A Frequent Concepts Based Document Clustering Algorithm

Rekha Baghel
Department of Computer Science & Engineering
Dr. B. R. Ambedkar National Institute of Technology,
Jalandhar, Punjab, 144011, India.

Dr. Renu Dhir
Department of Computer Science & Engineering
Dr. B. R. Ambedkar National Institute of Technology,
Jalandhar, Punjab, 144011, India.

ABSTRACT
This paper presents a novel technique of document clustering based on frequent concepts. The proposed technique, FCDC (Frequent Concepts based document clustering), a clustering algorithm works with frequent concepts rather than frequent items used in traditional text mining techniques. Many well known clustering algorithms deal with documents as bag of words and ignore the important relationships between words like synonyms. the proposed FCDC algorithm utilizes the semantic relationship between words to create concepts. It exploits the WordNet ontology in turn to create low dimensional feature vector which allows us to develop a efficient clustering algorithm. It uses a hierarchical approach to cluster text documents having common concepts. FCDC found more accurate, scalable and effective when compared with existing clustering algorithms like Bisecting K-means , UPGMA and FIHC.

Keywords
Document clustering, Clustering algorithm, Frequent Concepts based Clustering, WordNet.

1. INTRODUCTION
The steady and amazing progress of computer hardware technology in the last few years has led to large supplies of powerful and affordable computers, data collection equipments, and storage media. This technology provides a great boost to the database and information industry and makes a huge number of databases and information repositories available for transaction management, information retrieval, and data analysis. So we can say that this technology provides a tremendous growth in the volume of the text documents available on the internet, digital libraries, news sources and company-wide intranets. With the increase in the number of electronic documents, it is hard to manually organize, analyze and present these documents efficiently. Data mining is the process of extracting the implicit, previously unknown and potentially useful information from data. Document clustering is one of the important techniques of data mining which of unsupervised classification of documents into different groups (clusters), so those documents in each cluster share some common properties according to some defined similarity measure. So Documents in same cluster have high similarity but they are dissimilar to documents in other cluster [1].

Let's observe closely the special requirements for good clustering algorithm:

1. The document model should better preserve the relationship between words like synonyms in the documents since there are different words of same meaning.
2. Associating a meaningful label to each final cluster is essential.
3. The high dimensionality of text documents should be reduced.

The goal of this paper is to present a proposed document clustering algorithm, named FCDC (Frequent Concepts based clustering), is designed to meet the above requirements for good text clustering algorithm.

The special feature of proposed FCDC algorithm is: it treats the documents as set of related words instead of bag of words. Different words shares the same meanings are known as synonyms. Set of these different words that have same meaning is known as concept. So whether document share the same frequent concept or not is used as the measurement of their closeness. So our proposed algorithm is able to group documents in the same cluster even if they do not contain common words.

In FCDC, we construct the feature vector based on concepts and apply an Apriori paradigm [2] for discovering frequent concepts then frequent concepts are used to create clusters. We found our FCDC algorithm is more efficient and accurate than other clustering algorithms.

The rest of the paper is organized as follows: Section 2 describes the literature review of this work. Section 3 describes our algorithm in more detail. Section 4 discussed some experimental results. We conclude the paper in section 5.

2. RELATED WORK
Many clustering techniques have been proposed in the literature. Clustering algorithms are mainly categorized into hierarchical and partitioning methods [2, 3, 4 5]. A hierarchical clustering method works by grouping data objects into a tree of clusters [6]. These methods can further be classified into agglomerative and divisive hierarchical clustering depending on whether the hierarchical decomposition is formed in a bottom-up or top-down fashion. K-means and its variants [7, 8, 9] are the most well-known partitioning methods [10].

Lexical chains have been proposed in [11] that are constructed from the occurrence of terms in a document.

