A Semi-supervised approach to Document Clustering with Sequence Constraints

Murtaza Munawar Fazal[1], Muhammad Rafi[2]

[1]*Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST) Karachi, Pakistan*
[2]*FAST – National University, Karachi, Pakistan*

[1]murtaza.fazal@gmail.com
[2]rafi.muhammad@gmail.com

Abstract - Document clustering is usually performed as an unsupervised task. It attempts to separate different groups of documents (clusters) from a document collection based on implicitly identifying the common patterns present in these documents. A semi-supervised approach to this problem recently reported promising results. In semi-supervised approach, an explicit background knowledge (for example: Must-link or Cannot-link information for a pair of documents) is used in the form of constraints to drive the clustering process in the right direction. In this paper, a semi-supervised approach to document clustering is proposed. There are three main contributions through this paper (i) a document is transformed primarily into a graph representation based on Graph-of-Word approach. From this graph, a word sequences of size=3 is extracted. This sequence is used as a feature for the semi-supervised clustering. (ii) A similarity function based on common-word sequences is proposed, and (iii) the constrained based algorithm is designed to perform the actual cluster process through active learning. The proposed algorithm is implemented and extensively tested on three standard text mining datasets. The method clearly outperforms the recently proposed algorithms for document clustering in term of standard evaluation measures for document clustering task.

Keywords **– Document Clustering, Information Retrieval, Semi-supervised techniques, Data mining and Document graph**

I. INTRODUCTION

The amount of information is growing by two folds which is used by individuals, and different administrative and non-administrative organization. This information has created the need of tools that can be used to manage data without any user intervention. Every user is interested in a portion of data that is relevant to the need out of the large amount of data available throughout the world. For example, if a user searches the internet to find articles related to sports, the user would not be expecting search results to show data that are related to chemistry. Hence, the need of division or grouping the data arises through which data can be differentiated on the basis of requirement and relevancy.

For the solution of such problem, data mining technique is used which provides two broad categories; one is classification and other is clustering. Clustering can be further divided into two types of approaches; supervised and unsupervised approach. Unsupervised clustering attempts to differentiate data in different groups (clusters) based on certain similarity among the groups/clusters automatically whereas in the supervised approach, user intervention is required to cluster the data. The issues with unsupervised approach is that it may also attempt to cluster data based on a word being used frequently in multiple documents but that does not mean that they belong to the same category of documents or it may cluster data on gibberish or special character that belong to multiple documents. Likewise, there are problems with supervised data as well, such as that the supervision given to the algorithm should be correct and should cover a viable subset that can represent the data properly and if these conditions are not met, it will have an adverse effect on the entire result.

Another method known as semi-supervised approach which is a fusion of both supervised and unsupervised techniques, understands the problem with user intervention on a smaller dataset or as known direction. The algorithm utilizes these directions to identify different grouping or clustering on a larger dataset. There are generally two approaches to cater this; metric-based methods and constraint based methods. In metric-based approach, an existing algorithm for clustering is first trained on a supervised dataset and then on the actual dataset. In a constrained-based approach, the algorithm is modified to incorporate user knowledge to get proper clusters.

The approach that has been proposed in this paper is based on extracting word sequence from graph-of-word of a document and then comparing these documents using a semi-supervised learning. Each document is represented by word-sequences of that document and initially all documents are treated as independent clusters (candidate clusters). The

algorithm is provided with two documents that highlights the domain level constraints and instance level constraints. These constraints assist the clustering process to merge documents based on specific constraint (must-link) or separate each other (cannot-link). Further to this, the algorithm has an active learning process which learns the process and develops the similarity and dissimilarity features among the documents in stages and the final output in the form of clusters would produce better clustering results and be refined through two users given constraint dictionary and one self-learned.

II. LITERATURE REVIEW

Different document clustering techniques have been proposed by different researchers to efficiently characterize the document and improve the document clustering results. Some of the approaches prefer to use unsupervised approaches and some algorithms are efficient with supervised approach. Recently, semi-supervised approaches have gained a lot of importance as they tend to improve results over the unsupervised approach. This section is divided into two broad categories i.e. document representation and clustering process to summarize the work.

