Investigate the Performance of WordNet and Association Rules for Hard Clustering Web Document

Noha Negm
Faculty of Science, Menoufia University, Shebin El-Kom, Egypt
Faculty of Science and Arts, King Khalid University, Saudi Arabia

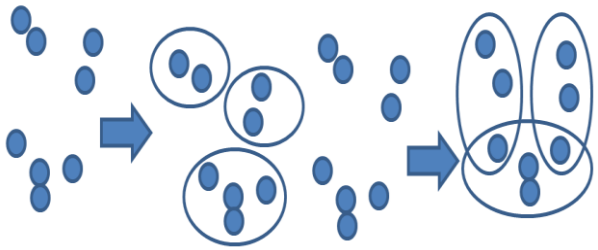
Hany Mahgoub
Faculty of Computers and Information, Menoufia University, Shebin El-Kom, Egypt
Faculty of Science and Arts, King Khalid University, Saudi Arabia

ABSTRACT
A powerful technique that has been widely used to organizing a large number of web documents into a small number of general and meaningful clusters is Document Clustering. High dimensionality, scalability, accuracy, extracting semantics relations from texts and meaningful cluster labels are the major challenges for document clustering. To improve the document clustering quality, we intend to introduce an effective methodological system using association rules instead of frequent term sets for clustering web documents into different topical groups called Hard Document Clustering using Association Rules (HDCAR). HDCAR characterized by high performance in the organization of web documents and navigates them effectively in order to keep up with the explosive growth of the number and size of web documents. Association Rule has the equally important advantage of having a higher descriptive power compared to single words (frequent term sets). Moreover, the external knowledge from both WordNet synonym and hypernyms will be used to enhance the ‘‘bag of words’’ used before the clustering process and to assist the label generation procedure following the clustering process. Then, Multi-Hash Tire Association Rule (MHTAR) algorithm is used to discover a set of highly-related association rules to overcome the drawbacks of the Apriori algorithm. Through the resulted association rules, the hidden topics are discovered as the first step and then the documents will be cluster based on them. Finally, each document is assigned to only one cluster (hard clustering) with the highest Document Weighted-measure, and then the highly similar clusters are merged. To evaluate the performance of HDCAR, we conducted experiments based on four different kinds of datasets Classic, Re0, WebKB and REUTER datasets. The experimental results show that HDCAR outperforms the major document clustering methods like k-means, Bisecting k-means, FIHC, and UPGMA with higher accuracy quality, efficiency and lower execution time. Furthermore, HDCAR provides more general and meaningful labels for documents and increases the documents clusterization speed, as a result of the reduction of their dimensionality.

Keywords
Web Mining, Document Clustering, Association rule mining, WordNet, Fuzzy weighting score.

1. INTRODUCTION
The rapidly increasing of the number of online information sources makes users suffer from the information overloading problem and makes information retrieval a tedious process for the average user. Document clustering considers an effective tool for managing information overload and organizing documents into meaningful clusters such that documents within a cluster are more similar to each other than documents belonging to different clusters. No labeled documents are provided in document clustering unlike classification, so clustering is known as unsupervised learning. Document clustering plays an important role in Document Organization [1] Summarization [2], and Topic Extraction [3]. It is used in many applications including web Information Retrieval [4], Natural Language Processing [5], Bioinformatics, and Technology analysis [6]. The document clustering problem is defined as follows: given a set of documents, we would like to partition them into a predetermined such that each cluster contains the documents that are more similar to each other than the documents assigned to different clusters. In other words, any cluster contains the documents that share the same topic, and the documents in different clusters represent different topics [7]. High dimensionality, scalability, accuracy, extracting semantics relations from texts and meaningful cluster labels are the major challenges that clustering techniques normally have to overcome [8-10].

According to [11], document clustering can be performed in two different modes as in Figure 1: Hard (Disjoint) or Soft (Overlapping) clustering. In hard clustering, each document is assigned to exactly one cluster and their algorithms generate a set of disjoint and non-overlapping clusters depending on the hard assignment. In soft clustering, each document allows appearing in multiple clusters and their algorithms compute the soft assignment and produce a set of overlapping clusters.



Hard Clustering Soft Clustering

Fig 1: Two different modes for document clustering

The existing clustering algorithms are classified into two generic categories: Hierarchical and Partitioning algorithms [12-16]. Typically partitioning algorithms partition the set of documents into a number of disjoint clusters by moving documents from one cluster to another. Moreover, partitioning algorithms can be used as divisive algorithms in the hierarchical clustering. An integer number of partitions that optimize a certain criterion function can be determined by the partitioning algorithms. The *k*-means is a commonly used algorithm in Partitioning algorithms and it is based on the idea that a centroid can represent a cluster. A distance measure is used to assign each document to a cluster after selecting *k*

centroids after those k centroids are recalculated. This step is repeated until an optimal set of k clusters are obtained [17]. On the other hand, hierarchical algorithms cluster a collection of documents into a hierarchical tree structure whose leaf nodes represent the subset of a document collection that facilitates browsing. In hierarchical algorithms, a series of partitions can be generated over the data, which may run from a single cluster containing all objects to n clusters each containing a single object. They are widely visualized through a divisive (Root to Leaves) or agglomerative (Leaves to Root) tree structure. Different hierarchical algorithms for text documents have been discussed in [18, 19].

Today, most of the existing documents clustering algorithms use the Vector Space Model (VSM) as data representation model for documents [20], which treats a document as a bag of words. Once terms are treated as individual items in VSM representation, the semantic content of a document is decomposed and cannot be reflected. WordNet has been widely used recently as ontology in grouping documents with its semantic relations of terms since it considers one of the largest lexical databases of English [21]. In WordNet, nouns, verbs, adjectives, and adverbs are grouped into sets of synonyms called synsets. These synsets are organized into senses that give the synonyms of each word, and also into hypernyms relationships that provide a hierarchical tree-like structure for each term [22]. We intend to utilize both synonym and hypernyms from WordNet to improve the performance of the document clustering by generating more general and conceptual labels for resulted clusters.

High-dimensionality of the feature space is a major characteristic of VSM representation and it imposes a big challenge to the performance of document clustering algorithms due to the inherent sparseness of the data [23]. To resolve the problem of high dimensionality, a new category of document clustering has been developed. It uses the concept of frequent Itemset for the document clustering and called "Frequent Itemset-based Clustering," [24]. This method reduces the dimensionality of term features efficiently for very large datasets by using only the frequent itemsets generated by the Association Rule Mining. Association Rule Mining is one of the successful data and text mining techniques for discovering meaningful association rules to represent a relationship between the most frequent itemsets [25]. The association rule form can be represented as $X \rightarrow Y$, where X and Y are sets of items and $X \cap Y = \Phi$. Support"s"and Confidence "c" are two important basic measures used for association rules. The rule $X \rightarrow Y$ has support s in the collection of documents D if s% of documents in D contain $X \cup Y$ The support is calculated y

$$Support\,(XY) = Support count of\ XY \,/\, Total number of\ documents D$$

The rule $X \rightarrow Y$ holds in the collection of documents D with confidence c if, among those documents that contain X, c% of them contain Y also. The confidence is calculated $Confidence\,(X \setminus Y) = Support\,(XY) \,/\, Support\,(X)$ .The association rules mining problems are defined as two independent sub-problems. First, generate all combinations of items whose support is greater than the user-specified minimum support [25]. This combination is called large frequent itemsets. Second, generate all rules from determined large frequent itemsets that satisfy a user specified minimum confidence. The frequent itemsets generation requires more effort while the rule generation is straightforward.