Problem to improve the clustering quality is addressed in [10] where the cluster size varies by a large scale. They have stated that variation of cluster size reduces the clustering accuracy for

India has warned Pakistan that any accumulation of warms will go against its own interest. Indian government admonished Pakistan for careless in handling the sadbhavna project. The measures now recommend by the planning commission to improve the Indian economy are very practical. The government has announced its export policy for the next three years. The new policy was broadcast on the television last evening in full detail. It proclaims some incentives for the manufacturer exporters. The policy was published in gazette by government of India.

some of the state-of-the-art algorithms. An algorithm called frequent Itemset based Hierarchical clustering (FIHC) has been proposed, where frequent items i.e. minimum fraction of documents have used to reduce the high dimensionality and meaningful cluster description. However, it ignores the important relationship between words.

The benefits of partial disambiguation of words by their PoS is explored in [12]. They show how taking into account synonyms and hypernyms, disambiguated only by PoS tags, is not successful in improving clustering effectiveness because of the noise produced by all the incorrect senses extracted from WordNet. A possible solution is proposed which uses a word-by-word disambiguation in order to choose the correct sense of a word. In [13] CFWS has been proposed. It has been found that most of existing text clustering algorithms use the vector space model which treats documents as bags of words. Thus, word sequences in the documents are ignored while the meaning of natural language strongly depends on them.

In [14] the authors have proposed various document representation methods to exploit noun phrases and semantic relationships for clustering. Using WordNet, hypernymy, hyponymy, holonymy, and meronymy have been utilized for clustering. Through a series of experiments, they found that hypernymy is most effective for clustering.

In this paper we propose a document clustering algorithm based on concepts. The proposed method generates better results from FIHC and other clustering methods.

2.1 WordNet

WordNet [6] is a large lexical database, a combination of a dictionary and thesaurus for the English language. WordNet organizes words into groups known as synsets. Each synset contains a group of synonymous words and collocations and corresponds to a concept. In addition, each synset also contains pointers to other semantically related synsets. WordNet database contains 155,287 words organized in 117,659 synsets for a total of 206,941 word-sense pairs. WordNet has four categories of words - nouns, verbs, adjectives and adverbs. Within each category, the words are organised into synsets. Each synset is identified by a unique synset number. Each word belongs to one or more synsets with each instance corresponding to different senses of the word and are numbered according to their frequency of occurrence in real world usage.

In this paper, we propose a document clustering method for generating the hierarchy of clusters from this representation. The proposed method produces better result from FIHC and other document clustering approaches. We validate our approach on a number of real-life document datasets.

3. FREQUENT CONCEPTS BASED DOCUMENT CLUSTERING

The idea of our proposed Frequent concepts based document clustering (FCDC) algorithm is to cluster the documents by using the concepts (i.e. the words that have the same meaning) that present in sufficient number of documents. Our approach does not consider the documents as bag of word but as a set of semantically related words. Proposed algorithm (Figure 1) first creates a feature vector based on the concepts identified using WordNet ontology. After creating the feature vector based on concepts, we utilize Apriori paradigm, designed originally for finding frequent itemsets in market basket datasets, to find the frequent concepts from the feature vector. Then we formed the initial clusters by assigning one frequent concept to each cluster. For example, FCDC created a cluster for the frequent concept (announce, broadcast, proclaims, publish) made up of all the documents that contain words which are either to identical or related to this concept. The algorithm process the initial clusters makes final clusters arranged in hierarchical structure.

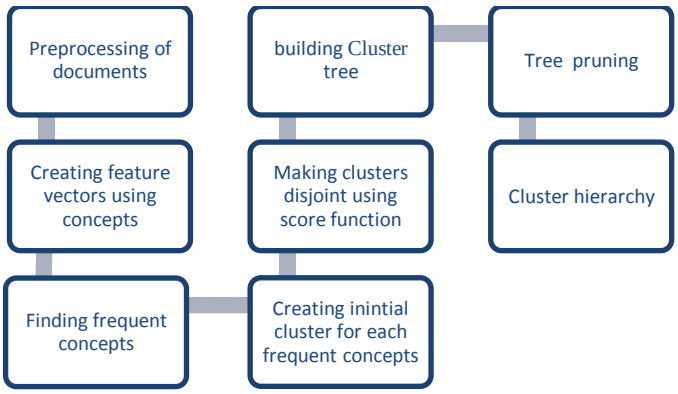

Figure 1. Overview of algorithm architecture

Figure 2. Contextual example

We first extract the noun and verb phrases from contextual example (Figure 2) and then find the following concepts:

Concept 1: admonish, warn, recommend;

Concept 2: announce, broadcast, proclaims, publish;

Concept 3: india, indian government, government, government of india;

Concept 4: export policy, policy, policy

Concept5: pakistan, pakistan

Here concept is a set of synonymous words named synset.

Document Pre-processing

Preprocessing is a very important step since it can affect the result of a clustering algorithm. So it is necessary to preprocess the data sensibly. Preprocessing have the several steps that take a text

document as input and output as a set of tokens to be used in feature vector.

Coreference resolution: it is a process that identifies the co referenced words. If two or more noun phrases referring to the same entity are known as corefer. Since different words expressing the same entity may also present, we tried to catch them by using the "Cherrypicker: A coreference resolution tool "[28]. It is used to identify the co referenced words. It co referenced the noun that belong to one of the seven semantic classes (namely PERSON, ORGANIZATION, GEO POLITICAL ENTITY, FACILITY, VEHICLE and WEAPON)

For instance,

Input: John Ray will join the board as a non-executive director. Mr. Ray is the chairman of Ericson. He lives in Newyork. The chairman has announced the company's one year plan. .

Output: John Ray will join the board as a non-executive director. John Ray is the chairman of Ericson. John Ray lives in Newyork. John Ray has announced the company's one year plan.

Here different words *Mr. Ray, He, The chairman are* expressing the same entity John Ray so CHERRYPICKER would create the following corefrenced list: *John Ray, Mr. Ray, He, The chairman.* In such cases we would keep only one of these entities which count each reference to all the others, that reducing the size of the feature vector.

Tokenization: this step splits sentences into individual tokens.

For instance,

Input: John Ray will join the board as a non-executive director. Mr. Ray is the chairman of Ericson.

Output: John Ray will join the board as a non-executive director.

Mr. Ray is the chairman of Ericson.

Parts of speech tagging: After breaking each document into sentences, we use Stanford parser to extract only noun and verb phrases and remove non-word tokens such as numbers, HTML tags and punctuation. We consider only noun and verbs since 80% of the total terms are noun and verbs [18].

Stopword removal: stopwords are words considered not to convey any meaning. We use a standard list of 571 stopwords and remove them from the documents [19, 20].

We use the morphological capabilities of WordNet 2.1 to find the base form of a word and apply the porter stemmer algorithm only on those terms that do not appear as morphs in WordNet 2.1.

Text document representation: In the vector space model, a document is represented as a feature vector $d = (tf_{t1},\ldots\ldots tf_{ti)}$, where tf_t returns the absolute frequency of term $t \in T$ in document $d \in D$, where D is the set of documents and $T = \{t_1,t_2,\ldots\ldots t_i\}$ is the set of all different terms occurring in D. In the WordNet-based clustering method [14], first, the concepts in documents are identified as a set of terms that have identity or synonym relationship, i.e. synsets in the WordNet ontology. Then the concept frequencies are calculated as given below:

$$Cfc = \sum_{t_m \in r(c)} tf_{tm} \quad (1)$$

Where r(c) is the set of different terms of document d_i those belong to concept C.

If three terms t_1,t_2,t_3 having terms frequencies tf_{t1}, tf_{t2} , tf_{t3} respectively and they have the same meaning and belongs to concept C_1, then $Cf_{c1} = tf_{t1} + tf_{t2} + tf_{t3}$.

It is worth note that WordNet returns an ordered list of synsets based on a term. The ordering is in a way that more commonly used terms are listed before less commonly used terms. It has been showed that using the first synset as identified concept for a term can improve the clustering performance more than that of using all the synsets to calculate concept frequencies [14]. In this paper, we also use only the first synset as the concept for a term for computing concept frequencies. Here the 'term' represents all different noun and verb phrases.