A. Document Representation

The document clustering process can be divided into three main categories (i) document representation, (ii) similarity measure and (iii) actual clustering algorithm. Approaches used by the traditional document clustering algorithms usually extract features like word, phrases and sequence from documents [1, 2, 3, 4]. These methods focus on the statistics and distribution of features and to define document's similarity. The issue with these approaches is that they extract features based on the frequent words being used or their meaning and they retain neither the words that are used in the document (lesser then a specific number) nor the order in which they were used. Such approach may not be able to retain the actual theme of the document and misguide the clustering process. Document representation is one of the challenging aspects of document clustering. One of the most commonly used document representation model is of Bag of Word (i.e. Vector Space Model) [5]. Other approaches like language modeling by Dirichlet prior [6] approaches, probabilistic BM25 [7] and the divergence from randomness framework PL2 [8] have also been used. These methods represent the entire document transformed as vector of words without any information about the relationship between words.

1) *Clustering Based on Frequent Word / Meaning Sequence (CFWS / CFWMS):* In one of the recent work on document clustering, Clustering Based on Frequent Word Sequence (CFWS) and Clustering Based on Frequent Word Meaning Sequence (CFWMS) claim to retain the context in which the feature sets are used [1] which was the lacking of traditional approaches. These algorithms maintain a list of distinct words that are used frequently in the entire document. Suppose a collection of document "D" which consists of 4 documents d1, d2, d3 and d4. Each of these documents will maintain an independent list of frequently used words in the respective document but the collection D will contain a subset of words from all the documents that are unique among the documents. To maintain common word sequences, 2-word pairs are extracted from all of the documents and words that have lesser frequency then the threshold value (5-15% occurrence of the word in the entire document) would be removed from the sequence pair list. This filtering process reduces the amount of words present in each document which improves the performance of the clustering process. These refined and compact form of documents can be represented as D' = d1' + d2' + d3' + d4'. In CFWS, the documents having similar sequences appearing frequently are merged into a cluster. To merge the documents into clusters, k-mismatch concept using the Landau-Vishkin (LV) algorithm [9] is implemented. The other algorithm CFWMS proposed [1] uses word meaning sequences among the documents. A word may be used in different meaning and to maintain the context of that word, it is converted into root word from its deviated word, synonyms, etc. by using WordNet [10]. For instance, words like cell-phones, mobile-phones, smart-phones and etc. counts toward one word. Words that do not exist in WordNet are not converted and retain as they are.

2) *Graph of Word:* In a recent approach of Graph-of Word [11, 12, 13], document is represented in the terms of a graph where each unique term is represented as vertices of the graph and the edges between them highlights the semantic relationship among the terms [14]. Based on the implementation, a word or even a sentence may be used as a term [12, 13, 15]. Different implementation of the graph can be used in the Graph of Word approach. Graphs can be directed-graph (ordered pairs of vertices) or undirected-graph (unordered pairs of vertices) graphs or it can be a weighted or un-weighted graph [16] such that the number of times two adjacent terms appear in the document may be represented as the weight between those vertices in the former approach and in the later one it is not be considered at all. The approach discussed by Li *et al* [11] uses the un-weighted directed graph as the term search can be maintained in the directed graph and using the un-weighted graph led to better results. To create edges between vertices, a moveable window was used to create edge between adjacent terms. Li *et al* [11] discussed that the window size can be between 3 and 13. To understand the working of the Graph of Word, assume that a document d1 = "Paris is a beautiful city". Here, each distinct word would be represented as a vertex and words within the sliding window of size 3 would have an edge between them. For clarity purpose, edges leading to and from the word "Paris" are shown in figure 1.

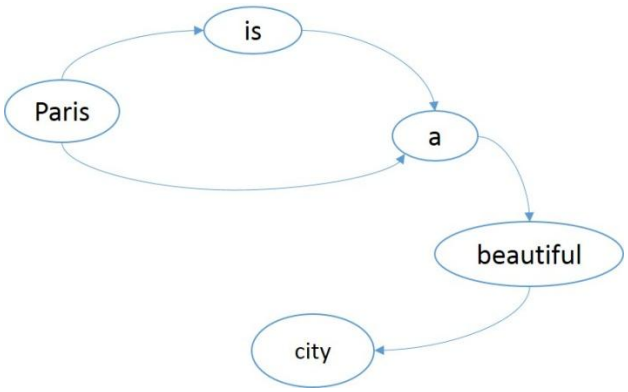

Fig. (1). Example of Graph of Word [17]

To produce clusters, different document graphs are compared together using the term frequency-inverse document frequency (TF-IDF) and documents having higher TF-IDF values are merged together into clusters. The retrieval model cab be defined as a function of a term weight (TW) and a document weight (DW) [11].