Frequent itemsets are the set of items that is co-occurring in more than a threshold percentage of all documents of a collection called support threshold. Apriori algorithm considers the basic algorithm for all developed association rule mining algorithms for generating frequent itemsets. Apriori still suffers from generating huge numbers of candidates and taking many scans of large databases for frequency checking while it achieves good reduction on the size of the candidate set. Although the drawbacks of Apriori algorithm, it still used up to now for generating frequent itemsets that used in the document clustering [24]. These extracted frequent itemsets are used for clustering the documents and for labeling the obtained clusters [26]. The structure of the resulted clusters could be in a hierarchical tree or in a flat set. In text mining, recent studies on frequent term sets fall into two categories. One is to use Association Rules to conduct text categorization [27] and the other one is to use frequent term sets for text clustering [28,29]. Although the Association rule has the equally important advantage of having a higher descriptive power compared to single words (frequent term sets), a little of work has been done using it for solving the problem of finding clusters of similar items for instance, in market-basket type data, a practical application of association rules is to identify clusters of similar items based on customer sales information. This aids to group items based on customer interests in addition to understand patterns in sales of items. From a given database, Association rule mining can find out association rules that satisfy the predefined minimum support and confidence.

The concept of Association Rule is not used in document clustering process although it has an important role. Since the main idea of the association rule-based clustering stage is based on a simple observation: the documents under the same topic should share a set of common keywords. Some minimum fraction of documents in the document set must contain these common keywords, and they correspond to the notion of frequent term sets which form the basis of the association rules. An essential property of association rules is its representation of the relations between words that commonly occur together in documents. To explain that the property of association rules is important for clustering, we consider two frequent terms, "apple" and "window". The documents that contain the word "apple" may discuss fruits or farming. While the documents that contain the word "window" may discuss renovation. However, if we found association rules between both words occur together in many documents, then we may identify another topic that discusses operating systems or computers. We can improve the accuracy of the clustering solution by accurately identifying these hidden topics as the first step and then clustering documents based on them.

In this paper, we will present HDCAR system that combines association rule mining to provide significant dimensionality reduction over interesting frequent itemsets with WordNet for clustering web documents. Moreover, we used an efficient Multi-Hash Tire Association Rule Mining algorithm to improve the mining process and to overcome all drawbacks of the Apriori algorithm. Through the resulted association rules, the hidden topics are discovered as the first step and then clustering documents based on them. This paper illustrates the effect of using HDCAR mathematical formula called Document Weighting-measure (DW-measure) to improve the accuracy by removing the overlapping between document clusters. The performance of HDCAR was experimentally checked by using a large pool of the four benchmarks dataset. The obtained results are analyzed and compared with these, obtained by using k-means, Bisecting k-means, FIHC and UPGMA (Frequent Itemset Hierarchical Clustering). The

analysis confirms the efficiency of the HDCAR system, their higher accuracy quality and lower execution time, and also their ability for further development and perfection.

The paper sections are organized as follows: Section 2 gives a concise review of the related work regarding frequent itemset-based clustering methodologies as well as the use of the WordNet database on this field. In Section 3, HDCAR document clustering system based on Association Rules is described in detail. In Section 4, we outline our experimental approach towards the document clustering methodologies used and present our evaluation results. Section 5 concludes this paper and we describe our future work that is currently underway.

2. RELATED WORK

High-quality document clustering algorithms play an important role in helping users to get relevant information, navigate, summarize and organize an enormous amount of documents available on the internet, news sources and in digital libraries. In order to solve the problems of high dimensionality, scalability and accuracy, a huge variety of techniques have been proposed especially for document clustering. This section presents most of the previous works related to clustering web documents based on frequent itemsets and the using of the WordNet database on this field.

Frequent itemset-based clustering method has been extensively developed to reduce the dimensionality of term features efficiently for very large datasets using frequent itemsets. According to these generated frequent itemsets, the documents will be a cluster. The first frequent itemsets-based algorithm, namely Hierarchical Frequent Term-based Clustering (HFTC) was developed by [24], where the frequent itemsets are generated based on the association rule mining [30]. Although the HFTC method minimized the overlap of clusters in terms of shared documents, it is not scalable for large document collections. In [31], the FIHC (Frequent Itemset-based Hierarchical Clustering) algorithm proposed to construct a hierarchical topic tree for clusters using frequent itemsets. Based on the global frequent items in document vectors, FIHC reduced the dimensionality of term features effectively. FIHC is not only scalable and non-overlapping algorithm but also accurate. To improve the clustering quality and scalability, in [32], another frequent itemset based algorithm, called TDC presented. Based on the closed frequent itemsets, TDC algorithm reduced the dimensionality and generated a topic directory from a document set. However, the clusters generated by TDC algorithms were non-overlapping. In [28], the two FTSC and FTSHC algorithms are introduced. FTSC algorithm reduced the dimension of the text data efficiently for very large databases, thus the accuracy and speed of the clustering algorithm are improved. The overlapped of texts' classes cannot be reflected by using the FTSC algorithm for clustering texts. An improved algorithm—Frequent Term Set-based Hierarchical clustering algorithm (FTSHC) is given based on the FTSC algorithm. This algorithm determined the overlap of texts' classes by the overlap of the frequent words sets and provided an understandable description of the discovered clusters by the frequent terms sets. In [23], a new text clustering algorithm proposed, named TCFS (Text Clustering with Feature Selection). During the clustering process, TCFS performed a supervised feature selection moreover the cluster label information was utilized as the known class label information for the feature selection. The using of selected features improved the quality of clustering iteratively, and the clustering result has higher accuracy. In [26], an effective Fuzzy Frequent Itemset-Based Hierarchical Clustering (F2IHC) approach proposed, which used fuzzy association rule mining algorithm to improve the clustering accuracy of Frequent Itemset Based Hierarchical Clustering (FIHC) method. The using of fuzzy association rule mining discovered important candidate clusters to increase the accuracy quality of document clustering. Therefore, it is worth extending in reality for concentrating on huge text documents management.

In document clustering, there are only a few methods proposed to utilize the semantic relationships between words to improve the clustering quality [33-39]. Some of the most common ontologies used to enhance the document representation include WordNet, Mesh, etc. Unlike the traditional vector space model, Li, Chung in [40] proposed a new document clustering algorithms based on the sequential patterns of the words in the document. The two algorithms named CFWS and CFWMS, which stands for clustering based on frequent word sequences and frequent word meaning sequences, respectively. CFWS algorithm explored unique characteristics of text documents to reduce the high dimension of the documents and to measure the closeness between them. The performance of CFWS algorithm was quite scalable. In CFWMS, the synonyms and hyponyms/hypernyms provided by the WordNet ontology were used to generate all frequent word meaning sequences. CFWMS has a better accuracy than CFWS since frequent word meaning sequences can capture the topics of documents more precisely than frequent word sequences. In [41], a hierarchical clustering algorithm using closed frequent itemsets proposed that used Wikipedia as an external knowledge to enhance the document representation. It handled high dimensional data and achieved compact clustering using concepts from generalized frequent itemsets. Moreover, it provided meaningful labels to the clusters and higher accuracy. In [36], an effective approach that combined fuzzy association rule mining with an existing ontology WordNet (FMDC) proposed. WordNet is utilized to enrich the document representation to find semantically related documents. Fuzzy data mining algorithm was applied on the structured document term vectors to generate fuzzy frequent itemsets and output a candidate cluster [42]. Furthermore, each document was assigned to multiple clusters by producing a Document-Cluster matrix (DCM) to represent the degree of importance of a document to a candidate cluster.