The weight of each concept C in document d is computed as:

$$Wc = Cfc \times idf_c \quad (2)$$

Where idf_c is the inverted document frequency of concept C by counting how many documents in which concept C appears. Finally a document d is represented as a vector• of concepts weights, i.e.

$$d = (Wc_1,\ldots\ldots\ldots\ldots, Wc_i) \quad (3)$$

To cluster similar documents together, the majority of the document clustering algorithms requires a similarity measure between two documents d1 and d2. There are of different types of similarity measures are proposed in literature, but the most common one is the cosine measure [5] and it is defined below

$$\text{Similarily } (d1, d2) = \text{cosine}(d1, d2) = \frac{(d1 \bullet d2)}{\|d1\|.\|d2\|} \quad (4)$$

Where ● represents the vector dot product and || represents the length of a vector

Document clustering

The Proposed document clustering algorithm consists of the following phases: finding frequent concepts, creating initial clusters for each frequent concept, making clusters disjoint using score function, building cluster tree, and tree pruning. The algorithm is explained in detail as following:

Step 1: Create feature vectors using concepts, each document is represented by a vector of frequencies of concepts creating using WordNet ontology.

Step2: Generate frequent concepts using Apriori paradigm [15, 21] based on threshold global support given by the user.

Step3: After finding the frequent concepts now we construct the initial clusters for global frequent concepts. All documents including frequent concept are putting in the same cluster. At this stage one document can have few frequent concepts. If a cluster1 belongs to concept1 then concept 1 is the cluster label for cluster1.

Step4: Since proposed algorithm is hard clustering type so next step is to make disjoint clusters i.e. one document can has only one cluster. Here we use a score function to make cluster Disjoint. If a document belongs to several clusters then we calculate the score against each respective cluster and assign the document to cluster that have best score among them. If there are few best clusters then we assign the document to the cluster that has longest cluster label.

$$\text{Score}\left(C_i \leftarrow doc_j\right) = [\sum_x n(x) * \text{cluster\_support}(x)] - [\sum_{x'} n(x') * \text{global\_support}(x')] \quad (5)$$

Where:

x represents the global frequent concept in doc_i and the concept is also cluster frequent in C_i.

x' represents a global frequent concept in doc_i but the concept is not cluster frequent in C_i.

n(x) is the frequency of concept in the feature vector of doc_i.

n(x') is the frequency of concept in the feature vector of doc_i.

step 5: After making clusters disjoint, next step is to build the tree ; start making a tree from bottom-up fashion by choosing a parent for each cluster. In this step we are going from specification to generalization i.e. the cluster C_i having longest cluster label K are choosing first then choosing the parent for cluster C_i at label K-1. Parent should have the label being a subset of child cluster label. Now all the documents in the subtree of C_i are combined into a single abstract document doc(C_i) and the calculate the score of doc(C_i) against each latent parent. The latent parent with the highest score would become the parent of C_i. at last remove any leaf cluster that does not contain any document.

Step 6: Prune the tree, the chief intend of tree pruning is to join similar clusters in order to produce a expected topic hierarchy for browsing and to increase the clustering accuracy. This process has two sub steps: Child Pruning and Sibling Merging. Inter-cluster similarity is an important term that typically used in both steps.. Inter-cluster similarity is the basis for merging clusters.

Inter-cluster similarity between two clusters C_a and C_b is calculated by measuring the similarity of C_a to C_b. it is done by treating one cluster as a conceptual document (by combining all documents in the cluster) and by calculating its score against the other cluster by using the following score equation:

$$\text{Sim}(C_a \leftarrow C_b) = \frac{\text{Score}(C_a \leftarrow \text{doc}(C_b))}{\sum_x n(x) + \sum_{x'} n(x')} + 1 \quad (6)$$

Where:

x represents a global frequent concepts in doc(C_b) and the concept is also cluster frequent in C_a.

x' represents a global frequent concept in doc(C_b) but the concept is not cluster frequent in C_a.

n(x) is the frequency of concept x in the feature vector of doc(C_b)

n(x') is the frequency of x' in the feature vector of doc(C_b) .