The proposed approach in this paper is extracting word-sequences from the graph of word. Initially, a graph of the entire document is created with the sliding window size as three, so the next two words following a word will have an edge from the first word. This produces a graph similar to the one proposed by Li *et al* [11] which retains the context in which the words were used. Treating this graph as the representation of the document, word-sequences [1] can be extracted which is the final representation of the document. This word-sequence differentiates from the one that is proposed by Jain *et al* [1] as in this approach the input is a graph instead of a full document. To simplify the approach, consider the example that was previously discussed where document d1 = "Paris is a beautiful city", a graph-of-word will be generated after applying all of the pre-processing as discussed in [11] i.e. stop-words removal, lemmatization and word-stemming. From this graph, word-sequences s1 = {"Paris", "beautiful"}, s2 = {"beautiful", "city"} and onwards will be extracted, where d1 = {s1 + s2 + …}. These sequences are the final conversion of the entire document which is then compared with word-sequences of another document to produce clusters. Our approach also differs from the graph-of-word [11] as word-sequences further extracted from the graph; hence, the comparison is not of graph-of-word but of word-sequences extracted from graph-of-word. The un-weighted graph approach is used hence, the frequency of words is not taken into consideration; therefore, each word present in the document would not be neglected and retain the true meaning of the document.

B. Clustering Process

Document clustering [18] aims at solving specific data clustering problems in which data is in the form of documents. It partitions a group of documents into different clusters such that the documents belonging to one cluster are relevant to each other by some features like (words, meaning, etc.) and differs from documents in other clusters by same feature set. The difficult part is to identify which document belongs to which cluster. By merging different documents to same clusters that have higher similarity in terms of feature set and documents that have lower similarity are parts of different clusters. Hence, it can also be said that cluster will have higher intra-cluster similarity among the documents that are part of it and lower intra-cluster similarity with other clusters. One more challenge is to learn exactly how many clusters are there in document collection.

Clustering Techniques:

Traditional document clustering methods uses unsupervised document approach that does not have any predefined knowledge about the clusters in which the documents will be divided (unlabeled documents). However, in the real world scenarios, the user may have some background knowledge about the documents which could help the clustering process. This is known as the semi-supervised approach which has gained its importance specially for information retrieval processing. In semi-supervised document clustering, the algorithm is provided with some information about the data but not for all of the documents. In addition to this, further supervision may be provided in the form of constraints which directs the algorithm towards the common goal. For instance in web mining, certain document should be part of the same cluster or different can be provided to the algorithm in the form of instance-level constraints [19]. Instance level can be of two types: Must-Link and Cannot-Link [19] where the former one guides the clustering algorithm to include set of documents to the same cluster whereas the later one guides the algorithm not to include documents in the same cluster which belong to the Cannot-Link list.

Categories of Clustering:

For the semi-supervised clustering, different methods can be utilized which are presented by Zezula *et al* [18]. Clustering methods and be divided into two major categories such as Hierarchical v/s Flat and Partition v/s Overlapping.

Hierarchical v/s Partitioning Clustering:

Agglomerative hierarchical clustering [18] is approach in which at the initial stage, each document is a cluster in itself (candidate cluster) and based on the similarity measure between the documents, each document is clustered with another one to form bigger clusters [20]. Similarity measure calculation is the most important and time consuming process in the entire clustering process. Partition based algorithms is the other category of document clustering which created a one level partitioning of documents [21, 22, 23]. Similarity measures like k-means measures similarity between documents based on some feature set. At the initial level when all the documents are independent clusters,

similarity measure is calculated as the base case and documents that have higher similarity are merged together. As the clustering process is initiated, the documents are merged into different clusters and the similarity matrix is recalculated at each document merger until there is no further clustering possible.

1) *Improvement over unsupervised Approach:* In this research paper, a semi-supervised approach is used with constraints and the algorithm will be provided instance level constraints (Must-Link, Cannot-Link), Domain Level constraints and active learning algorithm which will guide the clustering process to utilize the instance level constraints and apply to other unlabeled documents. Using Word Sequence from Graph of Word in an unsupervised approach proved to have a better document representation in an unsupervised environment but using the same document representation with user knowledge should help improve the results more than the unsupervised approach.

III. PROPOSED APPROACH

The approach that has been proposed in this research can be divided into two sections, i.e. Document representation in which the document is represented as word sequences from Graph of word and the second part is the technique that is being used to perform clustering which is the semi-supervised document clustering approach with constraints.

C. Document Representation

In this section, the way in which the document has been presented is discussed. Document representation is an essential part of document clustering as if the compact form of document does not holds the true meaning of the actual document then the clustering process will be misguided.