The clustering accuracy of this approach was improved. The drawback of this approach is that fuzzy association mining and the initial clustering stages are the two most time-consuming tasks, something that leads to high execution times in order to get the required cluster labels. In contrast, we are focusing on an effective document clustering system that will generate clusters as well as their more informative labels reasonably fast. Figure 2 presents a tree structure for our work.

3. THE FRAMEWORK OF HDCAR

The feature of HDCAR is performing two various tasks concerning web mining for documents that originate from the Web. The scope of HDCAR is the non-overlapping clustering of web documents based on hidden topics discovered from association rule mining. It is delivering more informative and faster document clustering to end users that do not have time to keep up with the explosive growth of the number and size of web documents. HDCAR consists of three main phases as shown in Figure 3. We explain to them as follows:

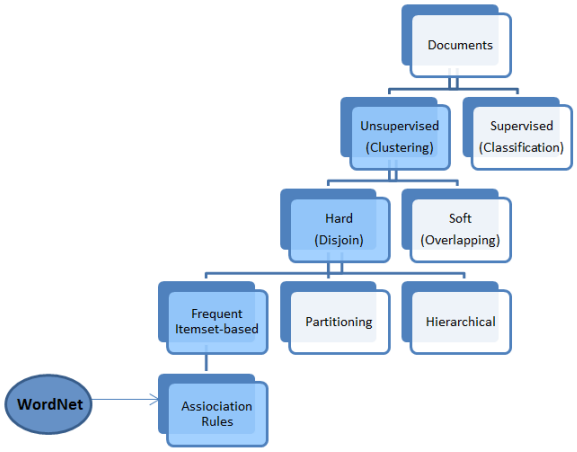


Fig 2: A tree structure for our work

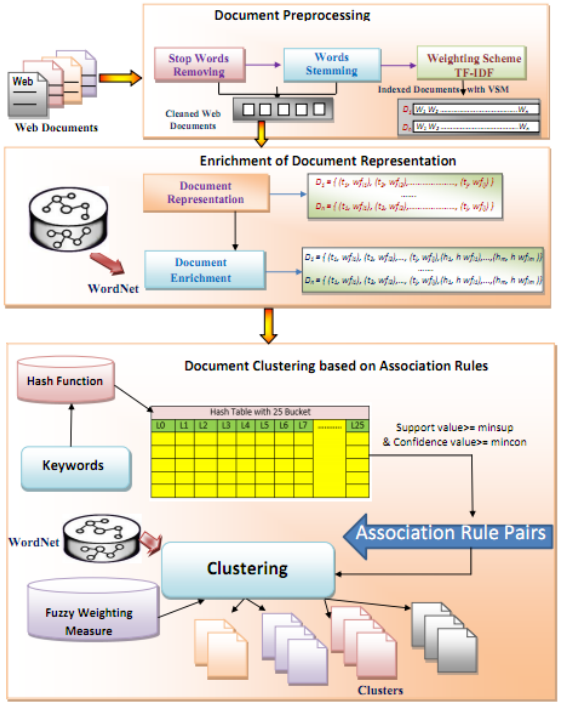


Fig 3: HDCAR System

3.1 Document Preprocessing

Please To generate the term set from the web document collection, we first divide the sentences into terms and extract the terms features. A term is regarded as the stem of a single word in this paper. Afterward applied a pre-defined stop word list to remove the non-information bearing words from the documents and reduce noise. Removing the stop words affords similar advantages: Firstly it could save a huge amount of space. Secondly, it helps to reduce the noises and keep the core words, and it will make later processing more effective and efficient. Next, the developed stemming algorithms, such as Porter Stemmer can be used to convert a word to its stem or root form. Moreover, it can reduce the number of index terms, save memory and may increase the performance of clustering algorithms to some extent. The terms with the same stem are combined for frequency counting. Finally, the frequency of each term in each document is recorded. After that, we apply the weighting schema (*TF-IDF*) as the feature selection method to reduce the set of term features and to measure the importance of a term within a document. Formula (1) is the used weighting schema whereas w(i,j) 0 :

$$w(i,j) = tfidf(d_i, t_j) = \begin{cases} Nd_i, t_j * \log_2 \frac{|D|}{Nt_j} & if\ Nd_i, t_j \geq 1 \\ 0 & if\ Nd_i, t_j = \end{cases} \quad (1)$$

In Formula (1), Nd_i,t_j is the number the term t_j occurs in the document d_i, Nt_j is the number of documents in collection D in which t_j occurs at least once (document frequency of the term t_j) and $|D|$ is the number of the documents in collection D. The first clause applies for words occurring in the document, whereas for words that do not appear (Nd_i,t_j =0), we set $w(i,j)$=0. A weighting schema weighs the keywords of each document based on their frequencies in the document such that the terms will be discarded if its weight is less than the minimum weighing threshold value γ. The terms of high weights form a set of key terms for the document set. Figure 4 represents the document preprocessing algorithm.

3.2 ENRICHMENT OF DOCUMENT REPRESENTATION

In this phase, we intend to use both Synonyms and Hypernyms provided by WordNet as useful features for document clustering. WordNet is the large online lexical database of English developed by [21] to include all background information on each word. Containing over 150,000 terms; nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms called synsets. The synsets are organized into Senses; giving thus the synonyms of each word, and also Hyponym/Hypernym (i.e Is-A), and Meronym/holonym (i.e Part-of) relationships, providing a hierarchical tree-like structure for each term. The applications of WordNet have been an association with clustering techniques, all these approached to come to the conclusion that noise is degrading their clustering. Compared to these approaches; the use of both Synonym and hypernyms can prevent this problem and will improve the efficiency of the applied clustering algorithm.

Algorithm 1. Document Preprocessing Algorithm

Input: A document set *D*; A pre-defined stop word list; minimum TF-IDF threshold γ.

Output: The set of significant and important key terms of each document K_D.

Divide the sentences into terms and extract the terms features

T_D ={ t_1,t_2,…….,t_j,……..t_t}

Remove all stop words from T_D

Apply Porter Stemmer algorithm for T_D

For each $d_i \in D$ *do*

each $t_j \in T_D$ *do*

$f(d_i,t_j) = Nd_i,t_j * \log_2 \frac{|D|}{Nt_j}$

If $tfidf(d_i,t_j) \geq \gamma$ Then

$= t_j$

Form a set of key terms for the document set

={ t_1,t_2,…….,t_j,……..t_t}, where $t \leq i$

Fig 4: Document Preprocessing Algorithm

After key terms are extracted from the document set, all key terms that denote the same concept and are interchangeable in many contexts grouped into unordered synonyms sets. In WordNet, each synset contains a brief definition and in most cases one or more short sentences illustrating the use of the synset member. Keyterm forms with several distinct meanings are represented in as many distinct synsets. Thus, each form-meaning pair in WordNet is unique.

After the synonym process, we can search the WordNet database for all the hypernyms of a resulted set of key terms. In this stage, each document d_i in D is represented using those terms in $KD = \{t_1, t_2, \ldots, t_p, p_1, \ldots, p_d\}$, where p_j is a hypernyms. Thus, each document $d_i \in D$, denoted $d_i = \{(t_1, fi_1), (t_2, fi_2), \ldots, (t_j, fi_j), \ldots, (t_h, fi_h)\}$, is represented by a set of pairs (*term, frequency*), where the frequency fi_j represents the frequency of the key term t_j in d_i.