Then inter similarity defined as:

$$\text{Inter\_Sim}(C_a \leftrightarrow C_b) = [\text{Sim}(C_a \leftarrow C_b) * \text{Sim}(C_b \leftarrow C_a)]^{1/2} \quad (7)$$

where C_a and C_b are two clusters including their descendants; Sim(Ca ← C_b) is the similarity of C_b against Ca; Sim(C_b ← C_a) is the similarity of C_a against C_b.

The two sub steps of tree pruning are described as follow:

Child Pruning: this sub step initiates by scanning the whole tree in bottom-up fashion. During this scan, for any non-leaf node calculates inter-similarity between this node and its children; and each child with inter-similarity greater than 1 is pruned.

Sibling Merging: it merges similar clusters at level 1, starting by calculating the inter-similarity for each pair of clusters at level 1 and merging the cluster pair that has the highest inter-similarity, the children of the two clusters become the children of the merged cluster. The Sibling merging stops when all inter-similarity between each pair becomes less than or equal to 1.

4. EXPERIMENTAL EVALUATIONS

To evaluate the effectiveness of proposed algorithm, this section presents the result comparisons between proposed algorithm and several popular hierarchical document clustering algorithms Bisceting K-means [7, 8, 22], FIHC and UPGMA [7, 8]. We obtained the freely available source code of FIHC from author's site and compiled the program into windows environment. For Bisecting K-means algorithm we use CLUTO-2.0 [22] Clustering Toolkit to generate the results of Bisecting K-means.

4.1 Datasets

The summary of data sets used in this paper is given in table1. The details of each data set are explained here. Classic data set [23] is combined from the four classes CACM, CISI, CRAN, and MED of computer science, information science, aerodynamics and medical articles. Data set re0 are taken from Reuters – 21578 Text Categorization Test Collection Distribution 1.0 [24]. Data set wap is taken from the WebACE project [25]. All the three data sets are real data sets.

Table 1. Summary description of data sets

Data set	No. of Docs.	No. of Classes	Class Size	Avg Class Size	No. of Terms
Classic	7094	4	1033-3203	1774	12009
Wap	1560	20	5-341	78	8460
Re0	1504	13	11-608	116	2886

4.2 Performance Evaluation Measures

We used the F-measure to evaluate the accuracy of the clustering algorithms. The F-measure is a combination of *precision* and *recall* values used in information retrieval. Each cluster obtained can be considered as a result of query, whereas each pre-classified set of documents can be considered as a desired set of documents for that query. We treat each cluster as if it was the result of a query and each class as if it was the relevant set of documents for a query. The recall, precision, and F-Measure for natural class K_i and cluster C_j are calculated as follows:

$$\text{Recall}\left(K_i, C_j\right) = \frac{n_{ij}}{|K_i|} \quad (8)$$

$$\text{Precision}\left(K_i, C_j\right) = \frac{n_{ij}}{|C_j|} \quad (9)$$

Where, n_{ij} is the number of members of class K_i in cluster C_j. The corresponding F-Measure F (K_i, C_j) is defined as:

$$F\left(K_i, C_j\right) = \frac{2 * \text{Recall}\left(K_i, C_j\right) * \text{Precision}\left(K_i, C_j\right)}{\text{Recall}\left(K_i, C_j\right) + \text{Precision}\left(K_i, C_j\right)} \quad (10)$$

$F(K_i, C_j)$ represents the quality of cluster C_j in describing class K_i. While computing $F(K_i, C_j)$ in a hierarchical structure, all the documents in the subtree of C_j are considered as the documents in C_j . The overall F-measure, F(C), is the weighted sum of the maximum F-measure of all the classes as defined below:

$$F(C) = \sum_{K_i \in K} \frac{|K_i|}{|D|} \max_{C_j \in C} \left\{ F\left(K_i, C_j\right) \right\} \quad (11)$$

Where, *K* denotes the set of natural classes; *C* denotes all clusters at all levels; $|K_i|$ denotes the number of documents in class K_i; and $|D|$ denotes the total number of documents in the data set.