Overview of Word Sequences from Graph-of-Word: To represent a document, word-sequences have been fetched from graph of word. In this technique, a graph is generated of an entire document with each distinct word as its vertex and words within a moveable window size have an edge between them. This moveable window on a particular word will have the word itself and the words which are adjacent to the word itself in the forward direction. This is known as the graph-of-word which is proposed by Rousseau and Vazirgiannis [17]. The significance of this technique is that it maintains the context in which the words were used. Considering the graph-of-word as the input, word-sequences can be extracted from the graph which is unlike to what was proposed by Li *et al* [11] as the author extracted sequences from the document itself whereas, in this research, the sequence is extracted from the graph which is a refined and pre-processed form of the document. Word sequences extracted from graph-of-word are the final document representation which would be used in the clustering process.

To summarize the process and understand it completely, a document d1 = "Paris is a beautiful city" is assumed. The document is passed through pre-processing techniques i.e. stop words removal, word-stemming and word lemmatization. The document d1 is then represented as a graph-of-word and then sequences 's' are obtained from that graph which can be written as d1 = { s1 + s2 + s3 , …..}. In this example, the sequences would be s1 = "Paris, beautiful", s2 = "Paris, city", s3 = "beautiful, city". These word sequences are the features of the document which carries the actual meaning of the document. These features are then compared with features of other documents to form clusters. The key difference in the approach of this research and graph-of-word [11] is that the authors of this paper compare different graph-of-words; the document was converted into graph-of-word then extract word-sequences which are then compared. Furthermore, this approach differs from [11] as the authors do not consider the word frequency and the input for approach is a graph rather than an entire document. The graph contains unique list of words as vertex and they are connected with other words through edges between them. This document representation is closer to the true representation semantically and therefore, the results obtained would be better than other document representation methods.

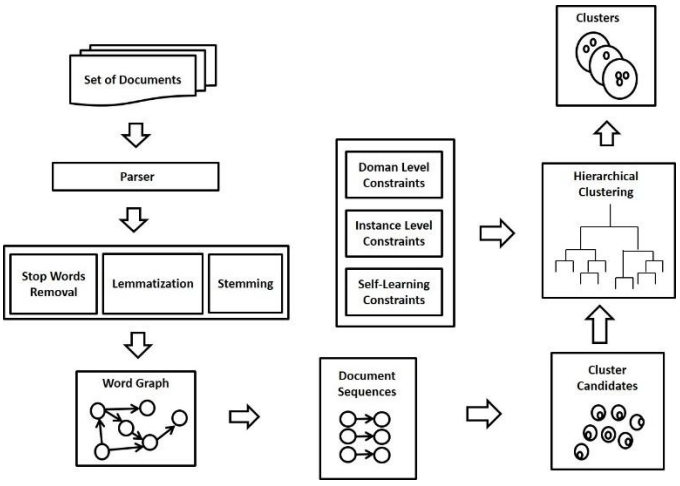

Fig. (2). WSFGW

1) Preprocessing: Documents that will be part of the clustering process are first pre-processed to refine and reduce the document. These are passed through different modules like stop word removal, word stemming and word lemmatization before they can be converted into WSFGW.

The first step of the pre-processing module is to pass the document through stop word removal routine that removes words like "is", "and" etc. If these words are removed from the document, the context of the document is not changed; hence, these words do not carry any weight. Onix Textual Retrieval Toolkit has published the list of Stop-Words with the name "stop-word list 1" and "stop-word list 2".

Second step of the pre-processing module is to pass the document through lemmatization routine which converts

every word to its root word such that a word "better" will be converted into "good". This conversion is necessary as both the words carry the same meaning but different forms would lead to different vertex in a graph; therefore, they are converted into root words and that will improve the similarity measure.

Third and the final step of the pre-processing module are to pass the document through word stemming routine. Word stemming means to convert adjective forms of word into root word such as "moved" can be converted into "move"; therefore, the document similarity measure will be improved as only root words will be considered.

2) Representation of Document: After the preprocessing module, the major challenge is the representation of the document. Inaccurate document representation may lead to inefficient and poor clustering results; therefore, representing the document into a form that retains all of the features of the document and helps the clustering process should be considered. The document is initially converted into graph-of-word using the QuickGraph library of C#. This graph represents the entire document which is then further processed and word sequences are extracted which is the final representation for a document. Following are the steps defined:

First step is to create a graph-of-word. A document is transformed into a graph which is called graph of word. This is an un-weighted directed graph where every unique word is represented as a vertex and the relationship between the words represented as edges. Every vertex has an edge with other adjacent words within the moveable window size which is taken as 3 in our experiment. The sequence of words can be shown in the graph as the direction of edges. The basic assumption is that words of an entire document have a relationship among each other within the moveable window and beyond the window size; the relationship was not considered. Without considering the meaning, this approach links together all co-occurring terms [17].