In order to decrease the noise from hypernyms, the fuzzy weighting schema [43] is executed to weigh them and finally chose representative hypernyms that seem to extend the overall meaning of the set of given key terms. The fuzzy weighting is a developed from the mathematical formula weighting schema to increase the accuracy rate of the selected key terms from the documents that will make the clustering result more accurate. We define the fuzzy membership value in equation (2) as follows:

$$\mu_{i,j} = \left\{ \frac{Nt_j}{|D|} \quad where \quad 0 \le \mu \le 1 \right. \qquad (2)$$

Therefore, from equation (1) the Fuzzy Weighting Schema is defined as follows:

$$Fuzzy.w(i,j) = \mu_{i,j} * tfidf(d_i, t_j)$$

$$Fuzzy \cdot w(i,j) = \mu_{i,j} * \begin{cases} Nd_i,t_j * \log_2 \frac{|D|}{Nt_j} & if\ Nd_i,t_j \ge 1 \\ 0 & if\ Nd_i,t_j = 0 \end{cases} \qquad (3)$$

For each keyword in all documents, its new fuzzy *TF-IDF* value is calculated and sorted the list of these values in descending order. Since all high weighted values were given to the key terms that are more occurrences in the documents. Based on the existing of the less important key terms in the bottom of the list, the system automatically eliminates 10% of these key terms. After that, the system stores all key terms without redundancy with their frequencies for using them as input to the mining process. Figure 5 represents the enrichment document representation algorithm. The reason for using synonyms and hypernyms of WordNet is to reveal hidden similarities to identify related topics, which potentially leads to better clustering quality.

3.3 Document Clustering

The final stage is to group the documents into clusters. In the following, we used Multi-Hash Tire Association Rule (MHTAR) Algorithm [44] for text to generate all frequent term sets after that generate all strong Association Rules. Based on the mining results, we illustrate the details of the clustering process.

3.3.1 Multi-hash tire association rule algorithm for text

The main characteristic of MHTAR algorithm for text is minimizing the I/O, where it uses a new methodology for generating frequent term sets by building the hash table during scanning the documents only once consequently; the number of documents scans is decreased. Moreover, it shows better performance in terms of time taken to generate frequent term sets. MHTAR algorithm generates frequent term sets based on the building and scanning process on the dynamic hash table and the minimum support s. To avoid the collision in the building process of the dynamic hash table, we build at first fixed number of a primary bucket array equals to the number of the English alphabet and give each cell a unique character from "*A*" to "*Z*". The Division method of the hash function is used to determine the location of each cell in the table. For each document di, all terms and term sets are inserted in a hash table and their frequencies are updated. The process continues until there is no document in the collection *D*. The MHTAR algorithm permits the end user to insert different minimum support values to determine the large frequent term sets without re-scanning the original documents again. Figure 6 shows the Multi-Hash Tire Association Rule Algorithm for Text.

Algorithm 2. Enrichment Document Representation Algorithm

Input: A document set *D*; WordNet; The set of key terms of each document K_D; frequency of the keyterm t_j in d_i.

Output: Enriched list of key terms.

1. For each $d_i \in D$ *do*

For each $t_j \in K_D$ *do*

If (t_j *has a synonym* t_m) *Then*

$t_m \rightarrow t_j$

$(t_m, f_{i,m}) \rightarrow (t_j, f_{i,j} + f_{i,m})$

$d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \ldots, (t_j, f_{ij}), \ldots, (t_p, f_{ip})\}$

2. For each $d_i \in D$ *do*

For each $t_j \in K_D$ *do*

If (p_i *is hypernyms of* t_j) *Then*

$pf_{ij} \rightarrow pf_{ij} + f_{ij}$

$K_D \rightarrow K_D \cup \{p_i\}$

3. For each $d_i \in D$ *do*

For each $t_j \in K_D$ *do*

$$tfidf(d_i, t_j) = Nd_i,t_j * \log_2 \frac{|D|}{Nt_j}$$

$$\mu_{i,j} = \frac{Nt_j}{|D|}$$

$$Fuzzy.tfidf(d_i, t_j) = \mu_{i,j} * tfidf(d_i, t_j)$$

4. *Sort all Fuzzy.* $tfidf_{ij}$ *weight into descending order*

5. *Eliminate 10% of weights from the bottom of the list.*

6. *Form the new keyterm sets*

$K_D = \{t_1, t_2, \ldots\ldots, t_m, p_1, p_2, \ldots\ldots p_n\}$

7. *For each* $d_i \in D$ *do*

$d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \ldots, (t_m, f_{im}), (p_1, pf_{i1}), \ldots\ldots\ldots\ldots, (p_n, hf_{in})\}$

Fig 5: Enrichment Document Representation Algorithm

3.3.2 Picking out all strong association rules

In our work, we use association rules as our information source to improve document clustering performance. Association rules are useful in constructing accurate descriptions of clusters than frequent term sets. Moreover, the

problem of finding clusters of similar items can be solved by using association rules. Once the frequent term sets from documents have been generated from the previous step, it is straightforward to generate all strong association rules from them as in figure 7 and figure 8. From each large frequent term sets at each level, a number of association rules can be generated which often results in very large association rules. The critical factors for generating association rules are the minimum support and confidence threshold. Since a low support threshold and high confidence threshold result in too many and more useful discovered associations. Increasing the support threshold significantly reduced the number of rules discovered but risks losing useful associations.

All generated association rules that satisfy the confidence threshold is used as input to the document clustering process. The MHTAR algorithm has the ability to generate different sets of association rules with different threshold confidence without the need of re-doing the mining process. The main advantages are improving and speeding up the clustering process and saving the execution time. In this work, we expand the generated association rules set Rk to include the association rules generated from the set of 2 and 3 large frequent term sets since

$$H_1 = \{\ t_i \rightarrow t_j\ : t_i, t_j \in f_k\ \}$$

$$H_2 = \{\ t_p, t_s \rightarrow t_q\ : t_p, t_s, t_q \in f_k\}$$

$$R = \{H_1 \cup H_2\}$$

Algorithm 3 Multi-Hash Tire Association Rule Algorithm for Text

T_m: Set of all term sets for each document d

C_m: Candidate term sets for each document d

I_k : Frequent term sets of size k.

All Text documents; Minimum Support; Division hash function.

: Building Multi-Tire Hash Table and Finding the frequent term sets

nerating all frequent term sets.

For each document $d_m \in D$ do

$T_m = \{\ t_i : t_i \in d_m\ ,\ 1 \le i \le n\ \}$

For each term $t_i \in T_m$ do

$h(t_i) = t_i \bmod N;$

$t_i.count{+}{+};$

C_k = all combinations of $t_i \in d_m$

$C_m = subset\ (C_k\ ,\ d_m\ );$

For each candidate $c_j \in C_m$ do

$h(c_j) = c_j \bmod N;$

$c_j.count{+}{+};$

For given s= minsup in hash table do

$I_1 = \{\ t\ |\ t.count \ge minsup\ \}$

$I_k = \{\ c\ |\ c.count \ge minsup,\ k \ge 2\}$

Fig 6: Multi-Hash Tire Association Rule Algorithm for Text

Algorithm 4 Generating strong association rules

he set of all frequent term sets *F*; Minimum Support *S*; Minimum Confidence; the total number of documents *n*.

nerating all strong association rules.