Taking the maximum of $F(K_i, C_j)$ can be viewed as selecting the cluster that can best describe a given class, and F(C) is the weighted sum of the F-Measure of these best clusters. The range of F(C) is [0, 1]. A larger F(C) value indicates a higher accuracy of clustering. F-Measure needs pre-classified datasets to be used so that recall and precision can be calculated. The datasets used to evaluate clustering algorithms should divided in classes where class size can vary from few hundred to thousands documents.

4.3 Experimental Results

All experiments were performed on Windows XP PC with a 2.8 GHz Processor and 1GB RAM. We have implemented algorithm in c# under Visual Studio 2005 using the .NET package (Crowe) for accessing WordNet 2.1[26]. The CherryPicker and Stanford tagger has been written in ANSI C.

Table 2 shows the F-measure values for all four algorithms with different numbers of clusters. Proposed algorithm outperforms all other algorithms in terms of accuracy. Figure 2, 3 and 4 shows F-Measure results of Classic, Re0 and Wap dataset with different number of clusters.

Table 2. F-measure comparison of clustering algorithms

Data set	Number of Clusters	Overall F-measure			
		FCDC	FIHC	UPGMA	Bisecting K-means
Classic	3	0.57	**0.62**	N/A	0.59
	15	**0.63**	0.52	N/A	0.46
	30	**0.67**	0.52	N/A	0.43
	60	**0.64**	0.51	N/A	0.27
	Average	**0.62**	0.54	N/A	0.43
Wap	3	0.52	0.40	0.33	**0.47**
	15	**0.59**	0.56	0.49	0.57
	30	0.56	**0.57**	0.49	0.44
	60	**0.64**	0.55	0.53	0.37
	Average	**0.57**	0.52	0.46	0.46
	3	0.55	0.53	**0.58**	0.34
	15	**0.59**	0.45	0.42	0.38
Re0	30	**0.52**	0.43	0.36	0.38
	60	0.53	0.38	0.35	**0.57**
	Average	**0.54**	0.44	0.42	0.41

Performance Investigations on Accuracy

Table 2 shows the F-measure values for all the four algorithms with different user given number of clusters. The F-measure represents the clustering accuracy. The same minimum support, from 5% to 15% is used, for FIHC, UPGMA and FCDC method. The highlighted results show the best algorithm for the particular number of cluster for the specified document dataset and the final average results indicates the better algorithm for specified data set. The proposed clustering algorithm FCDC has work better on all the dataset than all other algorithms and it indicates that FCDC has better accuracy than all algorithms including UPGMA which is regarded as the best in hierarchical document clustering algorithm. Figure 3, shows the F-measure results for the Classic dataset. Classic dataset is the largest dataset among all three datasets. It illustrates that the FCDC has the higher F-measure values then all competitive algorithms. Higher F-measure shows the higher accuracy. Figure 4 shows the F-measure values with number of clusters for Wap dataset, and it indicates that FCDC has high F-measure values in mostly cases. Figure 5 shows F-measure results of Re0 dataset respectively with different number of clusters. FCDC has the higher F-measure values then other algorithms, therefore FCDC provide more accuracy then others. FCDC has better F-measure because it uses a better model for text documents. All bisecting k-means, UPGMA and FIHC use the high dimensional vector space model for text documents. They cannot detain semantic relationship between words, which is important in representing the context in the text documents. FCDC method used the feature vector based on concepts for representing the text documents. FIHC uses the frequent word sets to cluster documents, whereas FCDC uses the frequent concepts to cluster documents