To demonstrate the graph-of-word approach, a sentence has been chosen from Wikipedia which is, "Information retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources". To build a graph of unique words, all the text is converted into lowercase and transformed into graph using a graph library. Hence, the resultant graph will contain only unique words of the sentence and an edge among vertex that are within the moveable window (with window size set to 3). Figure 3 shows the graph discussed by Rousseau and Vazirgiannis [17].

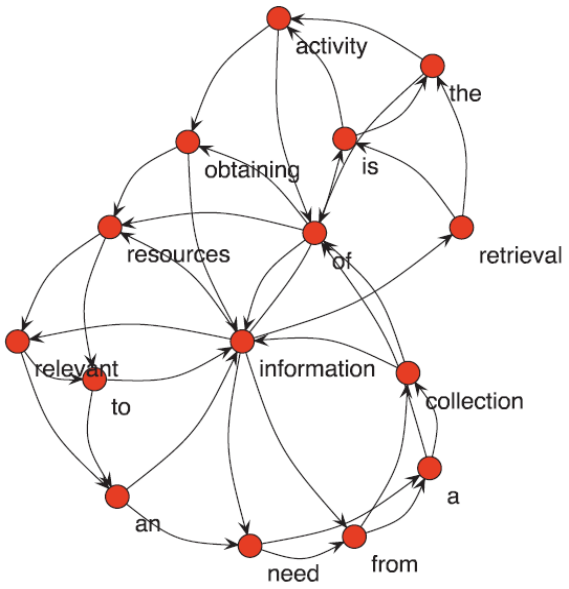

Fig. (3). Graph of Word [17]

Second step is to extract sequences. Word sequence is a pair of two or more words that are present in the actual document in an order. A sequence S can be denoted as S = {(w1, w2), (w2, w3) ...}. This does not strictly mean that the words would be adjacent to each other in the actual document; therefore, there can be words that are removed during the pre-processing phase. In word-sequence, order of the words in which they were present in the document is important and retained. As stated by Li *et al* [11], multiple occurrence of the same word may be treated as one. In this research approach, graph-of-word is the input to the word sequence and 2-word pair sequence is generated. To understand the working of word-sequence, the concept will be demonstrated using the same example that has been used in the previous section. The algorithm will start from the initial node and extract the word and following the edge, the word that it leads to, is part of the word sequence. This process is repeated until all of the vertices and their edges are traversed word by word. Hence sequences is obtained like s1 = {'Paris', 'beautiful'}, s2 = {"Paris", "city"}, s3 = {"beautiful", "city"} and so on. After all of the sequences are generated from the graph, this sequence is considered as the final representation of the document as they retain the relationship among the terms and the context in which they were utilized. Clustering process is executed on this form of document and this is highly compact form of document.

D. Document Clustering

To obtain clusters from all of the documents (i.e. word sequences from graph-of-word), hierarchical document clustering is used. Documents with higher similarity measure are merged together into same cluster and documents with lower similarity measure will be incorporated in other clusters. Similarity formula has been defined in section 4.6. Same formula will be used to calculate similarity between documents and clusters. Steps involved in the proposed approach are shown in Figure 2.

Semi-supervised: In this research, constraint based semi-supervised document clustering is used in which the clustering algorithm is capable to incorporate and utilize the user provided labels or constraints to achieve appropriate clustering [19]. In the next section, the constraints that have been used by the author are described.

Constraints: In the semi-supervised approach; to improve the clustering process, different types of constraints or predefined user knowledge is provided to the algorithm to improve the clustering outcome and are referred as constraints. These constraints guide the clustering algorithm to merge the documents together or to keep them apart. This technique improves the performance over unsupervised approaches. In this research, three types of constraints have been used which are mentioned below:

Instance level constraints are those constraints which direct the algorithm as which set of documents will be merged with documents and which documents will not be merged with other specified documents. Instance level constraints are divided into two categories i.e. Must-Link and Cannot-Link [19]. Must-Link contains set of documents that shows to be merged together into a cluster even if the similarity measure has less value. Opposite to this, Cannot-Link constraints direct the clustering algorithm not to merge these set of documents irrespective of their similarity measure value.