For each frequent *k*-term sets f_k in *F*, $k \ge 2$ do

If *confidence of* $t_i \rightarrow t_j \ge minconf$ and

Support $(t_i, t_j) \leftarrow f_k.count / n$ then

$H_1 = \{\ t_i \rightarrow t_i : t_i, t_i \in f_k,\ 1 \le i, j \le n\ \}$ // *1-term consequent rule of* f_k

Ap-genRules (f_k , H_1)

Fig 7: Generating strong association rules algorithm

Procedure**:** Ap-genRules (f_k , H_1)

1.If ($k > m$ +1) and ($H_m \neq \emptyset$) then // *H_m is the set of m-term consequents*

$H_{m+1} \leftarrow$ candidate-gen (H_m);

2.For each h_{m+1} in H_{m+1} do

$conf \leftarrow f_k.count\ /\ (f_k - h_{m+1}).\ count;$

if ($conf \ge minconf$) then

output the rule $(f_k - h_{m+1}) \rightarrow h_{m+1}$ with confidence = *conf* and *support* = $f_k.count\ /n;$

else

delete h_{m+1} from H_{m+1};

3.Ap-genRules (f_k , H_{m+1}) ;

Fig 8: Ap-genRules algorithm

3.3.3 Clustering process

For assigning documents to the target clusters, all strong association rules are generated according to the minimum confidence value from a large target textual document set as in Figure 9. Initially, the association rules set are sorted in descending order in accordance with their confidence level as:

$$Conf(R_1) > Conf(R_2) > \ldots\ldots\ldots\ldots > Conf(R_k)$$

An initial partition P_1 is constructed for first association rule in R_k. Afterward, all the documents containing both term sets that constructed the rules are included in the same partition. Next, to form a new partition P_2, we take the second association rules whose confidence is less than the previous one. This partition is formed in the same way of partition P_1. This procedure is repeated until every association rule is moved into partition P_k since

$$P_k = < R_k\ ,\ doc\ [\ R_k] >$$

The benefit of initial partitions is to ensure that all the documents in a cluster contain all the terms in the association rules that already defines the partition. Moreover, these rules can be considered as the mandatory identifiers for every document in the partition. To identify the partition, these association rules are used as the partition label to facilitate browsing for the user. To reduce the numbers of resulted partitions, all partitions that contain the similar documents are merged into one partition.

Since a document usually contains more than one frequent term set, the same document may appear in multiple initial partitions, i.e., initial partitions are overlapping. To assign each document to the optimal partition we developed Fuzzy Weighted Score ($P_i \leftarrow doc_j$) in equation (4) to belong each document to exactly one partition.

$$(P_i \leftarrow doc_j) = \sum_k fw_k * m_i / n_w \quad (4)$$

Where $\sum_k fw_k$ represents the sum of fuzzy weighted values of all words constructed the association rules from doc_j, m_i represents the number of documents in the initial partition P_i, and n_w represents the number of words that construct the partition P_i from doc_j. The fuzzy weighted values of words w_k are defined by the Fuzzy (*TF-IDF*) in the enrichment of document representation process. The Weighted Score measure used the Fuzzy weighed values of frequent term sets instead of the number of occurrences of the terms in a document. It caused a strong effect since high weighted values were given only to the key terms that are more occurrences in a document. Moreover, it caused to appear new key terms with high fuzzy weighted values although they are not appeared using the weighing schema. To make partitions non-overlapping, we assign each doc_j to the initial partition P_i of the highest score. After this assignment, if there is more than one P_i that maximizes the Fuzzy Weighted Score $(P_i \leftarrow doc_j)$, we will choose the one that has the most number of words in the partition label. Each document belongs to exactly one partition after this step.

After removing the overlapping and put each document in its optimal partition, we begin to cluster documents based on the partition labels. In this step, we don't require to pre-specify the number of clusters as the previous standard clustering algorithms. We have a set of non-overlapping partitions $P_{(i)}$ and each partition has a number of documents $D_{p(i)}$. We first identify the association rules that construct each partition. The set of all words that construct all association rules in $P_{(i)}$ are called the *labelling Words* $Ld_{w_{(i)}}$. Moreover, all the words in the partition label must be contained for every document in the partition since the partition label is used to identify the partition. We observed that the partition labelling words based on association rules are more informative than other based on frequent term sets. However the number of association rules is always greater than the number of frequent term sets, the rules carry out more information. Identifying hidden knowledge from documents can help us for improving the accuracy of the clustering process. The similarity measure plays significant role in obtaining effective and meaningful clusters. For each document $D_{p(i)}$ in partition $P_{(i)}$, to compute its similarity measure we must obtain the *Derived keywords* $Vd_{w_{(i)}}$ from taking into account the difference between the top fuzzy weighted frequent words for each document with the labeling words. Afterwards, within each partition, the total support of each derived word is computed. The set of words satisfying the partition threshold (*the percentage of the documents in partition* $P_{(i)}$ *that contains the term set*) are formed as *Descriptive Words* $Pw_{p_{(i)}}$ of the partition P_i.

$$Pw_{p_{(i)}} = \{x : P(x)\}$$
$$P(x) = \left[Vd_{w_{(i)}}\left[D^{(x)}_{P(i)}\right]\right] \geq P\_sup \qquad (5)$$

Afterward, the similarity of each document in the partitions is computed with respect to the descriptive words. We compute the similarity between two documents S_m as:

$$S\left(Vd_{w_{(i)}}\left[D^{(x)}_{P(i)}\right], Pw_{p_{(i)}}\right) = \left|Vd_{w_{(i)}}\left[D^{(x)}_{P(i)}\right] \cap Pw_{p_{(i)}}\right|$$
$$S_m\left(Vd_{w_{(i)}}\left[D^{(x)}_{P(i)}\right], Pw_{p_{(i)}}\right) = \frac{S\left(Vd_{w_{(i)}}\left[D^{(x)}_{P(i)}\right], Pw_{p_{(i)}}\right)}{\left|Pw_{p_{(i)}}\right|}$$
$$S_m\left(Vd_{w_{(i)}}\left[D^{(x)}_{P(i)}\right], Pw_{p_{(i)}}\right) \geq 0.5$$

A new cluster is formed from the partitions based on the similarity measure, i.e. each cluster will contain all partitions that have similar similarity measures or greater than the threshold value 0.5.

Algorithm 5 Association rule-based Clustering Algorithm for Text

Input: A Document set *D*; the frequent termset set *F*; minimum support; minimum confidence, minimum partition support.

Output: The target cluster set.