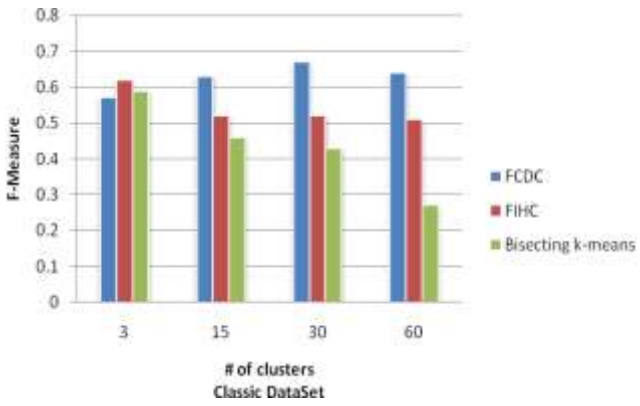

Figure 3. F-Measure Results Comparison with Classic Dataset

. As a result, FIHC has a higher probability of grouping unrelated documents into the same cluster. FCDC has better F-measure than all other algorithms in most cases because it can identify the same concept represented by the different terms i.e. pork and meat denotes the same concept in proposed method.

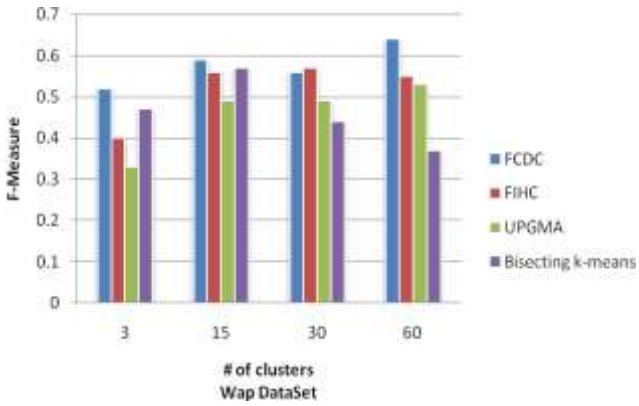

Figure 4. F-measure Results Comparison with Wap Dataset

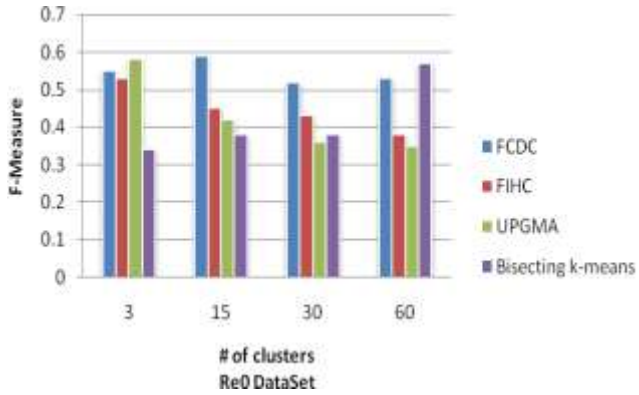

Figure 5. F-measure Results Comparison with Re0 Dataset

Sensitivity to number of clusters

Various algorithms depend on the number of clusters specified by user and numbers of clusters in turn indicate the total clusters at level 1 of tree. Comparing FCDC with bisecting k-means which is sensitive to number of clusters, the figure 1 shows that the accuracy of bisecting k-means starts decreasing as the number of clusters are increased. Thus FCDC is more insensitive to number of clusters and produces better results than FIHC and bisecting k-means.

5. CONCLUSION AND FUTURE SCOPE

The area of document clustering has many issues which need to be solved. In this work, few issues e.g. high dimensionality and accuracy are focused but there are still many issues that can be taken into consideration for further research which are as follows:

1. The proposed algorithm can be modified to soft clustering.
2. Efficiency of the proposed work can be improved by adding more issues.
3. Each concept represents a topic enclosed in the document. This fact could be used to generate titles for a document or a group of document by post processing the set of concepts assigned to a document.

6. REFERENCES

[1] J. Han and M. Kimber. 2000. Data Mining: Concepts and Techniques. Morgan Kaufmann.