Domain level constraints are domain level information that is provided by the domain experts with the purpose of guiding the clustering process with meaningful grouping [24]. This information is provided in pairwise relationship between words and helps the clustering algorithm to identify cluster for a document that satisfies the provided information.

Another type of constraint is Active Learning in which the algorithm is allowed to self-learn the clustering process and with each cluster that is merged, the Active Learning provides instincts on merging the next document or not. Since Instance and Domain Level constraints cover only a subset of documents and documents that are not part of those constraints will be orphan for these constraints; therefore, Active learning process builds a self-learning mechanism that is improved with each merger. Therefore, those documents whose information is not available, this mechanism will guide the clustering process and improve the results. This algorithm works in the following fashion:

i) Construct Cannot-Link and Must-Link constraints.
ii) Compute Similarity matrix for every document.
iii) Repeat
   a. Start by selecting two documents with maximum similarity.
   b. Verify Cannot-Link constraint such that if a document exists in the Cannot-Link then do not merge the documents by setting the similarity value as 0;
   c. Apply Must-Link Constraints such that
      1. If a document exists only in the Must-Link constraint and does not violate the Cannot-Link constraint then merge them together into a cluster.
      2. If it exists in both the list then Cannot-Link must supersede the constraint.
   d. Apply Domain Level constraints to find similarity with previous clusters.
   e. Merge the documents with highest similarity value.
   f. Update similarity matrix.
iv) Until no more clusters can be formed.

IV. EXPERIMENTAL SETUP

In this section, the paper evaluates the performance of Word Sequence from Graph-of-Word over semi-supervised approach with sequence constraints against the other approaches New Suffix Tree Clustering, Graph of Word and Clustering based on Frequent Word Meaning Sequence. The algorithm was implemented on C# 3.5 and executed the experiments on Windows 8.1 based standard PC. For the creation of graph, Quick Graph library is used.

E. Data Set

Two different types of data set have been used to see the effectiveness of the algorithms. First dataset used is "The 20 Newsgroups" which is a collection of about 20,000 documents which are further divided into 20 different categories. It is a freely available dataset popular for experiments of machine learning. The other dataset used is TREC 9 and 10 which is a collection of web documents. The Text Retrieval Conference (TREC) supported by the US Branch of Defense and the National Institute of Standards and Technology (NIST). It was initially started as a feature of TIPSTER Text program in 1992. From these datasets, 4 subsets of dataset will be generated with sizes of 50, 100, 200 and 400 random documents to see the difference in results with respect to sizes of documents. These datasets are labelled as dataset50, dataset100, dataset200 and dataset400 respectively.

All of the documents among the selected datasets are pre-processed before the document representation phase. Pro-processing module includes stop-words removal, word-stemming and word lemmatization. Porter's Suffix Stripping algorithm [4] performs the word-stemming and Morpha-Stemmer is used for word lemmatization.

F. Similarity Measure

1) Document Similarity: Documents that are similar in nature based on some feature set should be included within same cluster whereas the documents that are dissimilar

should be placed in a different cluster. To identify the similarity among documents, following formula is used:

Similarity = $\frac{d1 \cup d2}{d1 \cap d2}$

Where d1 and d2 are two distinct documents from the document set. Documents having higher number of common sequences would have greater similarity measure.

2) *F-Score:* The f-score uses a combination of precision and recall values of cluster. The total number of documents can be denoted as n_a in class a and the number of documents in a cluster b as c_b. Hence objects present in class 'a' belonging to cluster b can be represented as c_{ab}. Therefore the cluster 'b' precision with respect to class 'a' can be represented as prec (a,b) which can further be written as $prec(a,b) = \frac{C_{ab}}{C_b}$ and re-call as $rec(a,b) = \frac{C_{ab}}{C_b}$ (i.e. recall of cluster 'b' with reference to class 'a'). Hence f-score can be written as:

$$F(a,b) = \frac{2 * prec(a,b) * rec(a,b)}{prec(a,b) + rec\ (a,b)}$$

Entire cluster's f-measure can be written as:

$$\sum_a \frac{a}{n} \max(F(a,b))$$

3) *Purity:* Purity means that a cluster has maximum number of valid objects from each class of b. It is calculated as following:

$$Purity = \sum_b \frac{c_b}{N}\ purity(f)$$

Where N represents the sum of the objects present in every cluster. Hence, this is used as the quantity instead of size of document.

4) *Entropy:* It measures the similarity of each cluster b. It is denoted as:

$$Ea = -\sum_{b \in L} prec\ (a\ , b) * \log(prec(a,b))$$

And entropy of the entire cluster as:

$$Entropy_c = \sum_{a \in C} \left(\left(\frac{N_a}{N} \right) * E_a \right)$$

Entropy should be minimum and purity should be maximum to have better clustering results.