1) For each frequent *k*-term sets $f_k \in F$, ($k = 2$) do

If *confidence of* $t_i \text{ -> } t_j \geq minconf$ and

Support $(t_i, t_j) \leftarrow f_k.count / n$ then

$H_1 = \{ t_i \text{ -> } t_j : t_i, t_j \in f_k,\ 1 \leq i, j \leq n \}$

2) For each frequent *k*-term sets $f_k \in F$, ($k = 3$) do

If *confidence of* $t_p, t_s \rightarrow t_q \geq minconf$ and

Support $(t_p, t_s \rightarrow t_q) \leftarrow f_k.count / n$ then

$H_2 = \{ t_p, t_s \rightarrow t_q : t_p, t_s, t_q \in f_k,\ 1 \leq p, s, q \leq n \}$

3) $R = \{H_1 \cup H_2\}$

4) Sort each rule $h_i \in R$ in descending order such that

$Conf(h_1) > Conf(h_2) > Conf(h_k)$

For (i=1; 1< i < n ; i++) do

$P_i = < h_i$, doc[h_i] > , $h_i \in R$

If $conf(h_i) > conf(h_{i+1})$ *then go to 5*

5) For (i=1; 1< i < n ; i++) do

For (j=1; 1< j < m ; j++) do

If $(P_i \leftarrow doc_{(i)}) = (P_j \leftarrow doc_{(j)})$ *then*

$(P_i \leftarrow P_j)$

6) For (j=1; 1< i < m ; j++) do

For (i=1; 1< j < n ; i++) do

For each $doc_{(j)} \in P_i$ *do*

For each term $t_j \in doc_{(j)}$

Fuzzy Weighted score $= \sum_k fw_k * m_i / n_w$

$(P_i \leftarrow doc_{(i)}$ *with the highest value of Fuzzy Weighted score)*

7) For each $P_{(i)}$ do

$Ldw_{(i)} = \{ \forall\ t_i : t_i \in h_i, h_i \in P_{(i)} \}$

8) For each document $D_{p(i)}$ in partition $P_{(i)}$

$Vd_{w_{(i)}} = (Fw_{D(i)} - Ldw_{(i)})$

$Pw_{p_{(i)}} = \{x : P(x)\}$

If $P(x) = \left[Vd_{w_{(i)}}\left[D^{(x)}_{P(i)}\right]\right] \geq P\_sup$ then

$$S\left(Vd_{w_{(i)}}\left[D^{(x)}_{P(i)}\right], Pw_{p_{(i)}}\right) = \left|Vd_{w_{(i)}}\left[D^{(x)}_{P(i)}\right] \cap Pw_{p_{(i)}}\right|$$
$$S_m\left(Vd_{w_{(i)}}\left[D^{(x)}_{P(i)}\right], Pw_{p_{(i)}}\right) = \frac{S\left(Vd_{w_{(i)}}\left[D^{(x)}_{P(i)}\right], Pw_{p_{(i)}}\right)}{\left|Pw_{p_{(i)}}\right|}$$

9) $C_p = \{ \forall\ C_i : S_m\left(Vd_{w_{(i)}}\left[D^{(x)}_{P(i)}\right], Pw_{p_{(i)}}\right) \geq 0.5 \}$

Fig 9: Association rule-based clustering algorithm for text

4. EXPERIMENTAL EVALUATION

Several experiments have been carried out in this paper to prove that HDCAR is able to clustering web documents based on more meaningful labels which later extract from highly quality Association rules. Notice the focus of the work is on using both WordNet synonym and hypernyms to get more meaningful label generation and used Fuzzy weighted score measure to remove the overlapping. Moreover, we used MTHAR algorithm that semantically-enriched associations rules. In this section, we experimentally evaluated the performance of HDCAR system by comparing with that of FIHC, k-means, Bisecting k-means, and UPGMA algorithms [45]. For a fair comparison, we did not implement the algorithms by ourselves. Therefore the CLUTO clustering tool is applied to generate the results of k-means, Bisecting k-means and UPGMA. We make use of the FIHC 1.0 tool to generate the results of FIHC. The produced results are then fetched into the same evaluation program to ensure a fair comparison. All the experiments were performed on 2.50 GHz Intel Core i5processor, Windows 7 machine with 6 GB memory. The implementation was written with C#.net to allow fast and flexible development.

4.1 Dataset

In order to show the usefulness of HDCAR system, we used four different kinds of datasets: Classic, Re0, Reuters, and WebKB, which are widely adopted as standard benchmarks for the text categorization task. Table 1 summarizes the statistics of these datasets after the document pre-processing.

Table 1. Statistics for our test datasets

Datasets	Documents	Classes	Class size		D.length	
	Total	Total	Max	Average	Min	Average
Classic	7094	4	3203	1774	1033	39
Re0	1504	13	608	116	11	69
Reuters	8649	65	3725	131	1	42
WebKB	4199	4	1641	1050	504	124

They are heterogeneous in terms of document size, number of classes, cluster size, and document distribution. The smaller document set contains 1504 documents, and the largest one contains 8649 documents.

4.2 Evaluation of cluster quality: Overall F-measure

The F-measure is often employed to evaluate the accuracy of clustering results. F-measure is an aggregation of Precision and Recall concept of information retrieval (Fung, Wang, & Ester, 2003).

$$Recall\left(K_i, C_j\right) = \frac{n_{ij}}{|K_i|} \tag{7}$$

$$Precision\left(K_i, C_j\right) = \frac{n_{ij}}{|C_j|}$$

While F-measure for cluster C_j and class K_i is calculated as in:

$$F\left(K_i, C_j\right) = \frac{2 * Recall\left(K_i, C_j\right) * Precision\left(K_i, C_j\right)}{Recall\left(K_i, C_j\right) + Precision\left(K_i, C_j\right)}$$

where n_{ij} is the number of members of class K_i in cluster C_j. $|C_j|$ is the number of members of cluster C_j and $|K_i|$ is the number of members of class K_i . In our test to evaluate the generated clustering results, standard evaluation measures namely Overall F-measure [4] is widely used. More important, this measure balances the cluster precision and cluster recall denoted $F(C)$ is calculated as in:

$$F(C) = \sum_{K_i \in K} \frac{|K_i|}{|D|} max_{C_j \in C}\{F(K_i, C_j)\}$$

Where K denotes all natural classes; C denotes all clusters at all levels; $|K_i|$ denotes the number of documents in natural class K_i; $|D|$ denotes the total number of documents in the dataset. The range of $F(C)$ is [0,1]. In general, the higher the $F(C)$ values indicate the higher the quality of clustering.

4.3 Experimental results and analysis

The experiments were conducted by the following steps: First, we evaluated HDCAR on the four datasets mentioned earlier and compute its accuracy with that of FIHC, Bisecting k-means and UPGMA. Moreover, we verified if the use of WordNet can improve the clustering accuracy and aid in generating more useful labels for the derived clusters. Second, we evaluate the efficiency and scalability of HDCAR and compared with that of FIHC and Bisecting k-means.

4.3.1 Accuracy comparison

Table 2 presents the obtained overall F-measure values for HDCAR and the other three algorithms by comparing four different numbers of clusters, namely 3, 15, 30, and 60 on the four datasets respectively. Moreover, we run HDCAR with WordNet for all different datasets with the top 5 level of hypernyms and selected the best result. In the mining step, the minimum support has a critical role since it must be properly chosen such that it is not too high where we may lose some important terms or too low where uninteresting terms are generated.

We chose the minimum support ranging from 2% to 6% for all datasets. To allow the more useful association rules to generate, the suitable minimum confidence threshold chosen to be ranging from 85% to 100%. It is apparent that the average accuracy of HDCAR is considerably better than Bisecting k-means and FIHC in several cases. Moreover, UPGMA is not available for large datasets because of some experimental results cannot be generated for UPGMA, and we denoted them as NA. Since FIHC is not available for the documents of long average length, there is no experimental result generated on the WebKB dataset. By observing the average overall F-measure values of all test cases in Figure 10, we can realize that HDCAR produces high-quality clusters for all number of clusters and for all datasets. When the number of clusters changes we observed that the clustering accuracy of k-means, Bisecting k-means, and UPGMA are sensitive since these algorithms require users to specify the number of clusters as an input parameter.