[2] Jain, A.K, Murty, M.N., and Flynn P.J. 1999. Data clustering: a review. ACM Computing Surveys, pp. 31, 3, 264-323.

[3] M. Steinbach, G. Karypis, and V. Kumar. 2000. A comparison of document clustering techniques. KDD Workshop on Text Mining'00.

[4] P. Berkhin. 2004. Survey of clustering data mining techniques [Online]. Available: http://www.accrue.com/products/rp_cluster_review.pdf.

[5] Xu Rui. 2005. Survey of Clustering Algorithms. IEEE Transactions on Neural Networks, 16(3):pp. 634-678.

[6] Miller G. 1995. Wordnet: A lexical database for English. CACM, 38(11), pp. 39–41.

[7] L. Zhuang, and H. Dai. 2004. A Maximal Frequent Itemset Approach for Document Clustering. Computer and Information Technology, CIT. The Fourth International Conference, pp. 970 – 977.

[8] R. C. Dubes and A. K. Jain. 1998. Algorithms for Clustering Data. Prentice Hall college Div, Englewood Cliffs, NJ, March.

[9] D. Koller and M. Sahami. 1997. Hierarchically classifying documents using very few words. In Proceedings of (ICML) 97, 14th International Conference on Machine Learning, pp. 170–178, Nashville, US.

[10] B.C.M.Fung, K.Wan, M.Ester. 2003. Hierarchical Document Clustering Using Frequent Itemsets", SDM'03.

[11] Green, S. J. 1999. Building hypertext links by computing semantic similarity. T KDE, 11(5), pp. 50–57.

[12] Sedding, J., & Kazakov, D. 2004. Wordnet-based text document clustering. 3rd Workshop on Robust Methods in Analysis of Natural Language Data, pp. 104–113.

[13] Y. LI, and S.M. Chung. 2005. Text Document Clustering Based on Frequent Word Sequences. In Proceedings of the. CIKM, 2005. Bremen, Germany, October 31- November 5.

[14] Zheng, Kang, Kim. 2009. Exploiting noun phrases and semantic relationships for text document clustering. Information Science 179 pp. 2249-2262.

[15] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors,Proc 20th Int. Conf. Very Large Data Bases, VLDB, pp. 487–499.

[16] Agrawal, T. Imielinski, and A. N. Swami. 1993. Mining association rules between sets of items in large databases. In Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD93), pp. 207–216, Washington, D.C.

[17] Stanford Tagger available at http://nlp.stanford.edu/software/tagger.shtml

[18] K.M. Hammouda, M.S. Kamel. 2004. Document similarity using a phrase indexing graph model. Knowl. Inform. Syst. 6 (6) 710–727.

[19] StopWord List, http://www.lextek.com/manuals/onix/stopwords2.html

[20] Cognitive Science Laboratory at Princeton University Available at: http://www.cogsci. princeton .edu/.

[21] L. Kaufman and P. J. Rousseeuw. 1990. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Sons.

[22] G.Karypis. 2002. Cluto 2.0 clustering toolkit. http://wwwusers.cs.umn.edu/˜ karypis/cluto

[23] Classic. ftp://ftp.cs.cornell.edu/pub/smart/.

[24] E. H. Han, B. Boley, M. Gini, R. Gross, K. Hastings,G. Karypis, V. Kumar,B. Mobasher, and J. Moore. 1998. Webace: a web agent for document categorization and Exploration. In Proceedings of the second international conference on Autonomous agents, pp. 408–415. ACM Press.

[25] D.D.Lewis.Reuters. http://www.research.att.com/˜ lewis/.

[26] Crowe,M. 2000. Wordnet.net library. http://www.opensvn.csie.org/WordNetDotNet/

[27] M. F. Porter. 1980. An algorithm for suffix stripping. Program; automated library and information systems, 14(3), pp.130-137.

[28] CherryPicker Coreference resolution Tool. Available at http://www.hlt.utdallas.edu/~altaf/cherrypicker.html