V. RESULTS AND ANALYSIS

To analysis the data, four sets of document d1, d2, d3 and d4 has been chosen with number of documents as 50, 100, 200 and 400 respectively. First, the Word Sequence was compared from Graph of Word (WSFGW) approach as supervised and unsupervised with different level of pre-defined knowledge about the document by the user. Then using unsupervised document clustering with 20% constraints as input, different document representation algorithms were compared namely NSTC, Graph of Word (GOW) [17], Clustering based on Frequent Word Meaning Sequence (CFWMS) [11] with the approach of WSFGW.

G. Generated Clusters

Following are the results illustrated in table 1 and 2 obtained from comparing the Word Sequence from Graph of Word (WSFGW) with different levels of user knowledge about the documents as constraints.

Table 1. Number of Clusters based on different user knowledge

Word Sequence From Graph of Word						
# of Documents	**Unsupervised**	**Semi-supervised supervision**				Expected Clusters
	-	0%	10%	20%	50%	
50	4	4	5	5	5	5
100	7	7	8	9	9	9
200	15	15	11	12	13	13
400	18	18	19	20	21	21

Following are the results obtained after comparing WSFGW approach with different algorithms with semi supervised approach.

Table 2. Number of Clusters

# of Documents	Clusters with algorithm with 20% constraints				Expected Clusters
	NSTC	GOW	CFWMS	WSFGW	
50	8	5	4	5	5
100	14	8	8	9	9
200	17	12	9	12	13
400	24	18	16	20	21

From the results that have been achieved from this experiment, it is evident that the performance of WSFGW outperforms rest of the algorithms and it produces clusters which are near to the expected number of clusters.

H. F-Score

Following is the graph of F-Score based on the results obtained.

Fig. (4). F-Scores

This figure 4 shows that as the number of document sizes increases, the accuracy of result decreases but within all the algorithms, WSFGW has less number of incorrect clusters to document mapping.

I. Purity

Following figure 5 shows the purity of clusters which elaborates the results of NSTC, GOW, CFWMS having higher fluctuations whereas the results of WSFGW is more stable and clusters are more pure then others hence, it can concluded that the context behind the document is well-preserved.

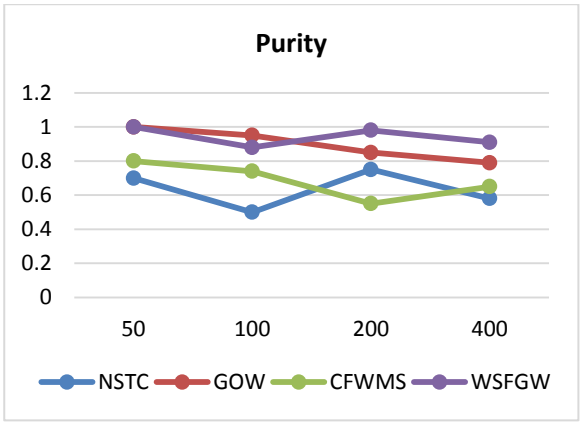

Fig. (5). Purity

J. Entropy

The least value of entropy results in better clustering; hence, it can be seen in figure 6 that NSTC performs the worst among the algorithms and our approach of WSFGW has better results.

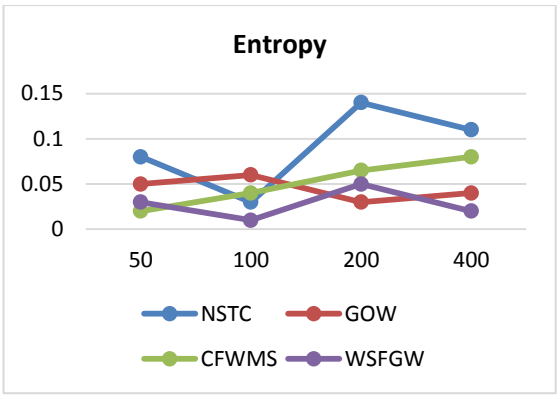

Fig. (6). Entropy

VI. CONCLUSION

It can be concluded based on the results obtained that the performance of Word Sequences from Graph of Word (WSFGW) outperforms other approaches and the performance is further improved with higher percentage of user defined knowledge. Since large amount of data is not possible but even with 20% of user knowledge, the performance of the clustering process is superior then unsupervised approaches.

