based on neighbors. *Journal of Data Knowledge and Engineering, 68*(11), 1271–1288.

Miller, GA. (1995). WordNet: a lexical database for English, communication. *ACM, 38*(11), 39–41.

Nadig, R, Ramanand, J, & Bhattacharyya, P. (2008). *Automatic evaluation of WordNet synonyms and hypernyms*. India: Proceedings of ICON-2008, 6th International Conference on Natural Language Processing.

Ni, X, Quan, X, & Wenyin, L. (2010). Short text clustering by finding core terms. *Journal of Knowledge and Information Systems,Springer Link, 27*(3), 345–365.

Pessiot, JF, Kim, YM, Amini, MR, & Gallinari, P. (2010). Improving document clustering in a learned concept space. *Journal of Information Processing and Management, Elsevier, 26*, 182–192.

Porter, MF. ((1998). An algorithm for suffix stripping program. *14*(3), 130–137.

Prathima, Y, & Supreethi, KP. (2011). A survey paper on concept based text clustering. *International Journal of Research in IT & Management, 1*(3), 45–60.

Rooney, N, Patterson, D, Galushka, M, & Dobrynin, V. (2006). A scaleable document clustering approach for large document corpora. *Information Processing and Management, 42*, 1163–1175.

Salton, G, & Buckley, C. (1998). Term-weighting approaches in automatic text retrieval. *Information Processing & Management, 24*(5), 513–523.

Sammut, C, & Webb, G. (2010). *Encyclopedia of machine learning: Springer reference* (Ith ed.). ISBN 978-0-387-34558-1.

Shehata, S. (2009). *AWordNet-based Semantic Model for Enhancing Text Clustering. IEEE International Conference on Data Mining Workshops* (pp. 477–482). 6 Dec. 2009.

Shehata, S. (2010). An efficient concept-based mining model for enhancing text clustering. *Journal of Knowledge and Data Engineering, 22*(10), 1360–1371.

Shehata, S, Fakhri, K, & Mohamed S, S. (2010). An efficient concept-based mining model for enhancing text clustering. *IEEE Transactions On Knowledge And Data Engineering, 22*(10), 1360–137.

Steinbach, M, Karypis, G, & Kumar, V. (2000). *A Comparison of Document Clustering Techniques* (pp. 1–2). International Conference on Data Mining: Knowledge Discovery and Data Mining (KDD) Workshop on Text Mining.

Tang, B, Shepherd, M, Milios, E, & Heywood, MI. (2005). *Comparing and Combining Dimension Reduction Techniques for Efficient TextClustering* (Proceedings of Canadian Conference on AI, pp. 292–296).

Van Rijsbergen, CJ. (1989). *Information Retrieval* (Secondth ed.). London: Buttersworth.

Wang, X, Tang, J, & Liu, H. (2011). Document clustering via matrix representation. In *11th IEEE International Conference on Data Mining ICDM 2011* (pp. 804–813).

Xiaoke, S, Yang, L, Renxia, W, & Yuming, Q. (2009). *A Fast Incremental Clustering algorithm* (*Proceedings of the 2009 International Symposium on Information processing (ISIP'09*), pp. 17–178). Academy Publisher.

Yan, J, Liu, N, Yan, S, Yang, Q, Fan, WP, Wei, W, & Chen, Z. (2011). Trace-oriented feature analysis for large-scale text data dimension reduction. *IEEE Transactions on Knowledge and Data Engineering, 23*(7), 1103–1117.

Zamir, O, Etzioni, O, Madani, O, & Karp, RM. (1997). *Fast and Intuitive Clustering of Web Documents* (*Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97*), pp. 287–290).

Zaniolo, C, & Wang, F. (2008). Temporal queries and version management in XML-based document archives. *Dataand Knowledge Engineering, 65*(04–324), 2008.

Zhang, T, Member, YY, Tang, BF, & Xiang, Y. (2011). Document clustering in correlation similarity measure space. *IEEE Transactions on Knowledge And Data Engineering, 24*(6), 1002–1013.

doi:10.1186/2193-8636-1-3
Cite this article as: Jayabharathy and Kanmani: **Correlated concept based dynamic document clustering algorithms for newsgroups and scientific literature.** *Decision Analytics* 2014 **1**:3.