4.3.2 The effect of enrichment document representation

As described in sec 3.2, we utilized WordNet to enrichment the document representation by exploiting both Synonyms and Hypernyms as useful features for clustering. We demonstrate the effect of adding both of them into the different datasets as follows: -

In this step, we tested only FIHC and HDCAR since FIHC is more accuracy than Bisecting k-means algorithm. We tested them twice, first with no enrichment representation and the second by the addition of Synonyms and Hypernyms of different levels (at least 5 levels). The results in Table 3 show that the average overall F-measure values of HDCAR are

superior to that of FIHC when adding Synonyms and Hypernyms for all 5 levels for all datasets and especially for Reuters. This is due to the effect of using Fuzzy Weighting Schema to reduce and filter out noise for clustering and potentially leads to better clustering quality.

Table 2. The obtained overall F-measure comparisonfor four clustering algorithms on the four datasets

Dataset	# of clusters	HDCAR	FIHC	UPGMA	Bi. *k* means
Classic	3	0.67	0.53	NA	0.59
	15	0.63	0.53	NA	0.60
	30	0.60	0.52	NA	0.45
	60	0.59	0.51	NA	0.28
	Average	**0.62**	**0.52**	**NA**	**0.48**
Re0	3	0.55	0.40	0.36	0.37
	15	0.53	0.42	0.45	0.38
	30	0.53	0.39	0.47	0.38
	60	0.51	0.40	0.34	0.30
	Average	**0.53**	**0.40**	**0.41**	**0.36**
Reuters	3	0.65	0.48	NA	0.42
	15	0.55	0.47	NA	0.43
	30	0.56	0.46	NA	0.37
	60	0.52	0.40	NA	0.30
	Average	**0.57**	**0.45**	**NA**	**0.38**
WebKB	3	0.53	NA	0.44	0.34
	15	0.56	NA	0.43	0.20
	30	0.50	NA	0.43	0.15
	60	0.49	NA	0.39	0.10
	Average	**0.52**	**NA**	**0.42**	**0.20**

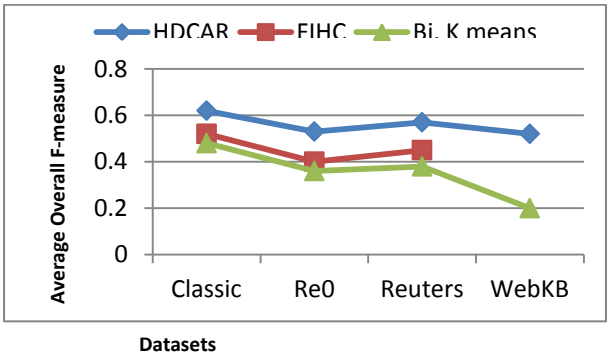


Fig 10: Average Overall F-measure comparisons for three clustering algorithms on the four datasets

Table 3. The effect of enriching the document representation on the datasets

Dataset	Classic		Re0		Reuters		WebKB
	HDCAR	FIHC	HDCAR	FIHC	HDCAR	FIHC	HDCAR
St	0.51	0.47	0.60	0.38	0.43	0.52	0.42
*h*1	0.64	**0.49**	0.58	**0.36**	0.40	**0.42**	**0.47**
*h*2	0.65	0.49	0.58	0.35	0.42	0.41	0.43
*h*3	**0.68**	0.48	**0.61**	0.36	0.48	0.37	0.38
*h*4	0.62	0.45	0.57	0.36	0.43	0.37	0.38
*h*5	0.63	0.45	0.55	0.36	**0.55**	0.36	0.33

4.3.3 Efficiency and scalability

Many experiments were conducted to analyze the efficiency and scalability of HDCAR. All previous methods don't take into account improving the execution time during the experiments although the time is a critical factor in the clustering process, especially with the large text documents. Figure 11 depicts the average execution time of HDCAR on the Reuters datasets. We set two different minimum support threshold values ranging from 2% to 6% for all datasets to evaluate the performance.

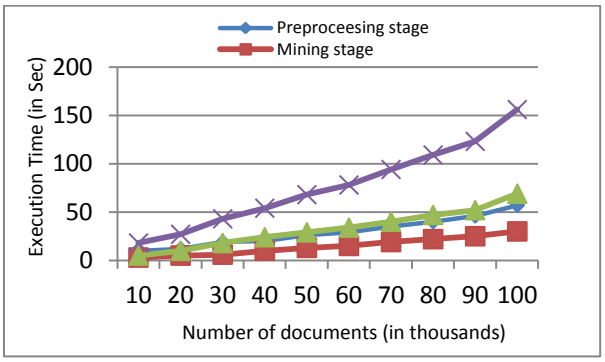


Fig 11: The detailed time cost analysis of HDCAR on Reuter dataset

From Figure 11, we further found that the average execution time of the mining stage on the two datasets is decreased incomparable to the other algorithms. Using MHTAR algorithm in the mining step has a significant impact for speeding up the mining and clustering steps since the time is consumed in building a hash table only one time. With decreasing the minimum support values, the runtime not increased. This is due to saving the hash table into secondary media in the first time; we only begin selecting large frequent terms from the saved hash table. Consequently, at different minimum support threshold, there is no time-consuming in generating new association rules. It also shows that there is no time-consumed in the mining stage especially for the large size of documents. In the clustering stage, most of the time is spent on constructing initial clusters and its runtime is almost linear with respect to the number of documents.

5. CONCLUSION

Although the extensively studied in recent years on document clustering methods, there still exist several challenges for increasing the quality of clustering. In this paper, three document clustering problems are solved by the proposed system at the same time. The problems were considering the semantic relationships among the terms, getting more meaningful cluster labels, and overcoming the overlapping between clusters. The approach combines multi-hash tire association rule-based approach with WordNet to alleviate these problems. The WordNet synonym and hypernyms are used to enhance the quality of the initial representation of all documents in order to exploit the semantic relations between terms. Then, Multi-Hash Tire Association Rules algorithm automatically generates a set of highly-related Association Rules used to cluster documents. Finally, each document is assigned to only one cluster with the highest fuzzy weighted score measure, and then the highly similar non-overlapping clusters are merged. HDCAR system has several ad-vantages that are: assigning meaningful labels to the generated clusters, based on association rules, can help users conveniently recognize each generated set and thus easily analyze the results. Moreover, the removing of the overlapping between clusters guarantees that every document in the cluster still contains the mandatory identifiers. The performance of HDCAR was experimentally evaluated in comparison with k-means, FIHC, Bisecting k-means and UPGMA methods. A large pool of Classic, Re0, Reuters and WebKB dataset are used in the experiments. The experimental results reveal that HDCAR system has better efficiency, accuracy quality, and lower execution time than other methods based on the comparison on the datasets. In addition, it increased the documents clusterization speed, as a result of the reduction of their dimensionality.

6. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their help to improve this paper. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

7. REFERENCES

[1] R. M. Aliguliyev, "Clustering of document collection – A weighting approach", Journal of Expert Systems with Applications, 2009, 36(4): 7904–7916.

[2] R. M. Aliguliyev, "Automatic document summarization by sentence extraction", Journal of Computational Technologies, 2007, 12(5): 5–15.

[3] J. Kuo, H. Chen, "Cross-document event clustering using knowledge mining from co-reference chains", Journal of Information Processing and Management, 2007, 327–343.