Hence, the semi-supervised approach of WSFGW performs much better than the unsupervised approach. WSFGW represents the document in a compact form but retains the true meaning of the textual document and the semi supervised document clustering with sequence constraints produces better clustering results.

REFERENCES

[1] K. Jain, M. N. Murty and P. J. Flynn, "Data Clustering: a review," *ACM Computing Survey*, vol. 31, no. 3, pp: 264-323, 1999.

[2] Hung and D. Xiaotie, "Efficient Phrase-Based Document Similarity for Clustering," *IEEE Transaction on Knowledge and Data Engineering*, vol. 20, pp: 1217-1229, 2008.

[3] C. M. Fung, K. Wang and M. Ester, "Hierarchical document clustering using frequent Item sets," In *Proceedings of SIAM International Conference on Data Mining*, 2003.

[4] M. Rafi, M. Maujood, M. M. Fazal and S. M. Ali. "A comparison of two suffix tree-based document clustering algorithms." In *Proceedings of International Conference on Information and Emerging Technologies (ICIET)*, 2010.

[5] S. E. Robertson, S. Walker, K. Spärck Jones, M. Hancock-Beaulieu, and M. Gatford. "Okapi at TREC-3," In *Proceedings of the 3rd Text Retrieval Conference (TREC-3)*, 1994, pp: 109-126.

[6] G. Amati and C. J. Van Rijsbergen. “Probabilistic models of information retrieval based on measuring the divergence phpfrom randomness”. *ACM Transactions on Information Systems*, vol. 20, no. 4, pp: 357-389, 2002.

[7] Zhai and J. Lafferty. “A study of smoothing methods for language models applied to ad hoc information retrieval,” In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR ’01)*, 2001, pp: 334-342.

[8] K. M. Hammouda and M. S. Kamel, "Efficient Phrase-Based Document Indexing for Web Document Clustering," *IEEE Transaction on Knowledge and Data Engineering*, vol. 16, no. 10, pp: 1279-1296, 2004.

[9] C. Fellbaum, Eds., *WordNet: An Electronic Lexical Database*, MIT Press, 1998.

[10] R. Mihalcea and P. Tarau. “TextRank: Bringing order into texts”. In *Proceedings of Empirical Methods in Natural Language Processing (*EMNLP ’04), 2004.

[11] Y. Li, S. M. Chung, and J. D. Holt. "Text document clustering based on frequent word meaning sequences." *Data & Knowledge Engineering*, vol. 64, no. 1, pp: 381-404, 2008.

[12] G. Erkan and D. R. Radev. “LexRank: Graph-based lexical centrality as salience in text summarization”. *Journal of Artificial Intelligence Research*, vol. 22, no. 1, pp: 457-479, 2004.

[13] E. Motter, A. P. S. de Moura, Y.-C. Lai, and P. Dasgupta. “Topology of the conceptual network of language,” *Physical Review E*, vol. 65, no. 6, pp: 065-102, 2002.

[14] R. Blanco and C. Lioma. “Graph-based term weighting for information retrieval”. *Information Retrieval*, vol. 15, no. 1, pp: 54-92, 2012.

[15] O. Jespersen. *The Philosophy of Grammar*. The University of Chicago Press, 1929.

[16] Singhal, J. Choi, D. Hindle, D. Lewis, and F. Pereira. “AT&T at TREC-7”. In *Proceedings of the 7th Text Retrieval Conference (TREC-7)*, 1999, pp: 239-252.

[17] Rousseau and M. Vazirgiannis. "Graph-of-word and TW-IDF: new approach to ad hoc IR." In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 2013.

[18] P. Zezula, G. Amato, V. Dohnal and M. Batko. *Similarity Search-The Metric Space Approach*: Springer Science and Business Media, Inc., 2006.

[19] W. Zhao, Q. He, H. Ma and Z. She, "Effective semi-supervised document clustering via active learning with instance-level constraints." *Knowledge and Information Systems*, vol. 30, no. 3, pp: 569-587, 2012.

[20] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey, "Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections," In *Proceedings of Fifteenth Annual International ACM SIGIR Conference*, 1992, pp: 318-329.

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

[22] B. Larsen and C. Aone, "Fast and effective text mining using linear time document clustering," In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, pp: 16-22.

[23] M. Landau, “Fast parallel and serial approximate string matching”, *Journal of Algorithms*, vol. 10, no. 2, pp: 157–169, 1989.

[24] M. E. Ares and Á. Barreiro. "Constrained Text Clustering Using Word Trigrams." In *Proceedings of the 2nd Spanish Conference on Information Retrieval*, 2012.