[4] Andrews O, Fox A. 2007. Recent developments in document clustering [R]. Computer Science, Virginia Tech, 1-25.

[5] S. Chow, H. Zhang, M. Rahman, "A new document representation using term frequency and vectorized graph connectionists with application to document retrieval", Journal of Expert Systems with Applications, 2009, 36(10):12023–12035.

[6] S. Jun, S. Park, D. Jang, "Technology forecasting using matrix map and patent clustering", Journal of Industrial Management and Data Systems, 2012, 112(5): 786–807.

[7] C. Luo, Y. Li, S. Chung, "Text document clustering based on neighbors", Journal of Data & Knowledge Engineering,2009, 68(11): 1271–1288.

[8] S. Jun, S. Park, D..Jang, "Document clustering method using dimension reduction and support vector clustering to overcome sparseness", Journal of Expert system with Applications, 2014, 41(7): 3204-3212.

[9] J. Zamora, M. Mendoza, H. Allende, "Hashing-based clustering in high dimensional data", Journal of Expert system with Applications, 2016, 62(c):202-211.

[10] F. França, "A hash-based co-clustering algorithm for categorical data", Journal of Expert System with Applications, 2016, 64(c):24-35.

[11] Steinbach M, Karypis G , Kumar V. 2000. A comparison of document clustering techniques. In Proceedings of the international Conference on Knowledge Discovery and Data Mining (KDD), 1-20.

[12] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, "A survey of kernel and spectral methods for clustering", Journal of Pattern Recognition, 2008, 41(1):176–190.

[13] J. Grabmeier, A. Rudolph, "Techniques of cluster algorithms in data mining", Journal of Data Mining and Knowledge Discovery, 2002, 6(4):303–360.

[14] A. K. Jain, M. N. Murty, P. J. Flynn, "Data clustering: A review", Journal of ACM Computing Surveys, 1999, 31(3):264–323.

[15] L. Parsons, E. Haque, H..Liu, "Subspace clustering for high dimensional data: A review", Journal of ACM SIGKDD Explorations Newsletter, 2004, 6(1): 90–105.

[16] R. Xu, D.Wunsch, "Survey of clustering algorithms", Journal of IEEE Transactions on Neural Networks, 2005, 16(3):645–678.

[17] MacQueen J. B. 1967. Some methods for classification and analysis of multivariate observations. In Proceedings of the international Conference on Mathematical Statistics and Probability, 281–297.

[18] Manning C. D, Schutze H. 2000. Foundations of statistical natural language processing. (2nd Ed.). Cambridge: England: MIT Press.

[19] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou, A. Song, "Efficient agglomerative hierarchical clustering", Journal of Expert system with Applications, 2015, 42(5): 2785-2797.

[20] Baeza-Yates R, Ribeiro-Neto R. 1999. Modern information retrieval. (2nd edition.). NY: Addison Wesley, ACM Press.

[21] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. Miller, "Introduction to WordNet: An On-line Lexical Database", Journal of Lexicography, 1990, 3(4):235–244.

[22] T. Wei, Y. Lu, H. Chang, Q. Zhou, X. Bao, "A semantic approach for text clustering using WordNet and lexical chains", Journal of Expert System with Applications, 2015, 4:2264-2275.

[23] Y. Li, C. Luo, S. M. Chung, "Text clustering with feature selection by using statistical data", Journal of IEEE Transactions on Knowledge and Data Engineering, 2008, 20(20):641–652.

[24] Beil F, Ester M, Xu X. 2002. Frequent term-based text clustering. In Proceedings of the international Conference on knowledge Discovery and Data Mining, 436–442.

[25] R. Agrawal, J. Shafer, "Parallel mining of association Rules", Journal of IEEE Transactions on Knowledge and Data Engineering, 1996, 8(6):962-969.

[26] Ch. Chen, F. Tseng, T. Liang, "Mining fuzzy frequent itemsets for hierarchical document clustering", Journal of Information Processing & Management, 2010, 46(2):193-211.

[27] Zaiane O, Antonie M. 2002. Classifying text documents by association terms with text categories. In Proceedings of the International Conference of Australasian Database, 215-222.

[28] Xiangwei L, Pilian H. 2005. A study on text clustering algorithms based on frequent term sets. In Proceedings of the international Conference on Advanced Data Mining and Applications. 347–354.

[29] L. Abualigah, A. Khader, M. Al-Betar, O. Alomari, "Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering", Journal of Expert system with applications, 2017, 84:24-36.

[30] Agrawal R, Imielinski T, Swami A. 1993. Mining association rules between sets of items in large databases. In Proceedings of the international Conference on Management of Data, 207–216.

[31] Fung B, Wang K, Ester M. 2003. Hierarchical document clustering using frequent itemsets. In Proceedings of the international Conference on Data Mining. 59–70.

[32] Yu H, Searsmith D, Li X, Han J. 2004. Scalable construction of topic directory with nonparametric closed termset mining. In Proceedings of the international Conference ICDM, 563–566.

[33] A. Abdelmalek, E. Zakaria, S. Michel, "Evaluation of text clustering methods using WordNet", Journal of Information Technology, 2010, 7(4):349-357.

[34] T. Gharib, M. Fouad, M. Aref, "Fuzzy document clustering approach using WordNet lexical categories", Journal of Advanced Techniques in Computing Sciences and Software Engineering, 2010. 181-186.

[35] Carmel D, Roitman H, Zwerdling N. 2009. Enhancing cluster labeling using Wikipedia. In Proceedings of the International Conference on Research and Development Information Retrieval, 139–146.

[36] C. Chen, F. Tseng, T. Liang, "An integration of WordNet and fuzzy association rule mining for multi-label document clustering", Journal of Data & Knowledge Engineering, 2011, 69:1208-1226.

[37] Y. Tseng, "Generic title labeling for clustered documents", Journal of Expert Systems with Applications, 2009, 37(3):2247–2254.

[38] Treeratpituk P, Callan J. 2006. Automatically labeling hierarchical clusters. In Proceedings of the international Conference on Digital Government, 167-176.

[39] Sedding J, Kazakov D. 2004. WordNet-based text document clustering. In Proceedings of COLING-Workshop on Robust Methods in Analysis of Natural Language Data, 104-113.

[40] Y. Li, M. Chung, D. Holt, "Text document clustering based on frequent word meaning sequences", Journal of Data and Knowledge Engineering, 2008, 64:381–404.

[41] Kiran R, Ravi S, Vikram P. 2010. Frequent itemset based hierarchical document clustering using Wikipedia as external knowledge. In Proceedings of the international conference on Knowledge-based and intelligent information and engineering systems, 11-20.

[42] V. Bhatia, R. Rani, "A parallel fuzzy clustering algorithm for large graphs using Pregel", Journal of Expert System with Applications, 2017, 78(c):135-144.

[43] Mahgoub H, keshk A, Torkey F, Ismail N. 2010. An Efficient Online System of Concepts Based Association Rules Mining. In Proceedings of the international Conference on informatics and systems, 1-8.

[44] N. Negm, P. Elkafrawy, M. Amin, A. Salem, "Clustering Web Documents based on Efficient Multi-Tire Hashing Algorithm for Mining Frequent Termsets", Journal of Advanced Research in Artificial Intelligence, 2013, 2(6): 6-14.

[45] C. Michenerand, R. Sokal, "A quantitative approach to a problem in classification", Journal of Evolution, 1957, 11(2):130–162.